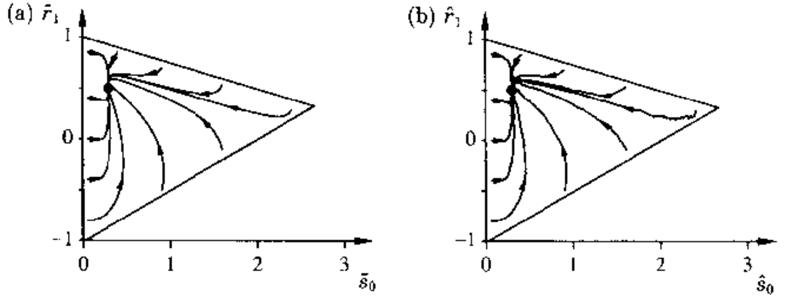

Figure 6.18 Phase plane for the controller parameters in Example 6.13 when recursive least-squares estimation is used. (a) Trajectories of the associated ODE. (b) Realizations of the difference equations. The parameter values corresponding to the minimum-variance controller are indicated by a dot.

The closed-loop system is described by

$$y(t) = \frac{(1 + cq^{-1})(1 + r_1 q^{-1})}{(1 + aq^{-1})(1 + r_1 q^{-1}) + s_0 q^{-1}(1 + bq^{-1})} e(t)$$

$$u(t) = \frac{-s_0(1 + cq^{-1})}{(1 + aq^{-1})(1 + r_1 q^{-1}) + s_0 q^{-1}(1 + bq^{-1})} e(t)$$

In this case,

$$\varphi^T(t-1) = \begin{pmatrix} u(t-2) & y(t-1) \end{pmatrix} \qquad \theta^T = \begin{pmatrix} r_1 & s_0 \end{pmatrix}$$

and

$$\varepsilon(t) = y(t)$$

Thus

$$f(\bar{\theta}) = \begin{pmatrix} r_{yu}(2) \\ r_y(1) \end{pmatrix} \qquad G(\bar{\theta}) = \begin{pmatrix} r_u(0) & r_{yu}(1) \\ r_{yu}(1) & r_y(0) \end{pmatrix}$$

where $r_y(\tau)$, $r_u(\tau)$, and $r_{yu}(\tau)$ are the covariance functions of y and u and the cross-covariance between y and u.

The stationary point is given by $f(\bar{\theta}) = 0$, which gives $r_{yu}(2) = 0$ and $r_y(1) = 0$. This is exactly the result obtained in Theorem 4.1. Figure 6.18(a) shows the phase plane of the ODE when recursive least-squares estimation is used. The stationary point corresponds to the minimum-variance controller, and the triangle indicates the stability boundary for the closed-loop system. Figure 6.18(b) shows realizations of the estimates $\hat{s}_0$ and $\hat{r}_1$ when recursive least-squares estimation has been used. The estimator is started with a very small step size. The realizations agree very well with the trajectories of the ODE. The ODEs have been simulated for $0 \le \tau \le 50$; 75,000 steps had to be simulated for the difference equations in Fig. 6.18(b). A forgetting factor of $\lambda = 0.99995$ was necessary to get close to the stationary point. □

EXAMPLE 6.14 Moving-average self-tuner

Consider an integrator with a time delay τ. (Compare Example 4.6.) For the time delay $\tau < h$ the system is described by

$$A(q) = q(q-1)$$
$$B(q) = (h-\tau)q + \tau = (h-\tau)(q+b) = (h-\tau)B'$$
$$C(q) = q(q+c)$$

where

$$b = \frac{\tau}{h-\tau} \qquad \text{and} \qquad |c| < 1$$

The system is minimum-phase, $|b| < 1$, when $\tau < h/2$. Moving-average controllers of different orders will now be analyzed. (Compare Section 4.2.)

Case 1 $(d = 1)$

The minimum-variance strategy obtained through

$$AR + (h-\tau)B'S = B'C$$

giving

$$R(q) = q + b$$
$$S(q) = \frac{1+c}{h-\tau} q$$

is the only possibility to get a moving average of order zero. Since the process zero is canceled, it is necessary for stability that the system be minimum-phase. The characteristic equation of K in Eq. (6.83) is in this case

$$(\lambda + 1)\left(\lambda + \frac{1}{1-bc}\right) = 0$$

Since $|b|$ and $|c|$ are both less than 1, it follows that the eigenvalues of K are both negative.

Case 2 $(d = 2)$

Since B is of first order and C is of second order, there are several possibilities to get an output that is a moving-average process. We get the following combinations:

Case	B^+	
2(a)	$q + b$	Minimum variance
2(b)	$q + b$	Deadbeat
2(c)	1	Moving average

To investigate the equilibria, first notice that Cases 2(a) and 2(b) can give stable equilibria only if $b < 1$ (i.e., $\tau < h/2$).

Case 2(a) corresponds to the minimum-variance controller. The characteristic equation of the matrix K is

$$\lambda^2 - \lambda\left(b + c + \frac{c}{1-bc}\right) + \frac{bc}{1-bc} = 0$$

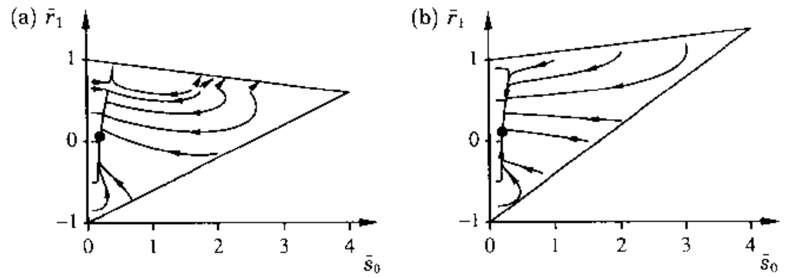

Figure 6.19 Simulation of the ODEs of the parameter estimates for the integrator when $d = 2$ and $c = -0.8$. (a) $\tau = 0.4$. (b) $\tau = 0.6$. The parameter values corresponding to the moving-average controller are indicated by dots.

Since b is nonnegative, it follows that this equation has roots in the right half-plane or at $\lambda = 0$ for all c in the interval $(-1, 1)$. The equilibrium is thus always unstable.

In Case 2(b) the characteristic equation of the matrix K is given by

$$\frac{(1+c^2)(1-bc)^2 + c^4(1-b^2)}{c(c-b)(1-bc)}\lambda^2 + \frac{1+c^2-bc}{c}\lambda + 1 = 0$$

This equation has all roots in the left half-plane if $b < c$.

In Case 2(c), moving-average control, the characteristic equation is

$$\lambda^2 + 2\lambda(b-c) + b(b-c) = 0$$

Since b is positive, it follows that this equation has its roots in the left half-plane if $b > c$. Notice that the moving-average controller is locally stable for $b > c$ even if $h/2 < \tau < h$, that is, when the controlled process is nonminimum-phase.

Summarizing, we find that if $d = 1$, there is only one equilibrium, which corresponds to the minimum-variance control. This equilibrium is locally stable only if $\tau < h/2$. When $d = 2$, there are three equilibria, corresponding to Cases 2(a), 2(b), and 2(c). Equilibrium 2(a) is always unstable; equilibrium 2(b) is stable if $b < c$; and equilibrium 2(c) is stable if $b > c$.

The phase portraits of the ODEs associated with the algorithm are shown in Fig. 6.19 for the case in which $d = 2$ and $c = -0.8$. When $\tau = 0.4$, there are three equilibria. They correspond to Case 2(a), which is a saddle point, Case 2(b), which is an unstable focus, and Case 2(c), which is a stable node. The stable node corresponds to the moving-average controller. The parameters are $r_1 = 0.08$ and $s_0 = 0.20$. For $\tau = 0.6$ there is only one equilibrium, which corresponds to the moving-average controller with the parameters $r_1 = 0.12$ and $s_0 = 0.20$. Figure 6.19 also shows that starting points exist for which the algorithm does not converge. The estimates are driven toward the stability boundary. □

The examples show how it is possible to use the associated ODE both to analyze the system and to get a feel for the behavior close to the stationary points as well as far away from them.

EXAMPLE 6.15 Local instability of a minimum-variance STR

Consider a process described by

$$\begin{aligned} y(t) &- 1.6y(t-1) - 0.75y(t-2) \\ &= u(t-1) + u(t-2) + 0.9u(t-3) + e(t) + 1.5e(t-1) + 0.75e(t-2) \end{aligned}$$

The B polynomial has zeros at

$$z_{1,2} = -0.50 \pm 0.81i$$

Furthermore,

$$C(z_{1,2}) = -0.40 \pm 0.40i$$

The real part of C is thus negative at the zeros of B. This implies that the parameters corresponding to minimum-variance control make an unstable equilibrium for the ODEs. Furthermore, it follows from Theorem 4.2 that these parameter values are the only possible equilibrium point for the parameters. The following heuristic argument indicates that the estimates are bounded:

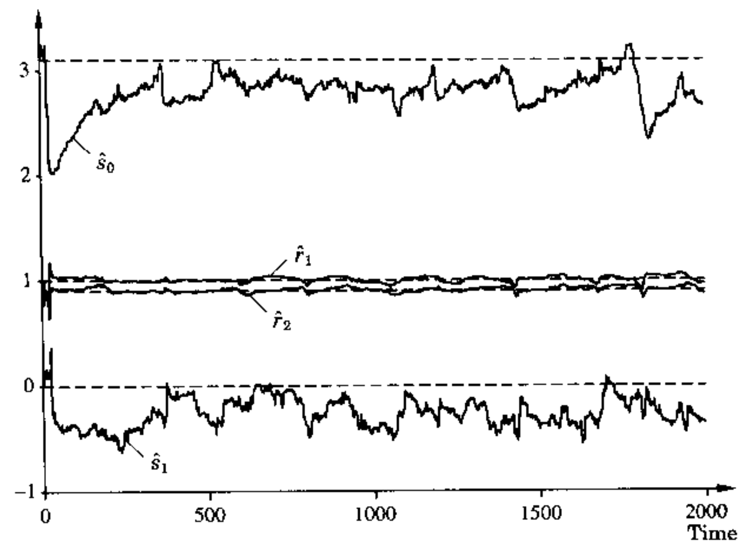

Figure 6.20 The parameter estimates when a self-tuning controller is used on the process in Example 6.15. The dashed lines correspond to the optimal minimum-variance controller.

If the parameters are such that the closed-loop system is unstable, the inputs and the outputs will be so large that they will dominate the stochastic terms in the model. The estimates will then quickly approach values that correspond to a deadbeat controller for the system, which gives a stable closed-loop system. This argument can be made rigorous (see Johansson (1988)). The estimates will thus vary in a bounded area without converging to any point. Figure 6.20 shows the parameter estimates when a direct self-tuning regulator with the controller structure

$$u(t) = -\frac{\hat{s}_0 + \hat{s}_1 q^{-1}}{1 + \hat{r}_1 q^{-1} + \hat{r}_2 q^{-2}} y(t)$$

is used. The simulation is initialized with values that correspond to the minimum-variance controller. The simulation is done by using RLS with a forgetting factor $\lambda = 0.98$. Figure 6.20 shows that the estimates try to reach the optimal values but are repelled when they get close. Notice also that the behavior is similar to that shown in Fig. 6.3. The example shows that a minimum-phase system exists in which the parameters corresponding to the minimum-variance controller are not a stable equilibrium for the self-tuning algorithm. This particular example led, in fact, to extensive research effort on the stability of stochastic self-tuners. □

6.9 ROBUST ADAPTIVE CONTROLLERS

In the previous sections we showed that both continuous-time and discrete-time adaptive controllers perform well in idealized cases. For the discrete-time self-tuning regulator, Assumptions A1–A4 in Theorem 6.7 were necessary to prove convergence and stability. The examples indicate that the MRAS algorithm in Eqs. (6.57) is incapable of dealing with unmodeled dynamics and disturbances. The insight given by the analysis also suggests various improvements of the algorithms. In this section, different ways to improve the robustness properties are discussed.

The first and most obvious observation is that the underlying controller structure must be appropriate. A pure proportional feedback is not appropriate, since the controller gain should be reduced at high frequencies to maintain robustness. Notice that a digital control law with appropriate prefiltering gives a very effective reduction of gain at frequencies higher than the Nyquist frequency associated with the sampling. However, any use of filtering in this way requires prior information about the unmodeled dynamics.

The examples also show that Theorem 6.7, although it is of significant theoretical interest, has limited practical value. The theorem clearly will not hold if Assumption A2 is violated. This assumption will not hold in a practical case, in which there are always unmodeled dynamics. It is also not realistic to neglect disturbances. This raises the possibility that global stability can be established only under unrealistic assumptions.

Theorem 6.7 also gives poor guidelines for the choice of controller complexity. To satisfy Assumption A2, it seems logical to increase the controller complexity. However, this will impose additional requirements on the input signal to maintain persistency of excitation.

Projections, Leakage, and Dead Zones

Equilibrium analysis based on averaging shows that the equilibria depend on the unmodeled dynamics and the nature of the command signal in a complicated way. Some general conclusions can be extracted, however. If the command signal is not persistently exciting of an order that corresponds to the number of updated parameters, the equilibrium set will in general be a manifold rather than a point. For systems that are linear in the parameters, the equilibria will actually be an affine set, which means that the controller gains may be very large on some points of the set. Small amounts of measurement noise or other disturbances may then cause a loss of equilibrium and result in drift of the parameters.

Several ideas have been proposed to modify the adaptive algorithms to avoid the difficulty. One possibility is to modify the algorithm so that the parameters are *projected* into a given fixed set. However, this requires that appropriate prior knowledge be available. For example, in Example 6.11 it is sufficient to project into a set such that $0 \le \hat{\theta}_2 \le 17$. A convenient way to obtain a controller with a finite gain is to introduce a path parallel to the process with gain ρ. Let G_r be the transfer function of the controller. The arrangement with the parallel path is equivalent to use a controller with the transfer function

$$G_r' = \frac{G_r}{1 + \rho G_r}$$

This is clearly bounded by $1/\rho$ when G_r has high gain.

In Section 5.3 we showed that the normalization in the estimator (6.2) is important to improve the properties of the algorithms. The normalization comes automatically when least-squares methods are used. Another modification is to change the parameter updating in Eq. (6.2) to

$$\frac{d\hat{\theta}}{dt} = \gamma \frac{\varphi e}{\alpha + \varphi^T \varphi} + \alpha_1(\theta^0 - \hat{\theta}) \tag{6.84}$$

where θ^0 is an *a priori* estimate of the parameters and $\alpha_1 > 0$ is an appropriate constant. The added term $\alpha_1(\theta^0 - \hat{\theta})$, sometimes called *leakage*, will make sure the estimates are driven toward θ^0 when they are far from θ^0. However, the modification will change the equilibrium. *A priori* knowledge is also required to choose θ^0 and α_1.

To avoid the problem of shift in equilibria, the following modification has also been suggested:

$$\frac{d\hat{\theta}}{dt} = \gamma \frac{\varphi e}{\alpha + \varphi^T \varphi} + \alpha_1 |e| (\theta^0 - \hat{\theta}) \tag{6.85}$$

A third way to avoid the difficulty is to switch off the parameter estimation if the input signal is not appropriate. There are several ways to determine when the estimates should be switched off. A simple way is to update only when the error is large, that is, to introduce a *dead zone* in the estimator. Such an approach is discussed below. However, it is necessary to have prior knowledge to select the dead zone.

It has also been suggested that the width of the dead zone be varied adaptively. From the equilibrium analysis it appears more appropriate to use a criterion based on persistent excitation. An alternative to switching off the estimate is to introduce intentional perturbation signals so as to ensure a proper amount of excitation.

Filtering and Monitoring of Excitation

From the system identification point of view the problem of unmodeled dynamics can be interpreted as follows. In fitting a low-order model to a system with complex dynamics, the results depend critically on the frequency content of the input signal. Precautions must thus be taken to ensure that the frequency content of the input signal is concentrated to the frequency range at which the simple model is expected to fit well. This indicates that the signals should be filtered before they are entered into the parameter estimator or the parameter update law. However, filtering alone is not sufficient, since it may happen that the input signal has only frequencies outside the useful frequency range. (A typical case is the system in Example 6.12 with $u_c(t) = \sin 16.09t$.) No amount of filtering can remedy such a situation. We are then left with only two options: to switch off the estimation or to introduce intentional perturbation signals.

Effects of Disturbances

Loss of robustness due to disturbances was found for the MRAS in Section 6.6. Similar problems can be encountered for discrete-time systems. In Section 6.5 direct self-tuning regulators were discussed in the ideal case, in which there are no disturbances, but the results can be extended in different directions to cover disturbances. Consider the case in which the process is described by

$$A(q)y(t) = B(q)u(t) + v(t) \tag{6.86}$$

where v is a bounded disturbance. To get some insight into what can happen, first consider an example. (See Egardt (1979).)

EXAMPLE 6.16 Bounded disturbances

Consider the system

$$y(t+1) + ay(t) = u(t) + v(t+1)$$

Use an adaptive control law with $A_o^* = A_m^* = 1$. (The desired response is thus $y_m(t+1) = u_c(t)$.) The control law is

$$u(t) = -\hat{\theta}(t)y(t) + u_c(t)$$

where

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \frac{y(t)}{1+y^2(t)} e(t+1)$$
$$e(t+1) = y(t+1) - \hat{\theta}y(t) - u(t)$$

Introduce

$$\tilde{\theta} = \hat{\theta} - \theta^0$$

where $\theta^0 = -a$. The closed-loop system can be described by the equations

$$\begin{aligned} \tilde{\theta}(t+1) &= \frac{1}{1+y^2(t)}\tilde{\theta}(t) + \frac{y(t)v(t+1)}{1+y^2(t)} \\ y(t+1) &= -\tilde{\theta}(t)y(t) + u_c(t) + v(t+1) \end{aligned} \tag{6.87}$$

To show that $y(t)$ may be unbounded, we want to construct a disturbance v and a command signal u_c such that the parameter error goes to infinity. Assume that initial conditions are chosen such that $\tilde{\theta}(1) = 0$ and $y(1) = 1$. Define

$$f(t) \triangleq \left(\sqrt{t(t-1)} - (t-1)\right)\left(1 + \frac{1}{t-1}\right) \qquad t = 2, 3, \ldots, T-5$$

for some large T. Choose the following disturbance:

$$v(t) = 1 - \frac{1}{\sqrt{t-1}} + f(t) \qquad t = 2, 3, \ldots, T-5$$

and the following command signal:

$$u_c(t-1) = \frac{1}{\sqrt{t}} - f(t) \qquad t = 2, 3, \ldots, T-5$$

The signals v and u_c are bounded. A straightforward calculation gives

$$\tilde{\theta}(t) = \sqrt{t} - 1$$
$$y(t) = \frac{1}{\sqrt{t}}$$

for $t = 1, \ldots, T-5$. Further, let

$$v(t) = 0 \qquad t = T-4, \ldots, T$$
$$u_c(t-1) = \begin{cases} 0 & t = T-4 \\ 1 & t = T-3, \ldots, T \end{cases}$$

It can then be verified that $\tilde{\theta}(t)$ and $y(t)$ for large T are approximately given by the following table.

t	$\tilde{\theta}(t)$	$y(t)$
$T-4$	$\sqrt{T}$	-1
$T-3$	$\dfrac{\sqrt{T}}{2}$	$\sqrt{T}$
$T-2$	$\dfrac{1}{2\sqrt{T}}$	$-\dfrac{T}{2}$
$T-1$	$\dfrac{1}{\sqrt{T}T^2}$	$\dfrac{\sqrt{T}}{4}$
T	$\dfrac{16}{\sqrt{T}T^3}$	1

Now choose $v(T+1)$ and $u_c(T)$ such that $\tilde{\theta}(T+1) = 0$ and $y(T+1) = 1$. The state vector of Eqs. (6.87) is then equal to the initial state. By repeating the procedure for increasing values of T, a subsequence of $y(t)$ will increase as $-T/2$ and therefore is unbounded. □

Example 6.16 shows that the algorithm may behave badly even if it is assumed that the disturbances are bounded. Robustness against bounded disturbances can be obtained by using conditional updating as shown in the following theorem.

THEOREM 6.8 Conditional updating
Consider the plant (6.86) where v is a disturbance that is bounded by

$$\sup_t \left| \frac{R}{A_o A_m B} v \right| \leq C_1$$

where R is the polynomial in the feedback law and C_1 is a constant. Assume that the direct adaptive algorithm defined by Eqs. (6.41) and (6.42) is used, with the modification that the parameters are updated only when the estimation error is such that

$$|e| \geq \frac{2C_1}{2 - \max(b_0/r_0, 1)}$$

Let Assumptions A1–A3 hold, and assume in addition that $0 < b_0 < 2r_0$. Then the inputs and outputs of the closed-loop system are bounded. □

Proofs of this theorem can be found in Egardt (1979) and Goodwin and Sin (1984). The modification of the algorithm is referred to as *conditional updating* or *introduction of a dead zone in the estimator*.

Of course, the result is of limited practical value because it requires an upper bound on the disturbance, which is not known *a priori*. The bound also depends on the ratio $b_0/r_0 = b_0/\hat{b}_0$, where b_0 is the instantaneous gain. The estimate of this gain is thus essential. If $b_0/r_0 = 1$ and $A_o = A_m = 1$, it follows that $R = B$, and the condition for updating becomes

$$|e(t)| \geq 2 \sup |v(t)|$$

This means that the estimate will be updated when the estimation error is twice as large as the maximum noise amplitude.

Another modification of the algorithm also leads to bounded signals. The modification consists of using the updating law of Eqs. (6.41) if the magnitude of the estimates is less than a given bound and to project into a bounded set if Eqs. (6.41) give estimates outside the bounds. We refer to Theorem 4.4 of Egardt (1979) for details. This method will, of course, require that the bounds on the parameters be known *a priori*.

Signal Normalization

Various modifications of the adaptive algorithm are discussed in more detail in Chapter 11. Therefore only a few sketchy remarks are given here. Notice that Theorem 6.8 gives stability conditions for adaptive control applied to the model (6.86), when v is a bounded disturbance. Unmodeled dynamics can, of course, be modeled by Eq. (6.86), but v will no longer be bounded, since it depends on the inputs and outputs. By introducing the signal defined by

$$Cr(t) = \max\left(|u(t)|, |y(t)|\right)$$

where C is a stable filter, and introducing the normalized signals

$$\bar{y} = \frac{y}{r}, \qquad \bar{u} = \frac{u}{r}, \qquad \bar{v} = \frac{v}{r}$$

the model of Eq. (6.86) can be replaced by

$$A\bar{y} = B\bar{u} + \bar{v}$$

where $\bar{v}$ is now bounded. By invoking Theorem 6.8, it can be established that adaptive control with a dead zone or projection gives a system with bounded signals. The detailed justification is complicated.

The Minimum-Phase Assumption

In Theorem 6.7 and for the MRAS the process is required to be minimum-phase. This assumption is used to conclude that the input signal is bounded when the output is bounded. The minimum-variance controller, which cancels the open-loop process zeros, cannot be used when the process is nonminimum-phase.

Instead, the LQG self-tuner or the moving-average controller with increased prediction horizon can be used.

It should be remarked that sampled data systems often can be non-minimum-phase because of "sampling zeros" even if the continuous-time system that is sampled is minimum-phase. These zeros are given by the following theorem.

THEOREM 6.9 Limiting sampled-data zeros

Let $G(s)$ be a rational function

$$G(s) = K\frac{(s-z_1)(s-z_2),\dots,(s-z_m)}{(s-p_1)(s-p_2),\dots,(s-p_n)} \tag{6.88}$$

and let $H(z)$ be the corresponding pulse transfer function. Assume that $m < n$. As the sampling period $h \to 0$, m zeros of H go to 1 as $\exp(z_i h)$, and the remaining $n - m - 1$ zeros of H go to the zeros of $B_{n-m}(z)$, where $B_k(z)$ is the polynomial

$$B_k(z) = b_1^k z^{k-1} + b_2^k z^{k-2} + \cdots + b_k^k \tag{6.89}$$

and

$$b_i^k = \sum_{l=1}^{i}(-1)^{i-l} l^k \binom{k+1}{i-l} \qquad i = 1,\dots,k \tag{6.90}$$

The first polynomials B_k are

$$B_1(z) = 1$$

$$B_2(z) = z + 1$$

$$B_3(z) = z^2 + 4z + 1$$

□

This theorem is proved in Åström *et al.* (1984). It implies that direct methods for adaptive control that require that the plant be minimum-phase cannot be used with too short a sampling period. When very fast sampling is required, a continuous-time representation may then be preferable. Another possibility is to describe the system in the *delta operator*, defined by

$$\delta = \frac{q-1}{h}$$

or in *Tustin's operator*:

$$\Delta = \frac{1}{2h}\frac{q-1}{q+1}$$

This yields parameterizations that give a much better resolution at $q = 1$. The δ operator gives a description that is equivalent to the q operator description. The advantage of the transformation is that the δ operator description has better numerical properties when the sampling is fast. All the poles of the q

operator form are clustered around the point $q = 1$. This gives rise to numerical sensitivity. For the δ operator it can be shown that the limiting value

$$\lim_{h\to 0}\frac{B_h(\delta)}{A_h(\delta)} = \frac{B_0(\delta)}{A_0(\delta)}$$

is such that the coefficients in B_0 and A_0 are the same as the coefficients in the continuous-time transfer function. This implies that the structure of the transfer function in the δ operator is essentially the same as that of the continuous-time transfer function, provided that the sampling period is sufficiently short.

The High-Frequency Gain

For a process that has no right half-plane zeros, the standard direct discrete-time algorithm is based on the model

$$A_o^* A_m^* y(t+d) = b_0\left(R^* u(t) + S^* y(t)\right)$$

where b_0 is the coefficient of the first nonvanishing term in the B polynomial. With some abuse of language this coefficient is called the *high-frequency gain* because it is the first nonvanishing coefficient of the impulse response. For continuous-time systems the transfer function of the process is approximately $G(s) = b_0 s^{-dh}$. In Theorem 6.7 it was required that the sign of the coefficient b_0 be known. There are several ways to deal with the parameter b_0. It may be absorbed into R and S and estimated. The polynomial R then has the form

$$R(q) = r_0 q^k + r_1 q^{k-1} + \cdots + r_k$$

The problem with this approach is that some safeguards must be taken to avoid the estimate r_0 becoming too small. Another possibility is to introduce a crude fixed estimate of b_0. The following analysis shows what happens when this is done. Let the true system be

$$y(t+1) = b_0\left(u(t) + \psi^T(t)\theta^0\right)$$

and let the model be

$$y(t+1) = r_0\left(u(t) + \psi^T(t)\theta\right) = r_0 u(t) + \varphi^T(t)\theta$$

With zero command signal the control law becomes

$$u(t) = -\psi^T(t)\hat\theta(t)$$

The equation for parameter updating is

$$\hat\theta(t+1) = \hat\theta(t) + P(t+1)\varphi(t)e(t+1)$$

where

$$e(t+1) = y(t+1) = b_0 u(t) + b_0 \psi^T(t)\theta^0$$
$$= -b_0\psi^T(t)\left(\hat\theta(t) - \theta^0\right) = -\frac{b_0}{r_0}\varphi^T(t)\left(\hat\theta(t) - \theta^0\right)$$

The estimation error is thus governed by

$$\tilde\theta(t+1) = \left(I - \frac{b_0}{r_0}P(t+1)\varphi(t)\varphi^T(t)\right)\tilde\theta(t)$$

With a pure projection algorithm we have

$$P(t+1) = \frac{1}{\varphi^T(t)\varphi(t)}$$

In this case the matrix in large parentheses has one eigenvalue $(1 - b_0/r_0)$ and the remaining eigenvalues 1. With least-squares updating, the averaged equation for $\tilde\theta$ becomes

$$\tilde\theta(t+1) = \left(1 - \frac{b_0}{r_0}\right)\tilde\theta(t)$$

Hence, to remain stable, it must be required that

$$0 < \frac{b_0}{r_0} < 2$$

If an algorithm with a fixed r_0 is used, it is convenient to absorb r_0 in the scaling of the signals. This is discussed in more detail in Chapter 11. When the parameter b_0 is estimated, it can be treated like the other parameters. However, because of the special structure of the model it is useful to use special algorithms such as the ones discussed in Section 5.8.

Universal Stabilizers

An interesting class of adaptive algorithms was discovered during attempts to investigate whether Assumption A3 is necessary. The following question was posed. Consider the scalar system

$$\frac{dy}{dt} = ay + bu \tag{6.91}$$

where a and b are constants. Does there exist a feedback law of the form

$$\begin{aligned} u &= f(\hat\theta, y) \\ \frac{d\hat\theta}{dt} &= g(\hat\theta, y) \end{aligned} \tag{6.92}$$

that stabilizes the system for all values of a and b? Morse (1983) suggested that there are no rational f and g that solve the problem. Morse's conjecture was verified by Nussbaum (1983), who proved the following result.

THEOREM 6.10 Universal stabilizer

The control law of Eqs. (6.92), with

$$\begin{aligned} f(\hat\theta, y) &= y\hat\theta^2\cos\hat\theta \\ g(\hat\theta, y) &= y^2 \end{aligned} \tag{6.93}$$

and $\hat\theta(0) = 0$, stabilizes Eq. (6.91).

Proof: The closed-loop system is described by

$$\frac{dy}{dt} = ay + by\hat\theta^2\cos\hat\theta$$
$$\frac{d\hat\theta}{dt} = y^2$$

Since $\hat\theta(0) = 0$ and $d\hat\theta/dt \ge 0$, it follows that $\hat\theta(t)$ is nonnegative and nondecreasing. $\hat\theta(t)$ is also bounded, which is shown by contradiction. Hence assume that $\lim_{t\to\infty}\hat\theta(t) = \infty$. Multiplication of the differential equation for y by y gives

$$y\frac{dy}{dt} = ay^2 + by^2\hat\theta^2\cos\hat\theta = a\frac{d\hat\theta}{dt} + b\hat\theta^2\cos\hat\theta\,\frac{d\hat\theta}{dt}$$

Integration with respect to time gives

$$y^2(t) = y^2(0) + 2a\hat\theta(t) + 2b\int_0^{\hat\theta(t)} x^2\cos x\,dx$$

Hence

$$\frac{y^2(t)}{\hat\theta(t)} = \frac{y^2(0)}{\hat\theta(t)} + 2a + \frac{2b}{\hat\theta(t)}\int_0^{\hat\theta(t)} x^2\cos x\,dx$$

But

$$\frac{1}{\hat\theta}\int_0^{\hat\theta} x^2\cos x\,dx = \hat\theta\sin\hat\theta + 2\cos\hat\theta - \frac{2}{\hat\theta}\sin\hat\theta$$

Hence

$$\liminf_{t\to\infty}\frac{1}{\hat\theta}\int_0^{\hat\theta} x^2\cos x\,dx = -\infty$$

This gives

$$\liminf_{\hat\theta\to\infty}\frac{y^2(t)}{\hat\theta(t)} = -\infty$$

which is a contradiction because $y^2/\hat\theta$ is nonnegative. It thus follows that

$$\lim_{t\to\infty}\hat\theta(t) = \theta^0 < \infty$$

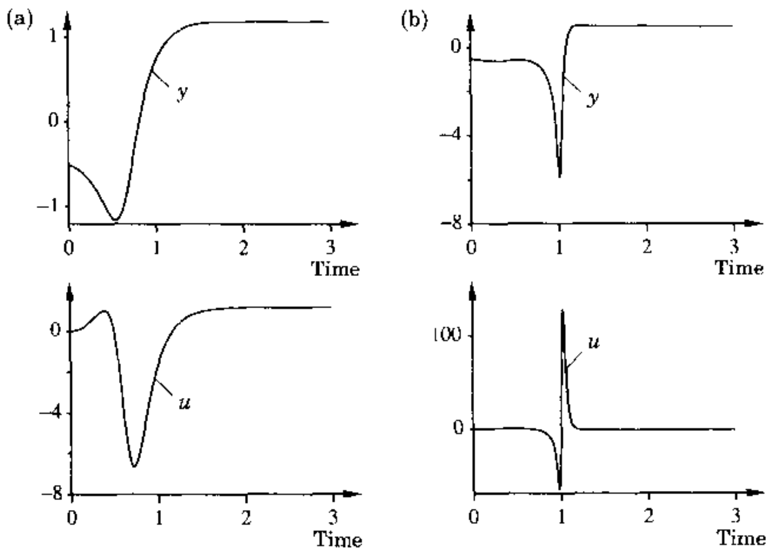

Figure 6.21 Simulation of the control law of Eqs. (6.94) applied to the plants (a) $G(s) = 1/(1 - s)$ and (b) $G(s) = 1/(s - 1)$.

Integration of the equation for $\hat{\theta}$ gives

$$\hat{\theta}(t) = \int_0^t y^2(t)\,dt$$

It then follows that

$$\lim_{t \to \infty} y(t) = 0$$

□

The behavior of a universal stabilizer is illustrated in Fig. 6.21. A reference value is used in the simulations, and the control law is then modified to

$$\begin{aligned} f(\hat{\theta}, y) &= (u_c - y)\hat{\theta}^2 \cos\hat{\theta} \\ g(\hat{\theta}, y) &= (u_c - y)^2 \end{aligned} \tag{6.94}$$

Notice that the control law of Eqs. (6.94) can be interpreted as proportional feedback with the gain $k = \hat{\theta}^2 \cos\hat{\theta}$. The behavior of the control law can be interpreted as follows. Sweep over all possible controller gains and stop when a stabilizing gain has been found. The function g can be interpreted as the rate of change of the gain sweep. The rate is large for large errors and small for small errors. The form $\cos\hat{\theta}$ makes sure that the gains can be both positive and negative. Universal stabilizers may show very violent behavior. This not surprising, since the system may be temporarily unstable during the sweep over the gains.

The control law of Eqs. (6.94) is useful because it does not contain any parameters that relate to the system that it stabilizes. It is therefore called a *universal stabilizer*. However, the control law is restricted to a first-order system. In attempting to generalize Theorem 6.10 to higher-order systems, the following question was posed. How much prior information about an unknown system is required to stabilize it? This question was answered in a general setting by Mårtensson (1985), who showed that it is sufficient to know the order of a stabilizing fixed-gain controller. If a transfer function is given, it is unfortunately a nontrivial task to find the minimal order of a stabilizing controller.

6.10 CONCLUSIONS

Analysis of adaptive systems is difficult because they are complicated. A number of different methods have been used to gain insight into the behavior of adaptive systems. The theory is useful to show fundamental limitations of the algorithms and to point out possible ways to improve them.

In this chapter a basic stability theorem has been derived on the basis of standard tools of the theory of difference equations. To show stability and convergence, it is necessary to make quite restrictive assumptions about the system to be controlled. The consequences of violating these assumptions have been analyzed.

It has been shown that analysis of equilibria and local properties around equilibria can be explored by the method of averaging. This method can also be applied to investigate global properties. Averaging can be applied in many different situations. For deterministic problems it can be used for steps and periodic signals. It can also be applied to stochastic signals. Averaging methods have also been applied to analyze what happens when adaptive systems are designed on the basis of simplified models. To apply averaging methods, it is necessary to use small adaptation gains. Unfortunately, there are no good methods to determine analytically how small the gains should be. It is also demonstrated that adaptive systems may have very complex behavior for large adaptation gains. Mechanisms that may lead to instability have been discussed. One mechanism is associated with lack of a parameter equilibrium or local instability of the equilibrium. Other mechanisms are parametric excitation and high adaptation gain. The last two mechanisms can be avoided by choosing a small adaptation gain.

PROBLEMS

6.1 Consider the indirect continuous-time self-tuning controller in Example 3.6. Collect all equations that describe the self-tuner, and show that they can be written in the form

$$\frac{d\xi}{dt} = A(\vartheta)\xi + B(\vartheta)u_c$$

$$\begin{pmatrix} e \\ \varphi \end{pmatrix} = C(\vartheta)\xi + D(\vartheta)u_c$$

$$\vartheta = \chi(\theta)$$

$$\frac{d\theta}{dt} = P\varphi e$$

$$\frac{dP}{dt} = \alpha P - P\varphi\varphi^T P$$

Give explicit expressions for all components of the vectors ξ, φ, ϑ, and θ and the matrix P.

6.2 Consider a system with unknown gain whose transfer function is SPR. Show that a closed-loop system that is insensitive to variations in the gain is easily obtained by applying proportional feedback. Carry out a detailed analysis for the case in which the transfer function is $G(s) = 1/(s+1)$.

6.3 Consider an MRAS for adjustment of a feedforward gain. Assume that the system is designed on the basis of the assumption that the process dynamics are

$$G(s) = \frac{a}{s+a}$$

(a) Investigate the behavior of the systems obtained with the SPR and MIT rules when the real system has the transfer function

$$G(s) = \frac{ab^2}{(s+a)(s+b)^2}$$

Determine in particular which frequency ranges give stable adaptation rules for sinusoidal command signals.

(b) Consider the MRAS based on the SPR rule when the reference signal is constant and when an additional constant load disturbance is acting on the input of the process. Investigate how the load disturbance influences the stationary point of the total system. Investigate the local stability properties through linearization.

6.4 Consider an MRAS for adjustment of a feedforward gain based on the MIT rule. Let the command signal be

$$u_c = a_1 \sin \omega_1 t + a_2 \sin \omega_2 t$$

and assume that the process has the transfer function

$$G(s) = \frac{1}{(s+1)^3}$$

Derive conditions for the closed-loop system to be stable.

6.5 Consider Theorem 6.7. Generalize the results to cover the case in which the polynomial B^* has isolated zeros on the unit circle.

6.6 Consider the system described by

$$y(t) = u(t - d)$$

Assume that a direct adaptive control (e.g., with $A_o^* = A_m^* = 1$) is designed according to the assumption that $d = 1$. Investigate how this controller behaves when applied to a system with $d = 2$.

6.7 Construct a proof analogous to Theorem 6.7 for continuous-time systems.

6.8 Consider the system

$$y(t) = u(t-1) + a$$

where a is an unknown constant. Construct an adaptive control law that makes y follow a command u_c asymptotically. Prove that it converges.

6.9 Consider the continuous-time model

$$y(t) = \varphi^T(t)\theta$$

Let the parameter θ be estimated by

$$\frac{d\hat\theta}{dt} = \gamma \frac{\varphi(t)}{\alpha + \varphi^T(t)\varphi(t)} e(t)$$

where $\gamma > 0$ and $\alpha > 0$ are real constants and

$$e(t) = y(t) - \varphi^T(t)\hat\theta(t)$$

Assume that $y(t)$ is given by

$$y(t) = \varphi^T(t)\theta_0$$

Prove that

$$|\hat\theta(t) - \theta_0| \le |\hat\theta(s) - \theta_0| \le |\hat\theta(0) - \theta_0| \qquad t > s > 0$$

and that

$$\frac{|e(t)|}{\sqrt{\alpha + \varphi^T(t)\varphi(t)}} \to 0$$

as $t \to \infty$.

6.10 Consider the system in Example 6.12. Interpret the results as if the adaptive algorithm tried to estimate parameters a and b in the transfer function $G(s) = b/(s+a)$. Use Eqs. (6.70) to show that

$$\hat a = \frac{229 - 31\omega^2}{259 - \omega^2}$$

$$\hat b = \frac{458}{259 - \omega^2}$$

Determine the parameters for $\omega = 2.72$ and $\omega = 17.03$. Explain the results by evaluating $G(s)$ for the corresponding frequencies.

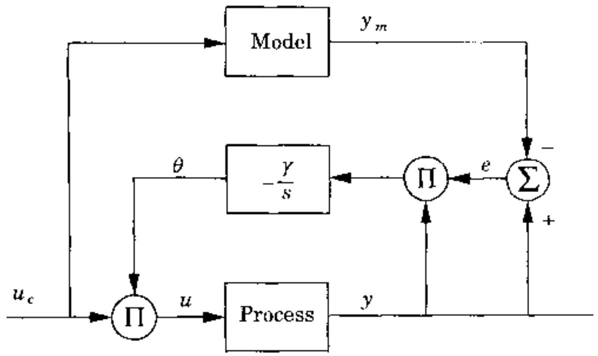

Figure 6.22 Adaptive feedforward controller in Problem 6.11.

6.11 A feedforward gain is adapted as shown in the block diagram in Fig. 6.22. The model is given by

$$\frac{dy_m}{dt} = -y_m + u_c$$

The process is not linear, however, but is given by

$$\frac{dy}{dt} = -y - ay^3 + u$$

Let $\gamma = 1$ and $u_c = 1$.

(a) What are the equilibrium points of the system?

(b) Linearize the system around the equilibrium points, and determine how the stability of the linearized system depends on the parameter a.

(c) Simulate the behavior of the nonlinear adaptive system to verify the results in part (b).

6.12 An integrator process

$$G(s) = \frac{1}{s}$$

is to be controlled by the error feedback law

$$sU(s) = (4s + \theta)(U_c(s) - Y(s))$$

where $U(s)$, $U_c(s)$, and $Y(s)$ are the Laplace transforms of the input, reference, and output signals, respectively. The desired response of the closed-loop system is given by the transfer function

$$G_m(s) = \frac{4(s+1)}{(s+2)^2}$$

An MRAS has been designed, giving the parameter update law

$$\frac{d\theta}{dt} = -\gamma e(t)\left(\frac{1}{(p+2)^2}u_c(t)\right)$$

where $e = y - y_m$.

(a) Find the equilibrium parameter set of the parameter update law, giving a parameter estimate that is constant for any reference input $u_c(t)$. Give an expression for the averaged nonlinear differential equation of the parameter update law.

(b) Determine the local stability of the equilibrium parameter set for a sinusoidal reference signal $u_c(t) = \sin \omega t$ by examining the characteristic polynomial of the linear differential equation obtained by linearizing the averaged differential equation around the equilibrium parameter set. Determine for what frequencies the linearized equation is stable.

6.13 Formulate the averaging equation for a discrete-time algorithm corresponding to Eq. 6.49.

6.14 Consider discrete-time adaptive control of the system

$$y(t+1) = ay(t) + bu(t)$$

Derive an MRAS that gives a closed-loop system

$$y_m(t+1) = a_m y_m(t) + b_m u_c(t)$$

Use averaging methods to analyze the system when the command signal is a step and a sinusoid.

6.15 Consider Problem 6.14. Investigate the behavior of the system when the command signal is a step and when there is sinusoidal measurement noise.

6.16 Consider the MRAS given by Eqs. (6.57). Investigate the local behavior of the closed-loop system when the command signal is a sinusoid and the gradient method

$$\frac{d\hat{\theta}}{dt} = \gamma\varphi e$$

is replaced with a least-squares method of the form

$$\frac{d\hat{\theta}}{dt} = P\varphi e$$

$$\frac{dP}{dt} = -P\varphi\varphi^T P + \lambda P$$

6.17 Show that there is no constant-gain controller that can simultaneously stabilize the systems $G(s) = 1/(1+s)$ and $G(s) = 1/(1-s)$.

6.18 Show that there is a fixed-gain controller that will simultaneously stabilize the systems $G(s) = 1/(s+1)$ and $G(s) = 1/(s-1)$.

6.19 Consider the MRAS given by Eqs. (6.57). Make a simulation study to investigate the consequences of introducing leakage as described by Eqs. (6.84) and (6.85) in the estimation algorithm. Study sinusoidal command signals as well as step commands and measurement noise.

6.20 Consider the MRAS in Problem 6.4. Make a simulation study to investigate the consequences of using conditional updating. Study sinusoidal command signals as well as step commands and measurement noise.

6.21 Consider the system in Problem 6.4. Let the input be sinusoidal with frequency ω. Investigate the effects of sinusoidal measurement noise on the system.

6.22 Consider direct algorithms for control of the system

$$y(t+1) = ay(t) + bu(t)$$

to give an input-output relation

$$y_m(t+1) = a_m y(t) + b_m u_c(t)$$

Investigate by simulation the convergence rates obtained when $\hat{b}$ is fixed to different values.

6.23 Investigate the behavior of the universal stabilizer in the presence of measurement noise.

6.24 Consider a system for adjustment of a feedforward gain based on the MIT rule. Let the command signal be $u_c(t) = \sin \omega t$, and let $G(s) = 1/(s+1)$. Simulate the parameter behavior for the MIT rule with adaptation gains $\gamma = 10$ and $\gamma = 11$. Compare the analysis in Example 6.8.

6.25 Consider the simulation shown in Fig. 6.11, which was performed with adaptation gain $\gamma = 1.0$. Repeat the simulation with different adaptation gains.

REFERENCES

A standard text on nonlinear systems is:

Guckenheimer, J., and P. Holmes, 1983. *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields*, Applied Mathematics Series. New York: Springer-Verlag.

The stability problem has been of major concern since the MRAS was proposed. Flaws in earlier stability proofs were pointed out in:

Feuer, A., and A. S. Morse, 1978. "Adaptive control of single-input single-output linear systems." *IEEE Trans. Automat. Contr.* **AC-23**: 557–569.

The proof of Theorem 6.7 follows the ideas in:

Goodwin, G. C., P. J. Ramadge, and P. E. Caines, 1980. "Discrete-time multivariable adaptive control." *IEEE Trans. Automat. Contr.* **AC-25**: 449–456.

Equivalent results for continuous-time systems are presented in:

Morse, A. S., 1980. "Global stability of parameter-adaptive control systems." *IEEE Trans. Automat. Contr.* **AC-25**: 433–439.

Narendra, K. S., Y.-H. Lin, and L. S. Valavani, 1980. "Stable adaptive controller design. Part II: Proof of stability." *IEEE Trans. Automat. Contr.* **AC-25**: 440–448.

Many variations of the stability theorem are given in:

Goodwin, G. C., and K. S. Sin, 1984. *Adaptive Filtering Prediction and Control*, Information and Systems Science Series. Englewood Cliffs, N.J.: Prentice-Hall.

Related results are presented in:

de Larminat, P., 1979. "On overall stability of certain adaptive control systems." *Preprints of the 5th IFAC Symposium on Identification and System Parameter Estimation*, pp. 1153–1159. Darmstadt, Germany.

Egardt, B., 1980a. "Stability analysis of discrete-time adaptive control schemes." *IEEE Trans. Automat. Contr.* **AC-25**: 710–716.

Egardt, B., 1980b. "Stability analysis of continuous-time adaptive control systems." *Siam J. Contr. Optimiz.* **18**: 540–558.

Gawthrop, P. J., 1980. "On the stability and convergence of a self-tuning controller." *Int. J. Contr.* **31**: 973–998.

Kumar, P. R., 1990. "Convergence of adaptive control schemes using least-squares parameter estimates." *IEEE Trans. Automat. Contr.* **AC-35**: 416–424.

A stability analysis for bounded disturbances is given in:

Egardt, B., 1979. *Stability of Adaptive Controllers*. Lecture Notes in Control and Information Sciences, vol. 20. Berlin: Springer-Verlag.

The case of mean square bounded disturbances was investigated in:

Praly, L., 1984. "Stochastic adaptive controllers with and without positivity condition." *Proceedings of the 23rd IEEE Conference on Decision and Control*, pp. 58–63. Las Vegas, Nevada.

The idea of conditional updating and projection of estimates into a bounded range is also treated in Egardt (1979). Conditional updating is also discussed in:

Peterson, B. B., and K. S. Narendra, 1982. "Bounded error adaptive control." *IEEE Trans. Automat. Contr.* **AC-27**: 1161–1168.

An elegant formalism for the growth-rate estimates in Lemma 6.2 is found in:

Narendra, K. S., A. M. Annaswamy, and R. P. Singh, 1985. "A general approach to the stability analysis of the adaptive systems." *Int. J. Contr.* **AC-41**: 193–216.

Proof of convergence for the original self-tuner based on recursive least-squares estimation and minimum-variance control is found in

Guo, L., and H.-F. Chen, 1991. "The Åström-Wittenmark self-tuning regulator revisited and ELS-based adaptive trackers." *IEEE Trans. Automat. Contr.* **AC-36**: 802–812.

Chen, H.-F., and L. Guo, 1991. *Identification and Stochastic Adaptive Control.* Boston: Birkhäuser.

The method of averaging to investigate nonlinear oscillations was developed by:

Krylov, A. N., and N. N. Bogoliubov, 1937. *Introduction to Non-linear Mechanics* (English translation 1943). Princeton, N.J.: Princeton University Press.

A simple presentation of the key ideas is given in:

Minorsky, N., 1962. *Nonlinear Oscillations.* Princeton, N.J.: Van Nostrand.

More detailed treatments are given in:

Hale, J. K., 1963. *Oscillations in Nonlinear Systems.* New York: McGraw-Hill.

Hale, J. K., 1969. *Ordinary Differential Equations.* New York: Wiley-Interscience.

Arnold, V. I., 1983. *Geometrical Methods in the Theory of Ordinary Differential Equations.* New York: Springer-Verlag.

Guckenheimer, J., and P. Holmes, 1983. *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields.* Berlin: Springer-Verlag.

Sastry, S., and M. Bodson, 1989. *Adaptive Control: Stability, Convergence, and Robustness.* Englewood Cliffs, N.J.: Prentice-Hall.

Many results on classical stability theory for ordinary differential equations are found in:

Bellman, R., 1953. *Stability Theory of Differential Equations.* New York: McGraw-Hill.

The example of nonrobustness in Example 6.8 is based on:

Rohrs, C., L. S. Valavani, M. Athans, and G. Stein, 1985. "Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics." *IEEE Trans. Automat. Contr.* **AC-30**: 881–889.

This initiated the discussion of the robustness problem. The analysis in Sections 6.6 and 6.7 is largely based on:

Åström, K. J., 1983. "Analysis of Rohr's counterexample to adaptive control." *Proceedings of the 22nd IEEE Conference on Decision and Control,* pp. 982–987. San Antonio, Texas.

Åström, K. J., 1984. "Interactions between excitation and unmodeled dynamics in adaptive control." *Proceedings of the 23rd IEEE Conference on Decision and Control,* pp. 1276–1281. Las Vegas, Nevada.

The idea of introducing leakage is found in:

Ioannou, P. A., and P. V. Kokotovic, 1983. *Adaptive Systems with Reduced Models.* New York: Springer-Verlag.

The idea of normalization was suggested by Praly. See, for example:

Praly, L., 1986. "Global stability of a direct adaptive control scheme with respect to a graph topology." In *Adaptive and Learning Systems: Theory and Applications,* ed. K. S. Narendra. New York: Plenum Press.

It is further explored in:

Narendra, K. S., and A. M. Annaswamy, 1987. "A new adaptive law for robust adaptation without persistent excitation." *IEEE Trans. Automat. Contr.* **AC-32**: 134–145.

Further discussion of robustness is given in:

Anderson, B. D. O., R. R. Bitmead, C. R. Johnson, Jr., P. V. Kokotovic, R. L. Kosut, I. M. Y. Mareels, L. Praly, and B. D. Riedle, 1986. *Stability of Adaptive Systems: Passivity and Averaging Analysis.* Cambridge, Mass.: MIT Press.

Goodwin, G. C., D. J. Hill, D. Q. Mayne, and R. H. Middleton, 1986. "Adaptive robust control. Convergence, stability and performance." *Proceedings of the 25th IEEE Conference on Decision and Control,* pp. 468–473. Athens, Greece.

Kreisselmeier, G., and B. D. O. Anderson, 1986. "Robust model reference adaptive control." *IEEE Trans. Automat. Contr.* **AC-31**: 127–133.

Ortega, R., and Y. Tang, 1989. "Robustness of adaptive controllers—A survey." *Automatica* **25**: 651–677.

Ydstie, B. E., 1992. "Transient performance and robustness of direct adaptive control." *IEEE Trans. Automat. Contr.* **AC-37**: 1091–1105.

Stochastic averaging was introduced in:

Ljung, L., 1977a. "Analysis of recursive stochastic algorithms." *IEEE Trans. Automat. Contr.* **AC-22**: 551–575.

The ordinary differential equations associated with a discrete time estimation problem were derived. This particular form of averaging is called the ODE method. Extensive applications of the method are given in:

Ljung, L., and T. Söderström, 1983. *Theory and Practice of Recursive Identification.* Cambridge, Mass.: MIT Press.

More recent proofs of the method are found in:

Kushner, H., 1984. *Approximation and Weak Convergence Methods of Random Processes.* Cambridge, Mass.: MIT Press.

Kushner, H., and D. Clark, 1978. *Stochastic Approximation Methods for Constrained and Unconstrained Systems,* Applied Mathematical Science Series 26. Berlin: Springer-Verlag.

Kushner, H., and A. Schwartz, 1984. "An invariant measure approach to the convergence of stochastic approximations with state dependent noise." *SIAM J. on Control and Optimization* **22**: 13–27.

Metivier, M., and P. Priouret, 1984. "Applications of a Kushner and Clark lemma to general classes of stochastic algorithms." *IEEE Trans. Information Theory* **IT-30**: 140–151.

An accessible account is also given in:

Kumar, P. R., and P. Varaiya, 1986. *Identification and Adaptive Control.* Englewood Cliffs, N.J.: Prentice-Hall.

Stochastic averaging was applied to the self-tuning regulator based on least-squares estimation and minimum-variance control in:

Ljung, L., 1977b. "On positive real transfer function and convergence of some recursive schemes." *IEEE Trans. Automat. Contr.* **AC-22**: 539–551.

More details about Example 6.14 are given in:

Åström, K. J., and B. Wittenmark, 1985. "The self-tuning regulators revisited." *Preprints of the 7th IFAC Symposium on Identification and System Parameter Estimation,* pp. xxv–xxxiii. York, U.K.

Conditions for local stability of the equilibrium are given in:

Holst, J., 1979. "Local convergence of some recursive stochastic algorithms." *Preprints of the 5th IFAC Symposium on Identification and System Parameter Estimation,* pp. 1139–1146. Darmstadt, Germany.

Analysis of stability in self-tuning regulators based on Lyapunov theory is given in:

Johansson, R., 1988. "Stochastic stability of direct adaptive control." Report TFRT-7377, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

The fact that rapid sampling may create zeros of the pulse transfer function outside the unit disc is discussed in:

Åström, K. J., P. Hagander, and J. Sternby, 1984. "Zeros of sampled systems." *Automatica* **20**: 31–38.

Work on universal stabilizers was initiated by a discussion of whether Assumption A4 of Theorem 6.7 is necessary. See:

Morse, A. S., 1983. "Recent problems in parameter adaptive control." In *Outils et Modèles Mathematiques pour l'Automatique, l'Analyse de Systèmes et le Traitement du Signal,* ed. I. D. Landau, vol 3, pp. 733–740. Paris: Editions du CNRS.

The problem was solved for scalar systems in:

Nussbaum, R. D., 1983. "Some remarks on a conjecture in parameter adaptive control." *Syst. Contr. Lett.* **3**: 243–246.

Universal stabilizers for multivariable systems are discussed in:

Mårtensson, B., 1985. "The order of any stabilizing regulator is sufficient a priori information for adaptive stabilization." *Syst. Contr. Lett.* **6**(2): 87–91.

Mårtensson showed (to summarize roughly) that the order of a stabilizing controller is the only information required for adaptive stabilization of a multivariable system.

CHAPTER 7

STOCHASTIC ADAPTIVE CONTROL

7.1 INTRODUCTION

In earlier chapters the adaptive control problem was approached from a heuristic point of view. The unknown parameters of the process or the regulator were estimated by using real-time estimation, and the estimated parameters were then used as if they were the true ones. The uncertainties of the parameter estimates were not taken into account in the design. This procedure gives a *certainty equivalence* controller. The model-reference adaptive controllers and the self-tuning regulators have been derived under the assumption that the parameters are constant but unknown. When the process parameters are constant, the estimation routines usually are such that the uncertainties decrease rapidly after the estimation is started. However, the uncertainties can be large at the startup or if the parameters are changing. In such cases it may be important to let the control law be a function of the parameter estimates as well as of the uncertainties of the estimates.

It would be appealing to formulate the adaptive control problem from a unified theoretical framework. This can be done by using nonlinear stochastic control theory, in which the process, its parameters, and the environment are described by using a stochastic model. The difference compared with the treatment in the previous chapters is that the parameters of the process also are described by using a stochastic model. The criterion is formulated so as to minimize the expected value of a loss function. It is difficult to find the controller that minimizes the expected loss function. Conditions for the existence of an optimal controller are not known. However, under the condition that a solution exists, it is possible to derive a functional equation by using dynamic program-

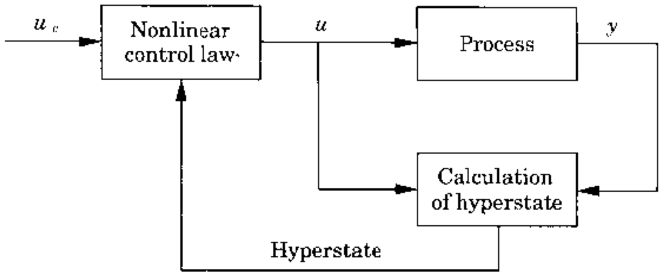

Figure 7.1 Block diagram of an adaptive regulator obtained from stochastic control theory.

ming. This equation, called the *Bellman equation*, can be solved numerically only in very simple cases. The structure of the optimal regulator is shown in Fig. 7.1. The controller is composed of two parts: an estimator and a feedback regulator. The estimator generates the conditional probability distribution of the state given the measurements. This distribution is called the *hyperstate* of the problem. The feedback regulator is a nonlinear function that maps the hyperstate into the space of control variables.

The structural simplicity of the solution is obtained at the price of introducing the hyperstate, which can be a quantity of very high dimension. Notice that the structure is similar to that of the self-tuning regulator. The self-tuning regulator can be regarded as an approximation; the conditional probability distribution is replaced by a distribution with all mass at the conditional mean value. In Fig. 7.1 there is no distinction between the parameters and the other state variables of the process. The regulator can therefore handle very rapid parameter variations. Furthermore, the averaging methods based on separation of the states of the process and the parameters (used in Chapter 6) cannot be used to analyze the system. The optimal control law has an interesting property. The control attempts to drive the output to the desired value, but it will also introduce perturbations when the estimates are uncertain. This will improve the estimates and the future control. The optimal controller achieves a correct balance between maintaining good control and small estimation errors. This is called *dual control*.

The chapter is organized in the following way. The idea with multistep decision problems is introduced in Section 7.2, where the two-armed bandit problem is introduced. A general stochastic adaptive control problem is formulated in Section 7.3, and Section 7.4 gives the derivation of the Bellman equation. The consequences of the structure of the solution are discussed, and the dual property is analyzed. Different ways to approximate the dual controller are discussed in Section 7.5. However, only very simple examples of dual controllers can be solved numerically, but the solutions give some useful indications of how suboptimal controllers can be constructed. Some examples

are given in Section 7.6, and the stochastic adaptive approach is summarized in Section 7.7.

7.2 MULTISTEP DECISION PROBLEMS

The idea of decision under uncertainty is discussed in this section. There are many situations in which decisions must be taken despite uncertainties about the processes or the statistics. One example is route planning, in which the traffic will influence the time it takes to get from one point to another. Another example is testing of medical drugs. In investigating the effect of a new drug, it is necessary to plan the test, but it is also important to have the possibility to go back to a standard procedure if the patient is not responding well to the new treatment. The characteristic features of these types of problems are that there are uncertainties about the possible outcome of different control actions. Further, there is a sequence of control actions to be taken. At each time, feedback is used to update or change the procedure. One of the first stochastic adaptive problems of this kind that was solved can be represented by the classical *two-armed bandit (TAB) problem*. This is a typical problem of sequential design of statistical experiments.

The TAB problem can be described in the following way. A player is faced with two slot machines, I and II. If machine I is played the gain is one unit with probability p; machine II gives a gain of one unit with probability q. In the simplest case, p is known and q is unknown and is chosen before each game of length N according to a given probability distribution. During the game the unknown quantity q has to be estimated, and the player must decide at each step which machine to play to maximize the total gain of each game of N plays.

The two-armed bandit problem can be used to illustrate the essential ideas of multistep decision problems. One strategy that can be used is open-loop control, that is, the control sequence is chosen without any measurements being made. The decision is taken with respect to the *a priori* knowledge about p and the distribution of q. In the TAB case, machine I should be played if p is larger than the mean value of q; otherwise, machine II should be played. A second strategy is what is called *open-loop optimal feedback (OLOF) control*. This controller is derived by maximizing the multistep gain function at each step under the assumption that no further measurements will be available, that is, an open-loop control sequence is determined. The first step in the control sequence is then used, and the performance of the system is measured. On the basis of the new information (feedback) a new maximization is done (compare the receding horizon controller in Chapter 4). The first step in the OLOF control is thus the same as the first step in the open-loop control. The measurements are thus in the TAB problem used to update the estimate of the unknown probability q.

To find the optimal solution to the TAB problem, it is possible to use dynamic programming to derive the optimal strategy that maximizes the expected gain depending on the outcome of previous plays in the game. How this is done is shown for a more general case in Section 7.4. The optimal strategy for a simple TAB problem is illustrated in the following example adopted from Yakowitz (1969).

EXAMPLE 7.1 Two-armed bandit problem

Assume that $p = 0.6$ and that q is uniformly distributed over the interval $[0, 1]$. If machine I is played all the time, the expected gain is 0.6 per play; if machine II is played all the time, the expected gain per play is 0.5. The open-loop strategy then suggests that machine I should be played all the time. However, for each game of length N there is a probability that the q has a larger value than 0.6. If infinitely many plays are available, the player can play machine II, estimate q, and then decide which machine to play for the rest of the plays.

To determine the profit of knowledge of q, assume that the player is told the value of q before each game. The player's optimal strategy is then to play the machine having the highest probability. In this case the expected gain per play is $E\{\max(p, q)\} = 0.68$. This means that the expected gain can be increased by 13% compared with the open-loop strategy if q is estimated. Table 7.1 shows the average gain per play for different values of N. As the number of plays increases, the gain will approach the maximum value 0.68. Relatively many plays are needed to get close to the optimum.

Figure 7.2 shows the state transition diagram for the optimal strategy when $N = 6$. The player is initially in the state (0, 0) and starts to play machine II to find out if machine II has better winning probability than machine I. The numbers in the circles indicate the number of times machines I and II, respectively, have given a gain of one unit. In states (0, 0) and (0, 1) there will be a switch to machine I after the player loses once; after state (0, 2) the optimal strategy allows one loss before switching to machine I. □

Table 7.1 Average gain per play for different values of N for the two-armed bandit problem in Example 7.1.

N	Gain per play
6	0.62
10	0.64
25	0.655
100	0.6676
500	0.6755
1000	0.6773

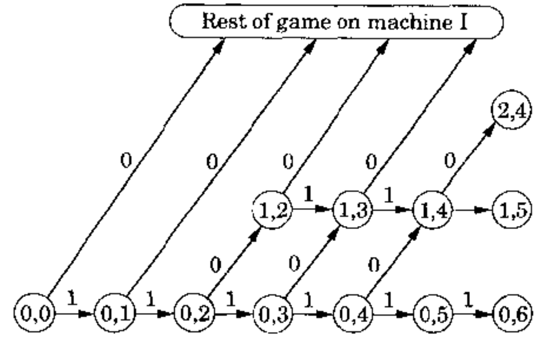

Figure 7.2 The optimal strategy for the two-armed bandit problem in Example 7.1 when $N = 6$.

7.3 THE STOCHASTIC ADAPTIVE PROBLEM

The stochastic adaptive control problem is formulated for a simple class of systems by giving the class of models, the criterion, and the admissible control strategies.

The Model

Consider the discrete-time, single-input, single-output system

$$y(t) + a_1(t)y(t-1) + \cdots + a_n(t)y(t-n) = b_0(t)u(t-1) + \cdots + b_{n-1}(t)u(t-n) + e(t) \qquad (7.1)$$

where y, u, and e are output, input, and disturbance, respectively. The noise sequence $\{e(t)\}$ is assumed to be Gaussian with zero mean and variance R_2. Further, it is assumed that $e(t)$ is independent of $y(t-1)$, $y(t-2)$, $\ldots$, $u(t-1)$, $u(t-2)$, $\ldots$, $a_i(t)$, $a_i(t-1)\ldots$, and $b_i(t)$, $b_i(t-1)$, $\ldots$. It is further assumed that $b_0(t) \neq 0$ and that the system is minimum-phase for all t. The time-varying parameters

$$x(t) = \begin{pmatrix} b_0(t) & \ldots & b_{n-1}(t) & a_1(t) & \ldots & a_n(t) \end{pmatrix}^T \qquad (7.2)$$

are modeled by a Gauss-Markov process, which satisfies the stochastic difference equation

$$x(t+1) = \Phi x(t) + v(t) \qquad (7.3)$$

where Φ is a known constant matrix and $\{v(t)\}$ is a sequence of independent, equally distributed normal vectors with zero mean value and known covariance

R_1. The initial state of the system in Eq. (7.3) is assumed to be normally distributed with mean value

$$Ex(0) = m \tag{7.4}$$

and covariance

$$\text{cov}\ \{x(0), x(0)\} = R_0 \tag{7.5}$$

It is assumed that $e(t)$ is independent of $v(t)$ and of $x(0)$.

The input-output relation of the system of Eq. (7.1) can be written in the compact form

$$y(t) = \varphi^T(t-1)x(t) + e(t) \tag{7.6}$$

where

$$\varphi^T(t-1) = \begin{pmatrix} u(t-1) & \ldots & u(t-n) & -y(t-1) & \ldots & -y(t-n) \end{pmatrix} \tag{7.7}$$

The model is thus defined by Eqs. (7.3) and (7.6).

The Criterion

It is assumed that the purpose of the control is to keep the output of the system as close as possible to a known reference value trajectory $u_c(t)$. The deviation is measured by the criterion

$$J_N = E\left\{\frac{1}{N}\sum_{t=1}^{N}(y(t) - u_c(t))^2\right\} \tag{7.8}$$

where E denotes mathematical expectation. This is called an *N-stage criterion*. The loss function should be minimized with respect to $u(0), u(1), \ldots, u(N-1)$. The controller obtained for $N = 1$ is sometimes called a *myopic controller*, since it is short-sighted and looks only one step ahead. The minimizing controller will be very different if $N = 1$ or if N is large.

Admissible Control Strategies

To specify the problem completely, it is necessary to define the admissible control strategies. A control strategy is admissible if $u(t)$ is a function of all outputs observed up to and including time t, that is, $y(t), y(t-1), \ldots$ all applied control signals $u(t-1), \ldots$ and the *a priori* data. Let $\mathcal{Y}_t$ denote all values of the output up to and including $y(t)$ or, more precisely, the σ-algebra generated by $y(t), \ldots, y(0)$ and $x(0)$.

Discussion of the Problem Formulation

To get a reasonable problem, it is assumed that the noise in Eq. (7.1) is of least-squares type, that is, $C(q) = q^n$. Further, there is no extra time delay in the system. In the formulation it has been assumed that the measurements $y(t)$ are obtained at each sampling interval. It is possible to define other control problems leading to other controllers by changing the way in which the future measurements will become available. The realism of the assumption that Φ is known in Eq. (7.3) is open to question. The case $\Phi = I$ can, however, be used as a generic case to study the dual control problem.

The process of Eq. (7.1) is a nonlinear model, since the parameters as well as the old inputs and outputs are the states of the system. Notice for instance that the distributions of the parameters and the disturbances are Gaussian but $y(t)$ is not Gaussian. The problem could also be phrased in more general terms by assuming that both the model and the criterion are general nonlinear functions. In this chapter we consider the special case defined by Eqs. (7.1) and (7.8) to illustrate the ideas and the difficulties with the stochastic adaptive approach.

7.4 DUAL CONTROL

We now analyze the problem formulated in Section 7.3. The problem of estimating the parameters of Eq. (7.1) is first considered. The control problem is first solved for the case in which the parameters are known. The problem is then solved for the case in which $N = 1$ in the criterion of Eq. (7.8). The solution of the complete problem is finally discussed. The control problem is solved by using dynamic programming.

The Estimation Problem

To solve the dual control problem, it is necessary to be able to evaluate the influence of the control signal on the future outputs and to estimate and predict the behavior of the stochastic parameters. The estimation problem is defined so as to compute the conditional probability distribution of the parameters, given the measured data.

The system is written in a standard state space form, using Eqs. (7.3) and (7.6). The conditional distribution of $x(t+1)$, given $\mathcal{Y}_t$, is given by the following theorem.

THEOREM 7.1 Conditional distribution of the states

Consider the model of Eq. (7.3) with the output defined by Eq. (7.6), where $e(t)$ and $v(t)$ are independent zero mean Gaussian variables with covariances

R_2 and R_1, respectively. The initial state of the system is given by Eqs. (7.4) and (7.5).

The conditional distribution of $x(t)$, given $\mathcal{Y}_{t-1}$, is Gaussian with mean $\hat{x}(t)$ and covariance $P(t)$ satisfying the difference equations

$$\begin{aligned}\hat{x}(t+1) &= \Phi\hat{x}(t) + K(t)\left(y(t) - \varphi^T(t-1)\hat{x}(t)\right)\\ P(t+1) &= \left(\Phi - K(t)\varphi^T(t-1)\right)P(t)\Phi^T + R_1 \\ K(t) &= \Phi P(t)\varphi(t-1)\left(R_2 + \varphi^T(t-1)P(t)\varphi(t-1)\right)^{-1}\end{aligned} \tag{7.9}$$

with the initial conditions

$$\hat{x}(0) = m$$
$$P(0) = R_0$$

Furthermore, the conditional distribution of $y(t)$, given $\mathcal{Y}_{t-1}$, is Gaussian with mean value

$$m_y(t) = \varphi^T(t-1)\hat{x}(t)$$

and covariance

$$\sigma_y^2(t) = R_2 + \varphi^T(t-1)P(t)\varphi(t-1)$$

Proof: If $\varphi(t-1)$ is a known time-varying vector, then the theorem is identical to the Kalman filtering theorem, which can be found in standard textbooks on stochastic control. Going through the details of the proof of the Kalman filtering theorem, we find that it is still valid, since $\varphi(t-1)$ is a function of $\mathcal{Y}_{t-1}$. In other words, the vector $\varphi(t-1)$ is not known in advance, but it is known when it is needed in the computations. □

Remark. Notice that the conditional distribution of $y(t)$, given $\mathcal{Y}_{t-1}$, is Gaussian even if $y(t)$ is not Gaussian. □

The estimation problem is thus easily solved for the model structure chosen. The conditional distribution of the state of the system is called the *hyperstate*. The distribution is Gaussian in the problem under consideration. It is then sufficient to consider the mean and covariance of $x(t)$. Further, some of the old inputs and outputs must be stored to compute the distribution defined in Eqs. (7.9). In the problem under consideration the hyperstate is finite-dimensional and can be characterized by the triple

$$\xi(t) = \begin{pmatrix} \tilde{\varphi}(t-1) & \hat{x}(t) & P(t) \end{pmatrix} \tag{7.10}$$

where

$$\tilde{\varphi}^T(t-1) = \begin{pmatrix} 0 & u(t-2) & \dots & u(t-n) & -y(t-1) & \dots & -y(t-n) \end{pmatrix} \tag{7.11}$$

The vector $\tilde{\varphi}^T(t-1)$ is the same as $\varphi^T(t-1)$, except that $u(t-1)$ is replaced by a zero. The updating of the hyperstate is given by Theorem 7.1 and the definition of $\tilde{\varphi}^T(t-1)$. In the general case the conditional probability distribution is not Gaussian. This will considerably increase the computational difficulties and the storage requirements.

Systems with Known Parameters

If the parameters of the system of Eq. (7.1) are known, it is easy to determine the optimal feedback. The vector $\tilde{\varphi}^T$ defined by Eq. (7.11) is used to show the dependence of $u(t)$:

$$\begin{aligned}y(t+1) &= \varphi^T(t)x(t+1) + e(t+1)\\ &= b_0(t+1)u(t) + \tilde{\varphi}^T(t)x(t+1) + e(t+1)\end{aligned}$$

The optimal feedback when $b_0(t+1)$ and $x(t+1)$ are known is then given by

$$u(t) = \frac{u_c(t+1) - \tilde{\varphi}^T(t)x(t+1)}{b_0(t+1)} \tag{7.12}$$

Notice that $\tilde{\varphi}(t)$ is a function of the admissible data. This controller gives

$$y(t+1) = u_c(t+1) + e(t+1)$$

and it minimizes Eq. (7.8), since $e(t+1)$ is independent of $\mathcal{Y}_t$ and $u(t)$. The minimal loss is given by

$$\min J_N = R_2$$

Notice that it is necessary to assume that $b_0(t+1) \neq 0$ and that the system is minimum-phase at every instant of time. The control signal may otherwise be unbounded.

Certainty Equivalence Control

When the parameters $x(t+1)$ are not known, it can be tempting to replace Eq. (7.12) with

$$u(t) = \frac{u_c(t+1) - \tilde{\varphi}^T(t)\hat{x}(t+1)}{\hat{b}_0(t+1)} \tag{7.13}$$

The true parameter values are replaced by the expected values, given $\mathcal{Y}_t$. The controller of Eq. (7.13) is called the *certainty equivalence controller*. Certainty equivalence control is the strategy used in the self-tuning regulators in Chapters 3 and 4 and in the model-reference adaptive systems in Chapter 5. In these controllers it was also necessary to ensure that $\hat{b}_0 \neq 0$.

Cautious Control

We now consider the special case in which $N = 1$ in Eq. (7.8). According to Theorem 7.1 the conditional distribution of $y(t+1)$, given $\mathcal{Y}_t$, is Gaussian with mean $\varphi^T(t)\hat{x}(t+1)$ and covariance $R_2 + \varphi^T(t)P(t+1)\varphi(t)$. Then

$$\begin{aligned}
E&\left\{(y(t+1) - u_c(t+1))^2 \mid \mathcal{Y}_t\right\} \\
&= \left(\varphi^T(t)\hat{x}(t+1) - u_c(t+1)\right)^2 + \varphi^T(t)P(t+1)\varphi(t) + R_2 \\
&= \left(\tilde{\varphi}^T(t)\hat{x}(t+1) + \hat{b}_0(t+1)u(t) - u_c(t+1)\right)^2 \\
&\quad + \tilde{\varphi}^T(t)P(t+1)\tilde{\varphi}(t) + u^2(t)p_{b_0}(t+1) \\
&\quad + 2u(t)\tilde{\varphi}^T(t)P(t+1)\ell + R_2
\end{aligned} \tag{7.14}$$

The first equality is obtained by using the standard formula that

$$E(\zeta^2) = m^2 + p$$

when ζ is a Gaussian variable with mean m and variance p. The column vector ℓ selects the first column of the matrix $P(t)$, that is,

$$\ell^T = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}$$

Further, p_{b_0} is the covariance of the parameter estimate $\hat{b}_0$. Equation (7.14) is quadratic in $u(t)$. Minimization of Eq. (7.14) with respect to $u(t)$ gives the admissible one-step optimal controller

$$u(t) = \frac{\hat{b}_0(t+1)u_c(t+1) - \tilde{\varphi}^T(t)\left(\hat{b}_0(t+1)\hat{x}(t+1) + P(t+1)\ell\right)}{\hat{b}_0^2(t+1) + p_{b_0}(t+1)} \tag{7.15}$$

The minimum value of the loss function is

$$\begin{aligned}
\min_{u(t)} E&\left\{(y(t+1) - u_c(t+1))^2 \mid \mathcal{Y}_t\right\} \\
&= \left(\tilde{\varphi}^T(t)\hat{x}(t+1) - u_c(t+1)\right)^2 + R_2 + \tilde{\varphi}^T(t)P(t+1)\tilde{\varphi}(t) \\
&\quad - \frac{\left(\hat{b}_0(t+1)u_c(t+1) - \tilde{\varphi}^T(t)\left(\hat{b}_0(t+1)\hat{x}(t+1) + P(t+1)\ell\right)\right)^2}{\hat{b}_0^2(t+1) + p_{b_0}(t+1)}
\end{aligned} \tag{7.16}$$

The *one-step-ahead controller*, or *cautious controller*, of Eq. (7.15) differs from Eq. (7.13) because the uncertainties of the parameter estimates are also taken into account. The controller becomes cautious when the estimates are uncertain. Notice that the cautious controller of Eq. (7.15) reduces to the certainty equivalence controller of Eq. (7.13) when $P(t+1) = 0$.

EXAMPLE 7.2 Integrator with time-varying gain

Consider an integrator in which the gain is changing. Let the process be described by

$$y(t) - y(t-1) = b(t)u(t-1) + e(t)$$

where

$$b(t+1) = \varphi_b b(t) + R_1 v(t)$$

The errors e and v are zero-mean Gaussian white noise with the standard deviations R_2 and 1, respectively. Further, it is assumed that $u_c = 0$. The certainty equivalence controller is given by

$$u(t) = -\frac{1}{\hat{b}(t+1)}y(t)$$

and the cautious controller is

$$u(t) = -\frac{\hat{b}(t+1)}{\hat{b}^2(t+1) + p_b(t+1)}y(t)$$

The gain in the cautious controller has been reduced by a factor

$$\frac{\hat{b}^2}{\hat{b}^2 + p_b}$$

compared with the certainty equivalence controller. Notice that the gain approaches zero when the uncertainty increases. □

Multistep Optimization

The general multistep optimization problem can be solved by using dynamic programming. The fact that the conditional distributions are Gaussian will simplify the problem.

It follows from a fundamental result of stochastic control theory (see Åström (1970), Lemma 8:3.2) that

$$\begin{aligned}
\min_{u(t-1)\dots u(N-1)} &E\left\{\sum_{k=t}^{N}(y(k) - u_c(k))^2\right\} \\
&= E_{\mathcal{Y}_{t-1}}\left(\min E\left\{\sum_{k=t}^{N}(y(k) - u_c(k))^2 \,\Big|\, \mathcal{Y}_{t-1}\right\}\right)
\end{aligned}$$

and it is assumed that the minimum exists. $E(\cdot \mid \mathcal{Y}_{t-1})$ is a function of the hyperstate of Eq. (7.10) and t. Define

$$V(\xi(t), t) = \min_{u(t-1)\dots u(N-1)} E\left\{\sum_{k=t}^{N}(y(k) - u_c(k))^2 \,\Big|\, \mathcal{Y}_{t-1}\right\}$$

$V(\xi(t), t)$ can be interpreted as the minimum expected loss for the remaining part of the control horizon given the data up to $t-1$.

Consider the situation at time $N-1$. When $u(N-1)$ is changed, only $y(N)$ will be influenced. This means that we have the same situation as for the one-step minimization. From Eq. (7.16) we get

$$\begin{aligned} &V(\xi(N), N) \\ &= \left(\tilde{\varphi}^T(N-1)\hat{x}(N) - u_c(N)\right)^2 + R_2 + \tilde{\varphi}^T(N-1)P(N)\tilde{\varphi}(N-1) \\ &\quad - \frac{\left(\hat{b}_0(N)u_c(N) - \tilde{\varphi}^T(N-1)\left(\hat{b}_0(N)\hat{x}(N) + P(N)\ell\right)\right)^2}{\hat{b}_0^2(N) + p_{b_0}(N)} \end{aligned}$$

At time $N-1$ we get

$$\begin{aligned} &V(\xi(N-1), N-1) \\ &= \min_{u(N-2)} E\left\{(y(N-1) - u_c(N-1))^2 + V(\xi(N), N)\,\middle|\,\mathcal{Y}_{N-2}\right\} \end{aligned}$$

Notice that the minimization is done only over $u(N-2)$, since $u(N-1)$ was eliminated in the previous minimization. This recursively defines the loss at time $N-1$, which then can be used for iteration backwards one more step of time, and so on. This dynamic programming procedure leads to a recursive equation, which defines the minimum expected loss. At time t we get

$$V(\xi(t), t) = \min_{u(t-1)} E\left\{(y(t) - u_c(t))^2 + V(\xi(t+1), t+1)\,|\,\mathcal{Y}_{t-1}\right\} \qquad (7.17)$$

This functional equation is called the *Bellman equation* of the problem. The simplicity of the form of Eq. (7.17) is misleading. The equation cannot be solved analytically, but it requires extensive numerical computations to get the solution even for very simple problems.

The first term on the right-hand side of Eq. (7.17) can be evaluated in the same way as in the one-step minimization. The second term causes the difficulties in the optimization, since we have to evaluate

$$E\left\{V(\xi(t+1), t+1)\,\middle|\,\mathcal{Y}_{t-1}\right\}$$

The average with respect to the distribution of $y(t)$, given $\mathcal{Y}_{t-1}$, must be computed. According to Theorem 7.1 this distribution is Gaussian with mean $m_y(t)$ and variance $\sigma_y^2(t)$. This gives

$$\begin{aligned} &E\left\{V(\xi(t+1), t+1)\,\middle|\,\mathcal{Y}_{t-1}\right\} \\ &= \frac{1}{\sigma_y\sqrt{2\pi}} \int_{-\infty}^{\infty} V(\tilde{\varphi}(t), \hat{x}(t+1), P(t+1), t+1)\, e^{-(s-m_y)^2/(2\sigma_y^2)}\, ds \qquad (7.18) \end{aligned}$$

where

$$\begin{aligned} \hat{x}(t+1) &= \Phi\hat{x}(t) + K(t)\left(s - \varphi^T(t-1)\hat{x}(t)\right) \\ P(t+1) &= \left(\Phi - K(t)\varphi^T(t-1)\right)P(t)\Phi^T + R_1 \\ K(t) &= \Phi P(t)\varphi(t-1)/\sigma_y^2(t) \\ \sigma_y^2(t) &= R_2 + \varphi^T(t-1)P(t)\varphi(t-1) \\ \tilde{\varphi}_1(t) &= u(t-1) \\ \tilde{\varphi}_i(t) &= \tilde{\varphi}_{i-1}(t-1) \qquad i = 2, \ldots, n, n+2, \ldots, 2n \\ \tilde{\varphi}_{n+1}(t) &= s \end{aligned}$$

These equations, together with Eq. (7.18), can be used to compute recursively the control signal and the loss as functions of the hyperstate. The control variable $u(t-1)$ influences the immediate loss (i.e., the first term on the right-hand side of Eq. (7.17)). Notice that $u(t-1)$ also influences the expected future loss, since it influences $\varphi(t-1)$, which influences $\hat{x}(t+1)$, $P(t+1)$, and $\tilde{\varphi}^T(t)$. This means that the choice of the control signal $u(t-1)$ influences the immediate loss, the future parameter estimates, their accuracy, and also the future values of the output signal. The optimal controller is a *dual controller*. It makes a compromise between the control action and the probing action.

The probing action will add an active learning feature to the controller, in contrast to the cautious and certainty equivalence controllers, in which the learning is "accidental." The optimal feedback will generate control actions that will improve the accuracy of the future estimates at the expense of the short-term loss. The cautious controller obtained when $N = 1$ will not benefit if probing is introduced; it only tries to make the loss as small as possible at the next instant of time.

Separation and Certainty Equivalence

The optimal one-step controller of Eq. (7.15) cannot be obtained by using the certainty equivalence principle, but the estimation and the control problems can be separated. As was mentioned in Section 7.1, most adaptive controllers are based on the hypothesis that the certainty equivalence principle can be used. The derivations in this section show that the separation principle also can be used in the considered problem. However, the uncertainties must also be used in the computation of the control signal. It is thus of interest to investigate whether there are classes of systems for which the certainty equivalence and separation principles hold.

One case in which the certainty equivalence principle holds is the celebrated linear quadratic Gaussian case for known systems. For adaptive controllers there are very few cases to which the certainty equivalence principle is applicable. One exception is when the unknown parameters are stochastic variables that are independent between different sampling intervals. The certainty equivalence principle also holds for the stochastic linear quadratic problem formulation, when the process noise is white but not necessarily Gaussian and when the measurement noise is additive but not necessarily white.

The separation principle is valid for much more general cases. The cautious controller and the dual controller derived in this section are obtained by using separation.

Numerical Solution

Even in the simplest cases there is no analytic solution to the Bellman equation (Eq. 7.17). It is therefore necessary to resort to numerical solution. One iteration of Eq. (7.17) involves

- Discretization of the loss V in the variables of the hyperstate,
- Evaluation of the integral in Eq. (7.18) using a quadrature formula, and
- Minimization over $u(t-1)$ for each combination of the discretized hyperstate.

Both V and u are functions of the hyperstate, so the storage requirements increase rapidly when the order of the system increases. Assume that the dimension of the hyperstate is 2 and that each variable is discretized into ten steps. Thus the loss and control tables contain 100 values each. Let the hyperstate have dimension 6, and let each variable be discretized in ten steps. The dimension of the loss and control tables is then 10^6 each. This means that only very simple problems have been solved numerically because of the "curse of dimensionality."

Discontinuity of the Control Signal

A feature of the optimal solution is that the control law can become discontinuous in situations such as that shown in Fig. 7.3. The figure shows the loss function as a function of the control signal for three different but close values of the hyperstate. If there are several local minima, the control signal can become

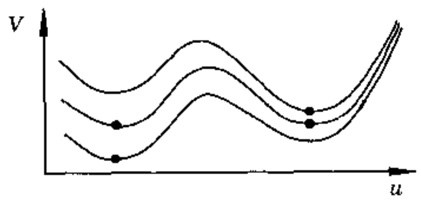

Figure 7.3 Illustration of how several local minima of the loss function can give a discontinuity in the control signal. The global minima are marked with dots. On the middle curve the local minima have the same value.

discontinuous when the global minimum changes from one local minimum to another. This can be interpreted as a change of mode for the controller. For instance, the controller may introduce probing to increase the knowledge of the unknown parameters.

7.5 SUBOPTIMAL STRATEGIES

The optimal multistep dual controller derived in Section 7.4 is of little practical use, because the numerical computations limit its applicability. The dual structure of the controller is very important, however. Many ways to make practical approximations have been suggested; this section surveys some of the possibilities. The properties of the cautious controller are first investigated, and different ways to improve this controller are then discussed.

Cautious Controllers

Minimization over only one step leads to the one-step or cautious controller of Eq. (7.15). This controller takes the parameter uncertainties into account, in contrast to the certainty equivalence controller of Eq. (7.13). However, the gain of Eq. (7.15) will decrease if the variance of $\hat{b}_0$ increases. This will give less information about b_0 in the next step, and the variance will increase further. The controller is then caught in a vicious circle, and the magnitude of the control signal becomes very small. This is called the *turn-off phenomenon.*

EXAMPLE 7.3 Turn-off

Consider the integrator with unknown gain in Example 7.2 with $R_1 = 0.09$ and $\varphi_b = 0.95$. Figure 7.4 shows a representative simulation of the cautious controller. The control signal is small for periods of time, and the variance of the estimated gain increases during the turn-off. After some time the control activity suddenly starts again. □

The turn-off will generally start when the control signal is small and when the parameter b_0 is small. The problem with turn-off makes the cautious controller unsuitable for control of systems with quickly varying parameters. The cautious controller can be useful if the parameters of the process are constant or almost constant, but the certainty equivalence controller with some simple safety measures can often be used in such cases also.

Classification of Suboptimal Dual Controllers

The problem of turn-off has led to many suggestions of how to derive controllers that are simple but still have some dual features. Some ways are:

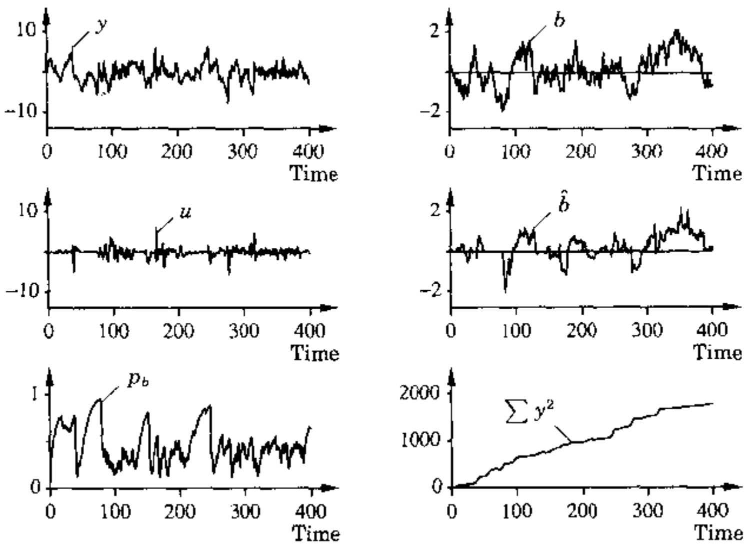

Figure 7.4 Representative simulation when an integrator is controlled by using a cautious controller. Turn-off occurs when the control signal is small.

- Adding perturbation signals to the cautious controller,
- Constraining the variance of the parameter estimates,
- Extensions of the loss function, and
- Serial expansion of the loss function.

Some of these modifications are now discussed.

Perturbation Signals

The turn-off is due to lack of excitation (compare Chapter 6). One way to increase the excitation is to add a perturbation signal. Pseudo-random binary sequences (PRBS) and white noise signals have been suggested. The perturbation can be added all the time or only when the variance is exceeding some limit. The addition of the extra signal will naturally increase the probing loss but may make it possible to improve the total performance.

Constrained One-Step Minimization

One class of suboptimal dual controllers is obtained by constrained one-step minimization. Suggested constraints are

- Limitation of the minimum value of the control signal and
- Limitation of the variance.

One method is to choose the control as

$$u(t) = \begin{cases} u_{\text{lim}} \cdot \operatorname{sign}(u_{\text{cautious}}) & \text{if } |u_{\text{cautious}}| < |u_{\text{lim}}| \\ u_{\text{cautious}} & \text{if } |u_{\text{cautious}}| \geq |u_{\text{lim}}| \end{cases}$$

This will give an extra probing signal if the cautious controller gives too small an input signal.

Different ways to constrain the minimization by using the P-matrix have been suggested. For instance, the one-step loss of Eq. (7.14) can be minimized under the constraint that

$$\operatorname{tr} P^{-1}(t+2) \geq M$$

P^{-1} is proportional to the information matrix. The constraint on the trace of P^{-1} means that the information about the parameters is always larger than some chosen value M. A similar approach is to constrain only the variance of $\hat{b}_0$ to

$$p_{b_0}(t+2) \leq \begin{cases} \gamma \hat{b}_0^2(t+2) & \text{if } p_{b_0}(t+1) \leq \hat{b}_0^2(t+1) \\ \alpha p_{b_0}(t+1) & \text{otherwise} \end{cases}$$

These modifications of the cautious controller have the advantage that the control signal can be easily computed, but the algorithms will contain application-dependent parameters that have to be chosen by the user. Finally, the approximations will not prevent the turn-off. The extra perturbation is not activated until the turn-off occurs.

Extensions of the Loss Function

An approach that is similar to constrained minimization is to extend the loss function to prevent the shortsightedness of the cautious controller. One obvious way is to try to solve the two-step minimization problem. The derivation in Section 7.4 shows that it is not possible to get an analytic solution when $N = 2$ in Eq. (7.8).

Another approach is to extend the loss function with a function of $P(t+2)$, which will reward good parameter estimates. The following loss function can be used:

$$\min_{u(t)} E\left\{ (y(t+1) - u_c(t+1))^2 + \rho f(P(t+2)) \middle| \mathcal{Y}_t \right\} \tag{7.19}$$

where ρ is a fixed parameter. Since the crucial parameter is b_0, we can use

$$f(P(t+2)) = p_{b_0}(t+2)$$

or

$$f(P(t+2)) = R_2 \frac{p_{b_0}(t+2)}{p_{b_0}(t+1)} \tag{7.20}$$

This leads to a loss function with two local minima; it is necessary to make a numerical search for the global minimum. It is possible to utilize the structure of the problem and make a serial expansion up to second order of the loss function. The expansion gives a simple noniterative suboptimal dual controller in which the increase in computations compared with a self-tuning or cautious regulator is very moderate.

Two similar approaches are to modify the loss functions to

$$\min_{u(t)} E\left\{ (y(t+1) - u_c(t+1))^2 - \rho \frac{\det P(t+1)}{\det P(t+2)} \,\Big|\, \mathcal{Y}_t \right\} \tag{7.21}$$

and

$$\min_{u(t)} E\left\{ (y(t+1) - u_c(t+1))^2 - \rho \varepsilon^2(t+1) \,\Big|\, \mathcal{Y}_t \right\} \tag{7.22}$$

respectively. The innovation $\varepsilon(t+1)$ is defined as

$$\varepsilon(t+1) = y(t+1) - \varphi^T(t)\hat{x}(t+1)$$

Both these loss functions lead to quadratic criteria that make it possible to derive simple analytic expressions for the control signal.

Serial Expansion of the Loss Function

The suboptimal dual controllers discussed above have been derived for the input-output model of Eq. (7.1). Suboptimal dual controllers have also been derived for state space models. One approach is to make an expansion of the loss function in the Bellman equation. Such an expansion can be done around the certainty equivalence or the cautious controllers. This approach has mainly been used when the control horizon N is rather short, usually less than 10. One reason is the quite complex computations that are involved.

Summary

There are many ways to make suboptimal dual controllers. Most of the approximations that are discussed start with the cautious controller and try to introduce some active learning. This can be done by including a term in the loss function that reflects the quality of the estimates. This term should also be a function of the control signal that is going to be determined. The suboptimal controllers should also be such that they can be used for higher-order systems without too much computation.

7.6 EXAMPLES

Some examples are used to illustrate the properties of the controllers discussed in this chapter.

EXAMPLE 7.4 Optimal dual controller

The first example is a numerically solved dual control problem from Åström and Helmersson (1982). Consider the integrator in Example 7.2. The gain is assumed to be constant but unknown, that is, $\varphi_b = 1$ and $R_1 = 0$. It is assumed that the parameter b is a random variable with a Gaussian prior distribution; the conditional distribution of b, given inputs and outputs up to time t, is Gaussian with mean $\hat{b}(t)$ and covariance $P(t)$. The hyperstate can then be characterized by the triple $(y(t), \hat{b}(t), P(t))$. The equations for updating the hyperstate are given by Eqs. (7.9).

Define the loss function

$$V_N = \min_u E\left\{ \sum_{k=t+1}^{t+N} y^2(k) \mid \mathcal{Y}_t \right\}$$

where $\mathcal{Y}_t$ denotes the data available at time t, that is, $\{y(t),\ y(t-1), \ldots\}$. By introducing the normalized variables

$$\eta = y/\sqrt{R_2} \qquad \beta = \hat{b}/\sqrt{P} \qquad \mu = -u\hat{b}/y$$

it can be shown that V_N depends on η and β only. Further introduce the normalized innovation

$$\varepsilon(t) = \frac{y(t+1) - y(t) - \hat{b}(t)u(t)}{R_2 + u(t)^2 P(t)}$$

For $R_2 = 1$ the Bellman equation for the problem can be written as

$$V_N(\eta, \beta) = \min_\mu U_N(\eta, \beta, \mu)$$

where

$$V_0(\eta, \beta) = 0$$

and

$$U_N(\eta, \beta, \mu) = 1 + \eta^2(1-\mu)^2 + \frac{\mu^2\eta^2}{\beta^2} - \int_{-\infty}^{\infty} V_{N-1}(\eta_p, \beta_p)\phi(s)\, ds$$

where ϕ is the normal probability density with zero mean and unit variance and

$$\eta_p = \eta - \mu\eta + s\sqrt{1 + \frac{\mu^2\eta^2}{\beta^2}}$$

$$\beta_p = \beta\sqrt{1 + \frac{\mu^2\eta^2}{\beta^2}} - \frac{\mu\eta}{\beta}s$$

Notice that η_p and β_p are the one-step-ahead predicted values of η and β. When the minimization is performed, the control law is obtained as

$$\mu_N(\eta, \beta) = \arg\min_\mu U_N(\eta, \beta, \mu)$$

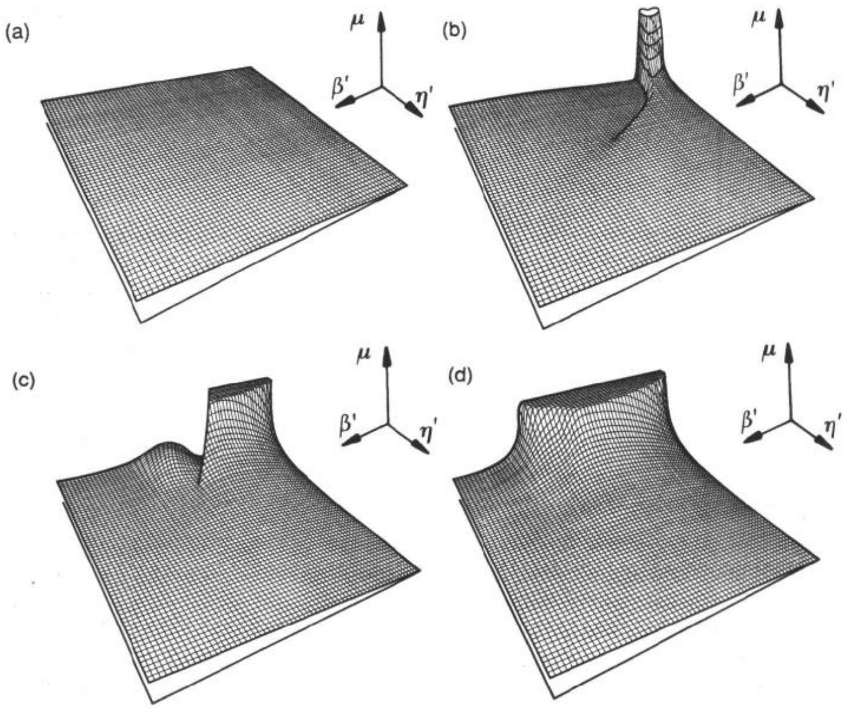

Figure 7.5 Illustration of the cautious control and dual control laws when (a) $N = 1$; (b) $N = 3$; (c) $N = 6$; and (d) $N = 31$. The control signal is shown as a function of $\eta' = \eta/(1+\eta)$ and $\beta' = \beta^2/(1+\beta^2)$. The control signal is limited, which explains the plateaus.

The minimization can be done analytically for $N = 1$, giving

$$\mu_1(\eta, \beta) = \arg\min \left(1 + \eta^2(1-\mu)^2 + \frac{\mu^2\eta^2}{\beta^2}\right) = \frac{\beta^2}{1+\beta^2}$$

The original variables give

$$u(t) = -\frac{1}{\hat{b}(t+1)} \cdot \frac{\hat{b}^2(t+1)}{\hat{b}^2(t+1) + P(t+1)}\, y(t)$$

This control law is the one-step control, or myopic control, derived in Example 7.2.

For $N > 1$ the optimization can no longer be done analytically. Instead, we have to resort to numerical calculations. Figure 7.5 shows the dual control laws obtained for different time horizons N. The discontinuity of the control law corresponds to the situation in which a probing signal is introduced to improve the estimates.

The certainty equivalence controller

$$u(t) = -y(t)/\hat{b}$$

can be expressed as

$$\mu = 1$$

in normalized variables. Notice that all control laws are the same for large β, that is, if the estimate is accurate. The optimal control law is close to the cautious control for large control errors. For estimates with poor precision and moderate control errors, the dual control gives larger control actions than the other control laws. The optimal dual controller has been computed on a Vax 11/780. The normalized variables η and β are discretized into 64 values each. The control table and the loss function table are thus of dimension 64 × 64. One iteration of the Bellman equation takes about 6 hours of CPU time. □

EXAMPLE 7.5 Probing

An interesting feature of the dual control law is that it behaves quite differently from the heuristic algorithms. The most significant feature is the probing that takes place to gain more information about the unknown parameters. The effect of probing is most significant when the output y is small. Probing can be illustrated by using the results of Example 7.4. Both the cautious and certainty equivalence control laws are continuous in y and zero for $y = 0$. However, the dual control law is very different. To show this, consider the control signal for $y = 0-$. Figure 7.6 shows the control signal for $y = 0-$ as a function of the normalized parameter precision, $\beta = \hat{b}/\sqrt{P}$, for different time horizons. All control laws give zero control signal when the parameter estimate

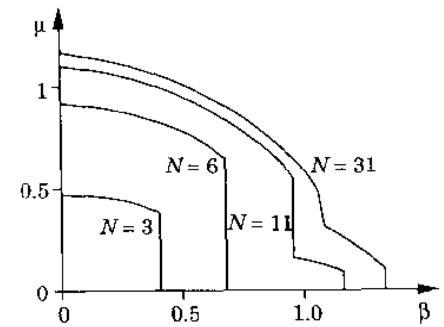

Figure 7.6 Control signal as a function of the normalized parameter precision β for optimal control laws for different time horizons.

is reasonably precise. However, for uncertain estimates, the control signal is different from zero, and the transition is discontinuous. This discontinuity can be used to define a probing zone. Notice that the probing zone increases with increasing time horizon. For $N = 31$, probing occurs when $\beta \le 1.3$, that is, when $\hat{b} \le 1.3\sqrt{P}$. □

EXAMPLE 7.6 Time-varying parameters

The system in Examples 7.2 and 7.3 will now be controlled by a suboptimal dual controller that minimizes Eq. (7.19), with $f(P(t+2))$ given by Eq. (7.20). (See Wittenmark and Elevitch (1985).) Figure 7.7 shows the same experiment using the same noise sequences as in Fig. 7.4. With the suboptimal dual controller there is no tendency toward turn-off. The simulation in Fig. 7.7 shows that the suboptimal dual controller is much better than the cautious controller. Comparisons using Monte Carlo simulations have also been done with the numerically computed optimal dual controller. The result is that the suboptimal dual controller is as good as the numerically computed optimal controller. A summary of some simulations is shown in Fig. 7.8, which shows mean values and standard deviations of the loss when the standard deviation of the parameter noise R_1 is changed. It is assumed that

$$\varphi_b = \sqrt{1 - R_1}$$

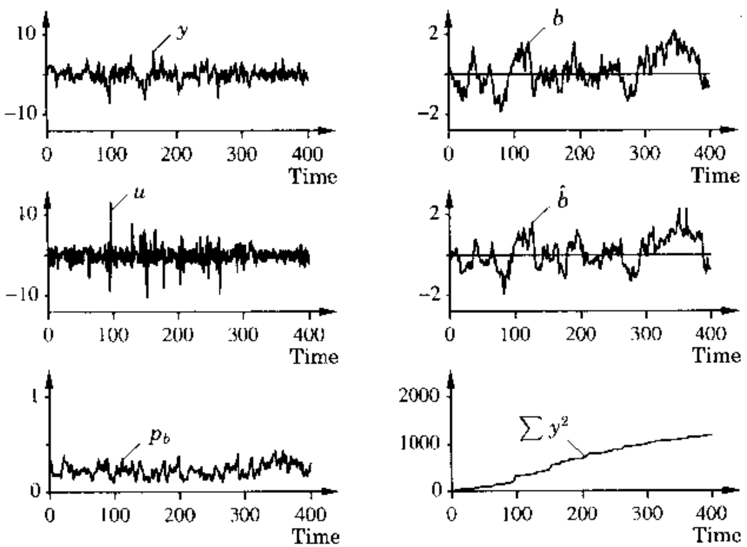

Figure 7.7 Simulation of the integrator with time-varying gain using a suboptimal dual controller. Compare Fig. 7.4.

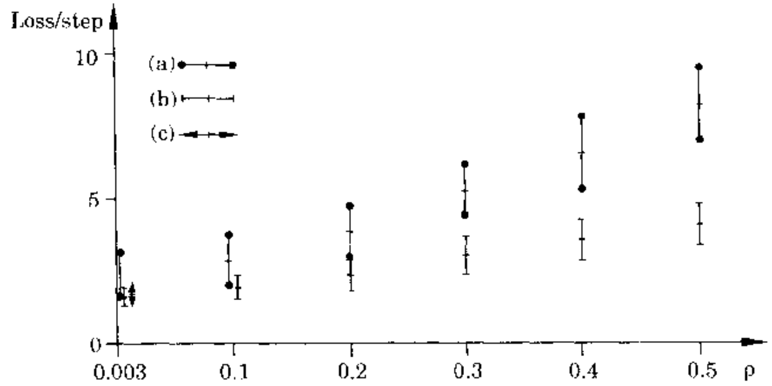

Figure 7.8 The mean values and standard deviations of the losses for Monte Carlo runs with different values of $\sqrt{R_1}$ for the system in Example 7.6. (a) Cautious controller. (b) Suboptimal dual controller from Wittenmark and Elevitch (1985). (c) Numerically computed optimal dual controller from Åström and Helmersson (1982).

For $\sqrt{R_1} = 0.003$ there is good agreement between the suboptimal controller and the optimal dual controller that was derived under the assumption that $R_1 = 0$. The optimal dual controller from Example 7.4 corresponds to $R_1 = 0$. The controller obtained has been used also for $\sqrt{R_1} = 0.003$.

In Eq. (7.20), $R_2/p_{b_0}(t+1)$ is used as a normalization factor in the term added to the loss function. The reason is an attempt to preserve a property of the dual optimal controller. In Example 7.4 the loss V_N was a function of the normalized variables η and β. The loss function of Eq. (7.19) with Eq. (7.20) will also have this property for the integrator example. Simulations indicate that the normalization in Eq. (7.20) also makes the choice of ρ less crucial. □

7.7 CONCLUSIONS

Optimal multistep controllers have been derived by using stochastic control theory. The solution is defined through the Bellman equation. This functional equation is difficult to solve even for very simple systems. The optimal solution has some interesting properties; it makes a compromise between good control and good estimation by introducing probing actions. This dual effect is of great importance, since it introduces an active learning feature into the controller. It is important to preserve this dual feature when suboptimal controllers are considered. The cautious or one-step controller does not have any active learning; the control may instead be turned off when the parameter uncertainties become too large. One important question is whether it is worth the effort to look at more elaborate control structures than the certainty equivalence controllers. The self-tuning regulators perform very well, as can be seen in Chapters 3, 4, and 6. The extra computations are not too extensive in several of the suboptimal dual controllers discussed in Section 7.5. This indicates that active learning can easily be included.

There are two situations in which dual control can pay off. One is when the time horizon is very short, in which case it is important to get good parameter estimates immediately. Areas in which this is the case are economic systems and control of missiles. The other situation in which dual features are important is when the parameters are varying rapidly and when the b_0 parameter can change sign, as in the simulations given in Section 7.6. Grinders are one type of physical process in which the gain may change sign. Grinders are common, for instance, in the mining, pulp, and paper industries.

Even if the optimal dual controller is impossible to calculate for realistic processes, it gives important hints about how to make sensible modifications of certainty equivalence and cautious controllers.

PROBLEMS

7.1 Discuss possible difficulties of extending the problem given in Section 7.3 to the case in which the system in Eq. (7.1) has an additional time delay.

7.2 Show that the cautious controller of Eq. (7.15) minimizes the loss function of Eq. (7.14) and that the minimum value of the loss function is Eq. (7.16).

7.3 Consider the process in Example 7.2, but with a constant but unknown gain b. Calculate and compare the minimum values of the loss function when

(a) the parameter b is known (i.e., the minimum-variance controller),

(b) the certainty equivalence controller is used,

(c) the cautious controller is used.

7.4 Compute the suboptimal control law that minimizes the loss function of Eq. (7.21). (*Hint*: See Goodwin and Payne (1977), p. 296.)

7.5 Compute the suboptimal control law that minimizes the loss function of Eq. (7.22). (*Hint*: See Milito *et al.* (1982).)

7.6 Assume that the process is described by one of the known models

$$y(t) = \varphi(t)\theta_i + e(t) \qquad i = 1,\dots,m$$

but it is not known which is the correct one. Let the initial information be described by the probabilities $p_i = P(\theta = \theta_i)$. Formulate the dual control problem and discuss the computational difficulties associated with the solution.

7.7 Discuss the consequences of formulating the dual control problem for the model

$$x(t+1) = \Phi(t)x(t) + \Gamma(t)u(t)$$
$$y(t) = C(t)x(t) + e(t)$$

where Φ, Γ, and C contain some unknown parameters. For simplicity, consider the case in which the system is given in controllable canonical form, that is,

$$\Phi(t) = \begin{pmatrix} -a_1(t) & -a_2(t) & \dots & -a_n(t) \\ 1 & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ 0 & \dots & 1 & 0 \end{pmatrix}$$

$$\Gamma^T(t) = \begin{pmatrix} b_0(t) & \dots & b_{n-1}(t) \end{pmatrix}$$

$$C(t) = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}$$

REFERENCES

The basic ideas of stochastic control and dynamic programming are discussed in:

Bellman, R., 1961. *Adaptive Control Processes: A Guided Tour.* Princeton, N.J.: Princeton University Press.

Åström, K. J., 1970. *Introduction to Stochastic Control Theory.* New York: Academic Press.

Bertsekas, D., 1978. *Stochastic Optimal Control.* New York: Academic Press.

More general treatments and surveys of stochastic adaptive control are found in:

Wittenmark, B., 1975. "Stochastic adaptive control methods: A survey." *Int. J. Control* **21**: 705–730.

Bar-Shalom, Y., and E. Tse, 1976. "Concepts and methods in stochastic control." In *Control and Dynamic Systems: Advances in Theory and Applications*, ed. C. T. Leondes, Vol. 12, pp. 99–172. New York: Academic Press.

Åström, K. J., 1978. "Stochastic control problems." In *Mathematical Control Theory. Lecture Notes in Mathematics*, ed. W. A. Coppel. Berlin: Springer-Verlag.

Kumar, P. R., and P. Varaiya, 1986. *Stochastic Systems: Estimation, Identification, and Adaptive Control.* Englewood Cliffs, N.J.: Prentice-Hall.

The two-armed bandit problem is discussed, for instance, in Bellman (1961) and in:

Yakowitz, S. J., 1969. *Mathematics of Adaptive Control Processes.* New York: American Elsevier.

The dual control concept with control loss and probing loss is discussed in:

Feldbaum, A. A., 1965. *Optimal Control Theory.* New York: Academic Press.

The difference between certainty equivalence and separation is treated in:

Witsenhausen, H. S., 1971. "Separation of estimation and control for discrete time systems." *Proceedings IEEE* **59**: 1557–1566.

Because of the difficulty of solving the Bellman equation, only a few dual optimal control problems have been solved. The simplified case in which the process is described as a Markov chain is discussed in:

Åström, K. J., 1965. "Optimal control of Markov processes with incomplete state information I." *J. Math. Anal. Appl.* **10**: 174–205.

Åström, K. J., 1969. "Optimal control of Markov processes with incomplete state information II." *J. Math. Anal. Appl.* **26**: 403–406.

Sternby, J., 1976. "A simple dual control problem with an analytical solution." *IEEE Trans. Automat. Contr.* **AC-21**: 840–844.

The case in which the process is a delay and there is an unknown gain is solved numerically in:

Åström, K. J., and Wittenmark, B., 1971. "Problems of identification and control." *J. Math. Anal. Appl.* **34**: 90–113.

This reference also gives examples of the turn-off phenomenon. The integrator with unknown gain is analyzed in:

Bohlin, T., 1969. "Optimal dual control of a simple process with unknown gain." Technical Paper PT 18.196, IBM Nordic Laboratory, Lidingö, Sweden.

Åström, K. J., and Helmersson, A., 1982. "Dual control of a lower order system." *Proceedings of the National CNRS Colloque "Développement et Utilisation d'Outils et Modèles Mathématiques en Automatique, Analyse de Systèmes et Traitement du Signal."* Belle-Ile, France.

Åström, K. J., and Helmersson, A., 1986. "Dual control of an integrator with unknown gain." *Comp. & Maths. with Appls.* **12A**(6): 653–662.

The computational problems of the optimal solution have led to many different suggestions for suboptimal dual controllers. Extra perturbation to avoid turn-off is discussed in:

Wieslander, J., and B. Wittenmark, 1971. "An approach to adaptive control using real-time identification." *Automatica* **7**: 211–217.

Jacobs, O. L. R., and J. W. Patchell, 1972. "Caution and probing in stochastic control." *Int. J. Control* **16**: 189–199.

Constrained minimization of the one-step loss function is treated in:

Alster, J., and P. R. Bélanger, 1974. "A technique for dual adaptive control." *Automatica* **10**: 627–634.

Hughes, D. J., and O. L. R. Jacobs, 1974. "Turn-off, escape and probing in non-linear stochastic control." *Preprint IFAC Symposium on Stochastic Control.* Budapest.

Mosca, E., S. Rocchi, and G. Zappa, 1978. "A new dual active control algorithm." *Preprints 17th IEEE Conference on Decision and Control*, pp. 509–512. San Diego, Calif.

Different extensions of the one-step loss function are discussed in:

Wittenmark, B., 1975. "An active suboptimal dual controller for systems with stochastic parameters." *Automatic Control Theory and Application* **3**: 13–19.

Goodwin, G. C., and R. L. Payne, 1977. *Dynamic System Identification: Experiment Design and Data Analysis.* New York: Academic Press.

Sternby, J., 1977. "Topics in dual control." Ph.D. thesis TFRT-1012, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Milito, R., C. S. Padilla, R. A. Padilla, and D. Cadorin, 1982. "An innovations approach to dual control." *IEEE Trans. Automat. Contr.* **AC-27**: 132–137.

Wittenmark, B., and C. Elevitch, 1985. "An adaptive control algorithm with dual features." *Preprints 7th IFAC Symposium on Identification and System Parameter Estimation*, pp. 587–592. York, U.K.

Linearization of the loss function is found in Bar-Shalom and Tse (1976) and in:

Wenk, C. J., and Bar-Shalom, Y., 1980. "A multiple model adaptive control algorithm for stochastic systems with unknown parameters." *IEEE Trans. Automat. Contr.* **AC-25**: 703–710.

Bar-Shalom, Y., P. Mookerjee, and J. A. Molusis, 1982. "A linear feedback dual controller for a class of stochastic systems." *Proceedings of the National CNRS Colloque "Développement et Utilisation d'Outils et Modèles Mathématiques en Automatique, Analyse de Systèmes et Traitement du Signal."* Belle-Ile, France.

A discussion of an industrial example in which dual control can be useful is found in:

Dumont, G., and K. J. Åström, 1988. "Wood chip refiner control." *IEEE Control Systems Magazine* **8**(2): 38–43.

AUTO-TUNING

8.1 INTRODUCTION

Adaptive schemes like MRAS and STR require *a priori* information about the process dynamics. It is particularly important to know time scales, which are critical for determining suitable sampling intervals and filtering. The importance of *a priori* information was overlooked for a long time but became apparent in connection with the development of general-purpose adaptive controllers. Several manufacturers were forced to introduce a *pre-tune mode* to help in obtaining the required prior information. The importance of prior information also appeared in connection with attempts to develop techniques for automatic tuning of simple PID regulators. Such regulators, which are standard building blocks for industrial automation, are used to control systems with a wide range of time constants.

From the user's point of view it would be ideal to have an auto-tuning function in which the regulator can be tuned simply by pushing a button. Although conventional adaptive schemes seemed to be ideal tools to provide automatic tuning, they were found to be inadequate because they required prior knowledge of time scales. Special techniques for automatic tuning of simple regulators were therefore developed. These techniques are also useful for providing pre-tuning of more complicated adaptive systems. In this chapter we describe some of these techniques. They can be characterized as crude robust methods that provide ballpark information. They are thus ideal complements to the more sophisticated adaptive methods. An overview of industrial PID controllers with auto-tuning is given in Section 12.3.

The chapter is organized as follows: The standard PID controller is discussed in Section 8.2. Different auto-tuning techniques are given in Section 8.3. Transient and frequency response methods for tuning are developed in

Sections 8.4 and 8.5, respectively, and Section 8.6 is devoted to analysis of relay oscillations. Conclusions are presented in Section 8.7.

8.2 PID CONTROL

The PID controllers are the standard tool for industrial automation. The flexibility of the controller makes it possible to use PID control in many situations. The controllers can also be used in cascade control and other controller configurations. Many simple control problems can be handled very well by PID control, provided that the performance requirements are not too high. The PID algorithm is packaged in the form of standard regulators for process control and is also the basis of many tailor-made control systems. The textbook version of the algorithm is

$$u(t) = K_c \left(e(t) + \frac{1}{T_i} \int_0^t e(s)\,ds + T_d \frac{de}{dt} \right) \tag{8.1}$$

where u is the control variable, e is the error defined as $e = u_c - y$ where u_c is the reference value, and y is the process output. The algorithm that is actually used contains several modifications. It is standard practice to let the derivative action operate only on the process output. It may be advantageous to let the proportional part act only on a fraction of the reference value. The derivative action is replaced by an approximation that reduces the gain at high frequencies. The integral action is modified so that it does not keep integrating when the control variable saturates (*anti-windup*). Precautions are also taken so that there will not be transients when the regulator is switched from manual to automatic control or when parameters are changed.

If the nonlinearity of the actuator can be described by the function f, a reasonably realistic PID regulator can be described by

$$\begin{aligned} u(t) &= f(v(t)) \\ v(t) &= P(t) + I(t) + D(t) \end{aligned} \tag{8.2}$$

where

$$\begin{aligned} P(t) &= K_c\,(\beta u_c(t) - y(t)) \\ \frac{dI}{dt} &= \frac{K_c}{T_i}\,(u_c(t) - y(t)) + \frac{1}{T_t}\,(v(t) - u(t)) \\ \frac{T_d}{N}\frac{dD}{dt} &= -D - K_c T_d \frac{dy}{dt} \end{aligned} \tag{8.3}$$

The last term in the expression for dI/dt is introduced to get anti-windup when the output saturates. This guarantees that the integral part I is bounded. The parameter T_t is a time constant for resetting the integral action when the actuator saturates. The essential parameters to be adjusted are K_c, T_i, and T_d.

The parameter N can be fixed; a typical value is $N = 10$. The tracking time constant is typically a fraction of the integration time T_i.

8.3 AUTO-TUNING TECHNIQUES

Several ways to do auto-tuning have been proposed. The most common method is to make a simple experiment on the process. The experiment can be done in open loop or closed loop. In the open-loop experiments the input of the process is excited by a pulse or a couple of steps. A simple process model, for instance of second order, is then estimated by using recursive least squares or some other recursive estimation method. If a second-order process model is estimated, then the PID controller can be used to make pole placement. The speed and the damping of the system are then the design parameters. A popular design method is to choose the controller zeros such that they cancel the two process poles. This gives good responses to setpoint changes, while the response to load disturbances is determined by the open-loop dynamics. The transient response method for automatic tuning of PID regulators is used in products from Yokogawa, Eurotherm, and Honeywell. It is used for pre-tuning in adaptive controllers from Leeds and Northrup and Turnbull Control.

The tuning experiments can also be done in closed loop. A typical example of this is the self-oscillating method of Ziegler and Nichols or its variants. The relay auto-tuner based on self-oscillation is used in products from SattControl and Fisher-Rousemount. In these regulators the tuning is initiated simply by pushing the tuning button. One advantage of making experiments in closed loop is that the output of the process may be kept within reasonable bounds, which can be difficult for processes with integrators if the experiment is done in open loop.

The auto-tuning function is often a built-in feature in standard stand-alone PID controllers. Automatic tuning can also be done by using external equipment. The tuner is then connected to the process and performs an experiment, usually in open loop. The tuner then suggests parameter settings, which are transferred to the PID controller either manually or automatically. Since the external tuner must be able to work with PID controllers from different manufacturers, it is important that the tuner have detailed information about the implementation of the PID algorithm in specific cases.

Another method for auto-tuning is to use an expert system to tune the controller. This is done during normal operation of the process. The expert system waits for setpoint changes or major load disturbances and then evaluates the performance of the closed-loop system. Properties such as damping, period of oscillation, and static gain are estimated. The controller parameters are then changed according to the built-in rules, which mimic the behavior of an experienced control engineer. Pattern recognition or expert system is used in controllers from Foxboro and Fenwal.

8.4 TRANSIENT RESPONSE METHODS

Several simple tuning methods for PID controllers are based on transient response experiments. Many industrial processes have step responses of the type shown in Fig. 8.1, in which the step response is monotonous after an initial time. A system with a step response of the type shown in Fig. 8.1 can be approximated by the transfer function

$$G(s) = \frac{k}{1 + sT} e^{-sL} \tag{8.4}$$

where k is the static gain, L is the apparent time delay, and T is the apparent time constant. The parameter a is given by

$$a = k\frac{L}{T} \tag{8.5}$$

The Ziegler-Nichols Step Response Method

A simple way to determine the parameters of a PID regulator based on step response data was developed by Ziegler and Nichols and published in 1942. The method uses only two of the parameters shown in Fig. 8.1, namely, a and L. The regulator parameters are given in Table 8.1. The Ziegler-Nichols tuning rule was developed by empirical simulations of many different systems. The rule has the drawback that it gives closed-loop systems that are often too poorly damped. Systems with better damping can be obtained by modifying the numerical values in Table 8.1. By using additional parameters it is also possible to determine whether the Ziegler-Nichols rule is applicable. If the time constant T is also determined, an empirical rule is established that the

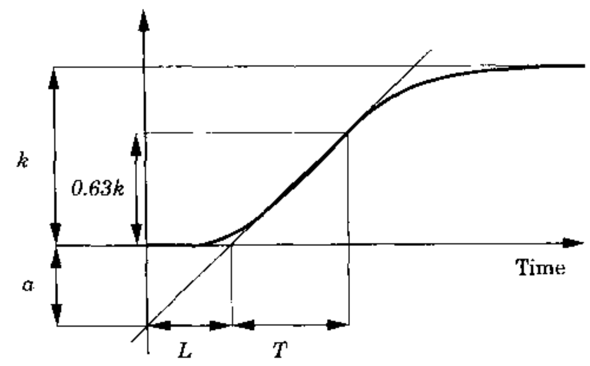

Figure 8.1 Unit step response of a typical industrial process.

Table 8.1 Regulator parameters obtained by the Ziegler-Nichols step response method.

Controller	K_c	T_i	T_d
P	$1/a$		
PI	$0.9/a$	$3L$	
PID	$1.2/a$	$2L$	$L/2$

Ziegler-Nichols rule is applicable if $0.1 < L/T < 0.6$. For large values of L/T it is advantageous to use other tuning rules or control laws that compensate for dead time. For small values of L/T, improved performance may be obtained with higher-order compensators. It is also possible to use more sophisticated tuning rules based on three parameters.

Characterization of a Step Response

The parameters k, L, and T can be determined from a graphical construction such as the one indicated in Fig. 8.1. It may be useful to take averages of several steps if the signals are noisy. There are also methods based on area measurements that can be used. One method of this type is illustrated in Fig. 8.2. The area A_0 is first determined. Then

$$T + L = \frac{A_0}{k} \tag{8.6}$$

The area A_1 under the step response up to time $T + L$ is then determined, and T is then given by

$$T = \frac{eA_1}{k} \tag{8.7}$$

where e is the base of the natural logarithm. The essential drawbacks of the method are that it may be difficult to know the size of the step in the control

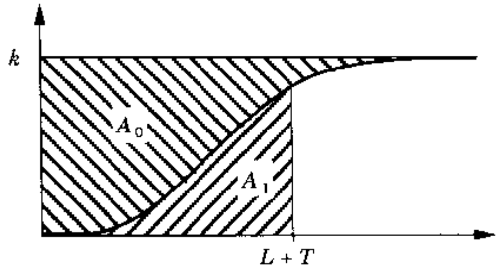

Figure 8.2 Area method for determining L and T.

signal and to determine whether a steady state has been reached. The step should be so large that the response is clearly noticeable above the noise but not so large that production is disturbed. Disturbances will also influence the result significantly.

On-line Refinement

If a reasonable regulator tuning is obtained, the damping and natural frequency of the closed-loop system can also be determined from a closed-loop transient response. The regulator tuning can then be improved.

8.5 METHODS BASED ON RELAY FEEDBACK

The main drawback of the transient response method is that it is sensitive to disturbances because it relies on open-loop experiments. The relay-based methods avoid this difficulty because the required experiments are performed in closed loop.

The Key Idea

The basic idea is the observation that many processes have limit cycle oscillations under relay feedback. A block diagram of such a system is shown in Fig. 8.3. The input and output signals obtained when the command signal u_c is zero are shown in Fig. 8.4. The figure shows that a limit cycle oscillation is established quite rapidly. We can intuitively understand what happens in the following way: The input to the process is a square wave with frequency ω_u. By a Fourier series expansion we can represent the input by a sum of sinusoids with frequencies ω_u, $3\omega_u$, and so on. The output is approximately sinusoidal, which means that the process attenuates the higher harmonics effectively. Let the amplitude of the square wave be d; then the fundamental component has the amplitude $4d/\pi$. Making the approximation that all higher harmonics can

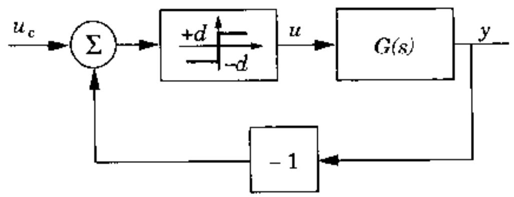

Figure 8.3 Linear system with relay control.

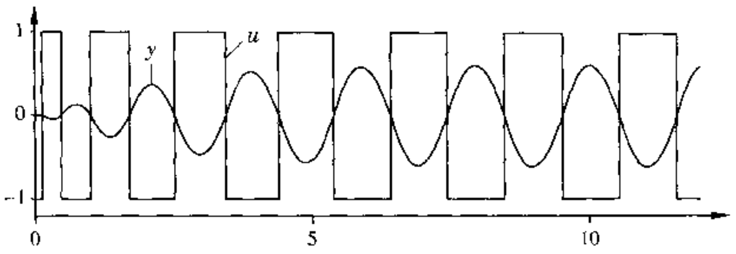

Figure 8.4 Input and output of a system with relay feedback.

be neglected, we find that the process output is a sinusoid with frequency ω_u and amplitude

$$a = \frac{4d}{\pi} |G(i\omega_u)|$$

To have an oscillation, the output must also go through zero when the relay switches. Moreover, the fundamental component of the input and the output must have opposite phase. We can thus conclude that the frequency ω_u must be such that the process has a phase lag of 180°. The conditions for oscillation are thus

$$\arg G(i\omega_u) = -\pi \qquad \text{and} \qquad |G(i\omega_u)| = \frac{a\pi}{4d} = \frac{1}{K_u} \tag{8.8}$$

where K_u can be regarded as the equivalent gain of the relay for transmission of sinusoidal signals with amplitude a. For historical reasons this parameter is called the *ultimate gain*. It is the gain that brings a system with transfer function $G(s)$ to the stability boundary under pure proportional control. The period $T_u = 2\pi/\omega_u$ is similarly called the *ultimate period*. An experiment with relay feedback is thus a convenient way to determine the ultimate period and the ultimate gain. Notice also that an input signal whose energy content is concentrated at ω_u is generated automatically in the experiment.

The Ziegler-Nichols Closed-Loop Method

Ziegler and Nichols have devised a very simple heuristical method for determining the parameters of a PID controller based on the critical gain and the critical period. The controller settings are given in Table 8.2. These parameters give a closed-loop system with quite low damping. Systems with better damping can be obtained by slight modifications of the numbers in the table. A modified method of this type is ideally matched to the determination of K_u and T_u by the relay method. This gives the relay auto-tuner shown in Fig. 8.5. When tuning is demanded, the switch is set to T, which means that relay feedback is activated and the PID regulator is disconnected. When a stable limit

Table 8.2 Regulator parameters obtained by the Ziegler-Nichols closed-loop method.

Controller	K_c	T_i	T_d
P	$0.5K_u$		
PI	$0.4K_u$	$0.8T_u$	
PID	$0.6K_u$	$0.5T_u$	$0.12T_u$

cycle is established, the PID parameters are computed, and the PID controller is then connected to the process. Naturally, the method will not work for all systems. First, there will not be unique limit cycle oscillations for an arbitrary transfer function. Second, PID control is not appropriate for all processes. Relay auto-tuning has empirically been found to work well for a large class of systems encountered in process control.

The Method of Describing Function

The approximative method used to derive the conditions for relay oscillations given by Eqs. (8.8) is called the *method of harmonic balance*. We will now describe a slight variation of the method that can be used to obtain additional insight. This is called the *describing function method*. It can be described as follows: Consider a simple feedback system composed of a linear part with the transfer function $G(s)$ and feedback with an ideal relay as shown in Fig. 8.3. The conditions for limit cycle oscillations can be determined approximately by investigating the propagation of sinusoidal signals around the loop. There will be higher harmonics because of the relay, but they will be neglected. The propagation of a sine wave through the linear system is described by the complex number $G(i\omega)$. Similarly, the propagation of a sine wave through

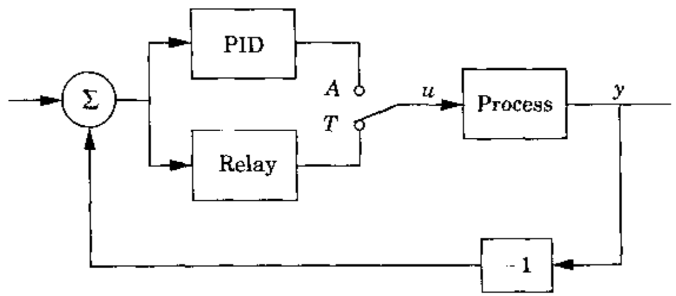

Figure 8.5 Block diagram of a relay auto-tuner.

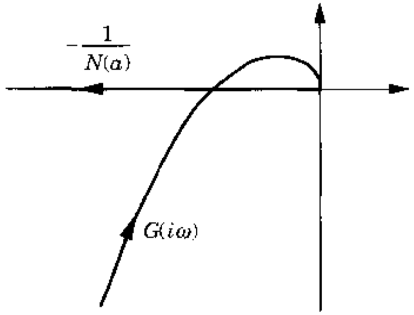

Figure 8.6 Nyquist curve $G(i\omega)$ and the describing function $N(a)$ for a relay.

the nonlinearity can also be characterized by a complex number $N(a)$, which depends on the amplitude of the signal at the input of the nonlinearity. $N(a)$ is called the describing function of the nonlinearity. The condition for oscillation is then that the signal comes back with the same amplitude and phase as it passes the closed loop. This gives the condition

$$G(i\omega)N(a) = -1$$

This condition can be represented graphically by also plotting the curve $N(a)$ in the Nyquist diagram. (See Fig. 8.6.) For the relay the nonlinearity is

$$N(a) = \frac{4d}{a\pi}$$

because a is the input signal amplitude and the fundamental component of the output has amplitude $4d/\pi$. A possible oscillation is at the intersection of the curves. The frequency is read from the Nyquist curve and the amplitude from the describing function.

EXAMPLE 8.1 Relay oscillation

Consider a system with relay feedback as in Fig. 8.3 with

$$G(s) = \frac{K\alpha}{s(s+1)(s+\alpha)}$$

$K = 5$, $\alpha = 10$, $d = 1$, and $u_c = 0$. This was the system used to generate Fig. 8.4. Simple calculations show that

$$\begin{aligned}\arg G(i\omega_u) &= -\frac{\pi}{2} - \tan^{-1}\omega_u - \tan^{-1}\frac{\omega_u}{\alpha} \\ &= -\frac{\pi}{2} - \tan^{-1}\frac{\omega_u(\alpha+1)}{\alpha-\omega_u^2} = -\pi\end{aligned}$$

This implies that the Nyquist curve intersects the negative real axis for $\omega_u = \sqrt{\alpha}$. The approximative analysis thus gives the following estimate of the period:

$$T_u = \frac{2\pi}{\sqrt{\alpha}} = \frac{6.28}{\sqrt{\alpha}} = 1.99$$

Using Eqs. (8.8) gives $a = 4d|G(i\omega_u)|/\pi = 0.58$. From the simulations it can be determined that the true values are $T_u = 2.07$ and $a = 0.62$, which show that the describing function method gives fair but not very accurate estimates in this example. □

Several refinements of the method are useful. The amplitude of the limit cycle oscillation can be specified by introducing a feedback that adjusts the relay amplitude. A hysteresis in the relay is useful to make the system less sensitive to noise. The parameters T_u and K_u can be used to determine the parameters of a PID regulator. The method can be made insensitive to disturbances by comparing and averaging over several periods of the oscillation.

EXAMPLE 8.2 Auto-tuning of cascaded tanks

The properties of a relay auto-tuner are illustrated by an example. The process to be controlled consists of three cascaded tanks. The level of the lower tank is measured, and the control variable is the voltage to the amplifier driving the pump for the inlet. The signals are noisy. The relay in the auto-tuner has a hysteresis, which is determined automatically on the basis of measurements of the process disturbances. The relay amplitude is also adjusted automatically to keep a specified amplitude of the limit cycle. The limit cycle is judged to be stationary by measuring the periods and amplitudes of two positive half-periods. Figure 8.7 shows the process inputs and outputs in one experiment, illustrating the effect of amplitude adjustment. When the tuning is finished,

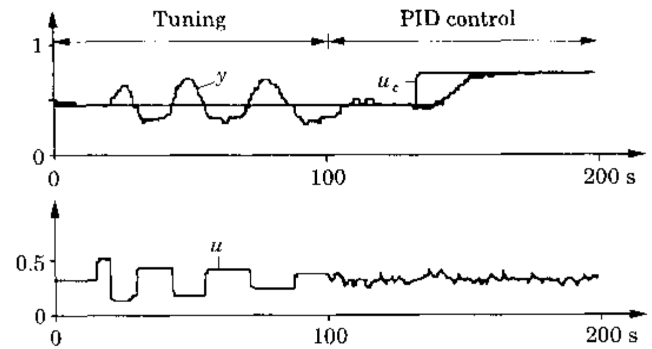

Figure 8.7 Results obtained by applying an auto-tuner to level control of three cascaded tanks.

the regulator is switched to PID control automatically. A change of the setpoint shows that the tuning has been successful. □

Improved Estimates and Pre-tuning

So far, only two parameters, K_u and T_u, have been extracted from the relay experiment. Much more information can be obtained. By changing the setpoint during the relay experiment it is possible to determine the static process gain k. The product kK_u can then be used to assess the appropriateness of PID control with Ziegler-Nichols tuning. A common rule is that the Ziegler-Nichols method can be used if $2 < kK_u < 20$. Values that are lower than 2 indicate that a control law that admits dead-time compensation should be used. Large values of kK_u indicate that improved performance can be obtained with a more complex control algorithm. The relay experiment can also be used to estimate a discrete-time transfer function by using standard system identification methods.

The relay method is ideally suited as a pre-tuner for a more sophisticated adaptive controller. A model such as Eq. (8.4) is very useful to select the sampling period and achievable closed-loop response for an MRAS or an STR. It provides a PID controller that can serve as a backup controller. If the static gain is also determined, the quantity kK_u can be used to assess the process dynamics. The ultimate period can be used to obtain an estimate of an appropriate sampling period. Parameter estimates that can serve as initial values in the recursive parameter estimator can be obtained by applying a parameter estimation method to the data from the relay experiments. If an adaptive controller based on a pole placement design is used, the ultimate period can also be used to find appropriate values of desired closed-loop bandwidths.

8.6 RELAY OSCILLATIONS

Since limit cycling under relay feedback is a key idea of relay auto-tuning, it is important to understand why a linear system oscillates under relay feedback and when the oscillation is stable. It is also important to have methods for determining the period and the amplitude of the oscillations. Consider the system shown in Fig. 8.3. Introduce the following state space realization of the transfer function $G(s)$:

$$\begin{aligned} \frac{dx}{dt} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{8.9}$$

The relay can be described by

$$u = \begin{cases} d & \text{if } e > 0 \\ -d & \text{if } e < 0 \end{cases} \tag{8.10}$$

where $e = u_c - y$. We have the following result.

THEOREM 8.1 Limit cycle period

Assume that the system defined in Fig. 8.3 and by Eqs. (8.9) and (8.10) has a symmetric limit cycle with period T. The period T is then the smallest value of $T > 0$ that satisfies the equation

$$C(I + \Phi)^{-1}\Gamma = 0 \tag{8.11}$$

where

$$\Phi = e^{AT/2}$$

and

$$\Gamma = \int_0^{T/2} e^{As}\, ds\, B$$

Proof: Let t_k denote the times when the relay switches. Since the limit cycle is symmetric, it follows that

$$t_{k+1} - t_k = T/2$$

Assume that the control signal u is d over the interval (t_k, t_{k+1}). Integration of Eqs. (8.9) over the interval gives

$$x(t_{k+1}) = \Phi x(t_k) + \Gamma d$$

Since the limit cycle is symmetric, it also follows that

$$x(t_{k+1}) = -x(t_k)$$

Hence

$$x(t_k) = -(I + \Phi)^{-1}\Gamma d$$

Since the output $y(t)$ must be zero at t_k, it follows that

$$y(t_k) = Cx(t_k) = -C(I + \Phi)^{-1}\Gamma d = 0$$

which gives Eq. (8.11). □

Remark 1. The condition of Eq. (8.11) can also be written as

$$H_{T/2}(-1) = 0 \tag{8.12}$$

where $H_{T/2}(z)$ is the pulse transfer function obtained when sampling the system of Eqs. (8.9) with period $T/2$.

Remark 2. The result that the period is given by Eq. (8.12) also holds for linear systems with a time delay, provided that $T/2$ is larger than or equal to the delay.

Remark 3. Similar conditions can also be derived for relays with hysteresis. □

Comparison with the Describing Function

Having obtained the exact formula of Eq. (8.11) for T, it is possible to investigate the precision of the describing function approximation. Consider the symmetric case and introduce $h = T/2$. The pulse transfer function obtained in sampling the system of Eqs. (8.9) with period h is given by

$$H_h(e^{sh}) = \frac{1}{h}\sum_{n=-\infty}^{\infty} \frac{1}{s + in\omega_s}\left(1 - e^{-h(s+in\omega_s)}\right) G(s + in\omega_s)$$

where $\omega_s = 2\pi/h$. Put $sh = i\pi$:

$$\begin{aligned} H_h(-1) &= \sum_{-\infty}^{\infty} \frac{2}{i(\pi + 2n\pi)} G\left(i\frac{\pi + 2n\pi}{h}\right) \\ &= \sum_{0}^{\infty} \frac{4}{\pi(1+2n)} \operatorname{Im}\left(G\left(i\frac{\pi + 2n\pi}{h}\right)\right) = 0 \end{aligned}$$

The first term of the series gives

$$H_h(-1) \approx \frac{4}{\pi} \operatorname{Im}\left(G\left(i\frac{\pi}{h}\right)\right) = \frac{4}{\pi} \operatorname{Im}\left(G\left(i\frac{2\pi}{T}\right)\right) = 0$$

which is the same result for calculation of T obtained from the describing function analysis. This implies that the describing function approximation is accurate only if $G(s)$ has low-pass character. An example illustrates determination of the period of oscillation.

EXAMPLE 8.3 Limit cycle period

Consider the same process as in Example 8.1. To apply Theorem 8.1, the system $G(s)$ is sampled with period h. The pulse transfer function is

$$H_h(z) = \frac{Kh}{(z-1)} - \frac{K\alpha(1-e^{-h})}{(\alpha-1)(z-e^{-h})} + \frac{K(1-e^{-\alpha h})}{\alpha(\alpha-1)(z-e^{-\alpha h})}$$

Hence

$$\begin{aligned} H_h(-1) &= -\frac{Kh}{2} + \frac{K\alpha(1-e^{-h})}{(\alpha-1)(1+e^{-h})} - \frac{K(1-e^{-\alpha h})}{\alpha(\alpha-1)(1+e^{-\alpha h})} \\ &= -\frac{Kh}{2} + \frac{K\alpha}{\alpha-1}\left(\frac{1-e^{-h}}{1+e^{-h}} - \frac{1}{\alpha^2}\frac{1-e^{-\alpha h}}{1+e^{-\alpha h}}\right) = 0 \end{aligned}$$

Numerical search for the value of h that satisfies this equation gives $h \doteq 1.035$. This gives $T_u = 2.07$, which agrees with the simulation in Fig. 8.4. □

Stable periodic solutions will not be obtained for all systems. A double integrator under pure relay control, for example, will give periodic solutions with an arbitrary period.

8.7 CONCLUSIONS

In this chapter we have described simple robust methods that can be used to get crude estimates of process dynamics. The methods can be used for automatic tuning of simple regulators of the PID type or as pre-tuners for more sophisticated adaptive control algorithms. Two types of methods have been discussed: a transient method based on open-loop step tests and a closed-loop method based on relay feedback.

PROBLEMS

8.1 Consider a process characterized by the transfer function

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

Show that parameters T and L are exactly given by Eqs. (8.6) and (8.7).

8.2 Consider a process with the transfer function

$$G(s) = \prod_{k=1}^{n} \frac{1}{(1+sT_k)} e^{-sL}$$

Show that Eq. (8.6) gives

$$T + L = \sum_{k=1}^{n} T_k + L$$

8.3 Consider a process described by the transfer function

$$G(s) = \frac{k}{s}e^{-sL}$$

Determine a proportional regulator that gives an amplitude margin $A_m = 2$. Show that it is identical to the setting obtained by applying the Ziegler-Nichols rule in Table 8.2.

8.4 Determine the period of the limit cycle obtained when processes with transfer functions

(a) $G(s) = \frac{k}{s}e^{-sL}$ (b) $G(s) = \frac{1}{(s+1)^3}$ (c) $G(s) = \frac{1}{s^2}$

are provided with relay feedback. Use both the approximate and exact methods.

8.5 Consider a process with the transfer function given in Problem 8.3. Determine a proportional regulator obtained with the Ziegler-Nichols method given in Table 8.2.

REFERENCES

The PID regulator is very common. It is the standard tool for solving most process control problems. Various aspects of PID control are discussed in:

Smith, C. L., 1972. *Digital Computer Process Control.* Scranton, Pa.: Intext Educational Publishers.

Shinskey, F. G. , 1979. *Process-Control Systems Application Design Adjustment.* New York: McGraw-Hill.

Desphande, P. B., and R. H. Ash, 1981. *Elements of Computer Process Control with Advanced Control Applications.* Research Triangle Park, N.C.: Instrument Society of America.

The Ziegler-Nichols tuning rules were presented in:

Ziegler, J. G., and N. B. Nichols, 1942. "Optimum settings for automatic controllers." *Trans. ASME* **64**: 759–768.

Tuning rules based on three parameters k, T, and L are presented in:

Cohen, G. H., and G. A. Coon, 1953. "Theoretical consideration of retarded control." *Trans. ASME* **15**: 827–834.

A discussion of many different tuning rules for PID controllers is found in:

McMillan, G. K., 1983. *Tuning and Control Loop Performance.* Research Triangle Park, N.C.: Instrument Society of America.

Interesting views on PID control versus more advanced controls for process control applications are found in:

McMillan, G. K., 1986. "Advanced control algorithms: Beware of false prophecies." *Intech* January: 55–57.

The relay auto-tuner was presented in:

Åström, K. J., and T. Hägglund, 1984. "Automatic tuning of simple regulators with specifications on phase and amplitude margins." *Automatica* **20**: 645–651.

It is also patented:

Hägglund, T., and K. J. Åström, 1985. "Method and an apparatus in tuning a PID regulator." U.S. Patent Number 4549123.

A detailed treatment of PID control is given in:

Åström, K. J., and T. Hägglund, 1995. *PID Control.* Research Triangle Park, N.C.: Instrument Society of America.

Discussions of automatic tuning of simple regulators are found in:

Åström, K. J., and T. Hägglund, 1988. *Automatic Tuning of PID Regulators.* Research Triangle Park, N.C.: Instrument Society of America.

Åström, K. J., T. Hägglund, C. C. Hang, and W. K. Ho, 1993. "Automatic tuning and adaptation for PID controllers: A survey." *Control Eng. Practice* **1**: 699–714.

Tsypkin pioneered the research in relay feedback. His results on relay oscillations are treated in detail in:

Tsypkin, Y. Z., 1984. *Relay Control Systems.* Cambridge, U.K.: Cambridge University Press.

CHAPTER 9

GAIN SCHEDULING

9.1 INTRODUCTION

In many situations it is known how the dynamics of a process change with the operating conditions of the process. One source for the change in dynamics may be nonlinearities that are known. It is then possible to change the parameters of the controller by monitoring the operating conditions of the process. This idea is called *gain scheduling,* since the scheme was originally used to accommodate changes in process gain only. Gain scheduling is a nonlinear feedback of special type; it has a linear controller whose parameters are changed as a function of operating conditions in a preprogrammed way. The idea of relating the controller parameters to auxiliary variables is old, but the hardware needed to implement it easily was not available until recently. To implement gain scheduling with analog techniques, it is necessary to have function generators and multipliers. Such components have been quite expensive to design and operate. Gain scheduling has thus been used only in special cases, such as in autopilots for high-performance aircraft. Gain scheduling is easy to implement in computer-controlled systems, provided that there is support in the available software.

Gain scheduling based on measurements of operating conditions of the process is often a good way to compensate for variations in process parameters or known nonlinearities of the process. It is controversial whether a system with gain scheduling should be considered an adaptive system or not, because the parameters are changed in an open-loop or preprogrammed fashion. If we use the informal definition of adaptive controllers given in Section 1.1, gain scheduling can be regarded as an adaptive controller. Gain scheduling is a very useful technique for reducing the effects of parameter variations. In fact it is the foremost method for handling parameter variations in flight control systems. There are also many commercial process control systems in which gain scheduling can be used to compensate for static and dynamic nonlinearities. Split-range controllers that use different sets of parameters for different ranges of the process output can be regarded as a special type of gain-scheduling controllers.

Section 9.2 gives the principle of gain scheduling. Different ways to design systems with gain scheduling are treated in Section 9.3, and Section 9.4 gives a method based on nonlinear transformations. Section 9.5 describes some applications of gain scheduling. Conclusions are given in Section 9.6.

9.2 THE PRINCIPLE

It is sometimes possible to find auxiliary variables that correlate well with the changes in process dynamics. It is then possible to reduce the effects of parameter variations simply by changing the parameters of the controller as functions of the auxiliary variables (see Fig. 9.1). Gain scheduling can thus be viewed as a feedback control system in which the feedback gains are adjusted by using feedforward compensation. The concept of gain scheduling originated in connection with the development of flight control systems. In this application the Mach number and the dynamic pressure are measured by air data sensors and used as scheduling variables.

A main problem in the design of systems with gain scheduling is to find suitable scheduling variables. This is normally done on the basis of knowledge of the physics of a system. In process control the production rate can often be chosen as a scheduling variable, since time constants and time delays are often inversely proportional to production rate. (Compare Example 1.5.)

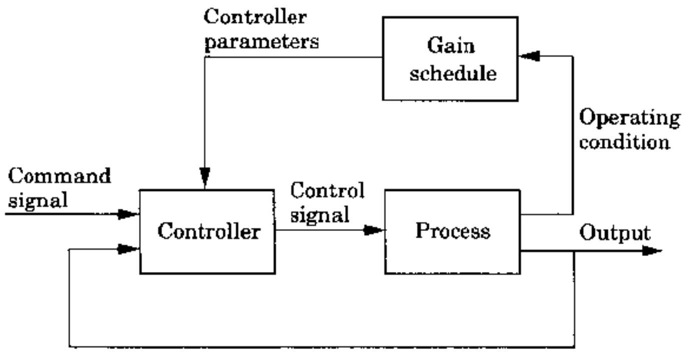

Figure 9.1 Block diagram of a system in which influences of parameter variations are reduced by gain scheduling.

When scheduling variables have been determined, the controller parameters are calculated at a number of operating conditions by using some suitable design method. The controller is thus tuned or calibrated for each operating condition. The stability and performance of the system are typically evaluated by simulation; particular attention is given to the transition between different operating conditions. The number of entries in the scheduling tables is increased if necessary. Notice, however, that there is no feedback from the performance of the closed-loop system to the controller parameters.

It is sometimes possible to obtain gain schedules by introducing nonlinear transformations in such a way that the transformed system does not depend on the operating conditions. The auxiliary measurements are used together with the process measurements to calculate the transformed variables. The transformed control variable is then calculated and retransformed before it is applied to the process. The controller thus obtained can be regarded as being composed of two nonlinear transformations with a linear controller in between. Sometimes the transformation is based on variables that are obtained indirectly through state estimation. Examples are given in Sections 9.4 and 9.5.

One drawback of gain scheduling is that it is an open-loop compensation. There is no feedback to compensate for an incorrect schedule. Another drawback of gain scheduling is that the design may be time-consuming. The controller parameters must be determined for many operating conditions, and the performance must be checked by extensive simulations. This difficulty is partly avoided if scheduling is based on nonlinear transformations.

Gain scheduling has the advantage that the controller parameters can be changed very quickly in response to process changes. Since no estimation of parameters occurs, the limiting factors depend on how quickly the auxiliary measurements respond to process changes.

9.3 DESIGN OF GAIN-SCHEDULING CONTROLLERS

It is difficult to give general rules for designing gain-scheduling controllers. The key question is to determine the variables that can be used as scheduling variables. It is clear that these auxiliary signals must reflect the operating conditions of the plant. Ideally, there should be simple expressions for how the controller parameters relate to the scheduling variables. It is thus necessary to have good insight into the dynamics of the process if gain scheduling is to be used. The following general ideas can be useful:

- Linearization of nonlinear actuators,
- Gain scheduling based on measurements of auxiliary variables,
- Time scaling based on production rate, and
- Nonlinear transformations.

The ideas are illustrated by some examples.

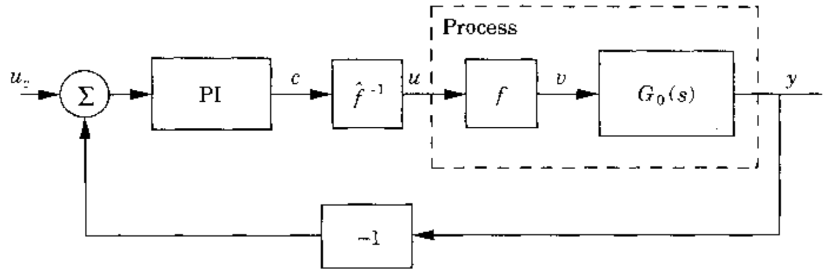

Figure 9.2 Compensation of a nonlinear actuator using an approximate inverse.

EXAMPLE 9.1 Nonlinear actuator

Consider the system with a nonlinear valve in Example 1.4. The nonlinearity is assumed to be

$$v = f(u) = u^4 \qquad u \geq 0$$

Let $\hat{f}^{-1}$ be an approximation of the inverse of the valve characteristic. To compensate for the nonlinearity, the output of the controller is fed through this function before it is applied to the valve (see Fig. 9.2). This gives the relation

$$v = f(u) = f\left(\hat{f}^{-1}(c)\right)$$

where c is the output of the PI controller. The function $f(\hat{f}^{-1}(c))$ should have less variation in gain than f. If $\hat{f}^{-1}$ is the exact inverse, then $v = c$.

Assume that $f(u) = u^4$ is approximated by two lines (see Fig. 9.3): one connecting the points (0, 0) and (1.3, 3) and the other connecting (1.3, 3) and

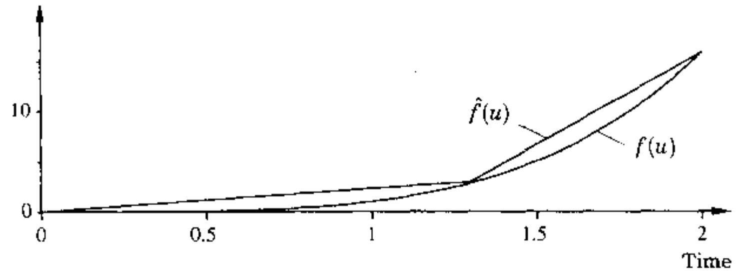

Figure 9.3 The nonlinear valve characteristic $v = f(u) = u^4$ and a two-line approximation $\hat{f}(u)$.

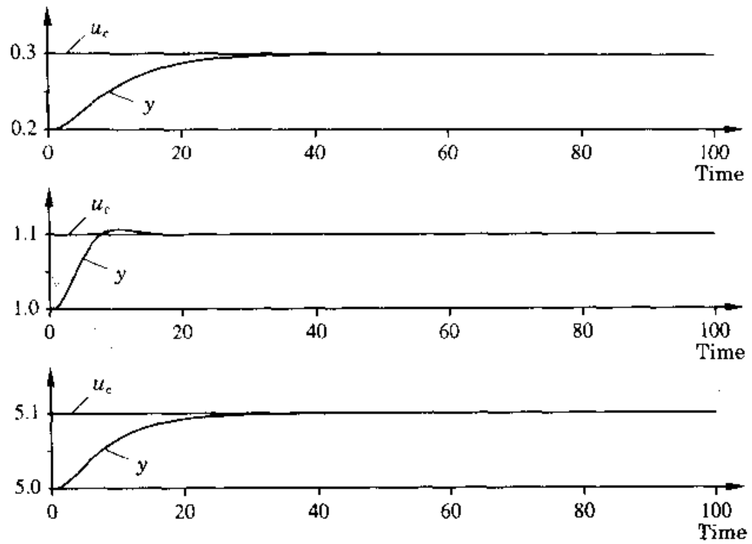

Figure 9.4 Simulation of the system in Example 9.1 with nonlinear valve and compensation using an approximation of the valve characteristic. Compare Fig. 1.9.

(2, 16). Then

$$\hat{f}^{-1}(c) = \begin{cases} 0.433c & 0 \leq c \leq 3 \\ 0.0538c + 1.139 & 3 \leq c \leq 16 \end{cases}$$

Figure 9.4 shows step changes in the reference signal at three different operating conditions when the approximation of the inverse of the valve characteristic is used between the controller and the valve. (Compare with the uncompensated system in Fig. 1.9.) There is considerable improvement in the performance of the closed-loop system. By improving the inverse it is possible to make the process even more insensitive to the nonlinearity of the valve. □

Example 9.1 shows a simple and very useful idea to compensate for known static nonlinearities. In practice it is often sufficient to approximate the nonlinearity by a few line segments. There are several commercial single-loop controllers that can make this kind of compensation. DDC packages usually include functions that can be used to implement nonlinearities.

The resulting controller in Example 9.1 is nonlinear and should (in its basic form) not be regarded as gain scheduling. In Example 9.1 there is no measurement of any operating condition apart from the controller output. In other situations the nonlinearity is determined from measurement of several variables. However, a gain-scheduling controller should contain measurement of a variable that is related to the operating point of the process. Gain scheduling based on an auxiliary signal is illustrated in the following example.

EXAMPLE 9.2 Tank system

Consider a tank in which the cross section A varies with height h. The model is

$$V = \int_0^h A(\tau)\,d\tau$$

$$\frac{dV}{dt} = A(h)\frac{dh}{dt} = q_i - a\sqrt{2gh}$$

where V is the volume, q_i is the input flow, and a is the cross section of the outlet pipe. Let q_i be the input, and let h be the output of the system. The linearized model at an operating point, q_{in}^0 and h^0, is given by the transfer function

$$G(s) = \frac{\beta}{s+\alpha}$$

where

$$\beta = \frac{1}{A(h^0)} \qquad \alpha = \frac{q_{in}^0}{2A(h^0)h^0} = \frac{a\sqrt{2gh^0}}{2A(h^0)h^0}$$

A good PI control of the tank is given by

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int e(\tau)\,d\tau\right)$$

where

$$K = \frac{2\zeta\omega - \alpha}{\beta}$$

and

$$T_i = \frac{2\zeta\omega - \alpha}{\omega^2}$$

This gives a closed-loop system with natural frequency ω and relative damping ζ. Introducing the expressions for α and β gives the following gain schedule:

$$K = 2\zeta\omega A(h^0) - \frac{q_{in}^0}{2h^0}$$

$$T_i = \frac{2\zeta}{\omega} - \frac{q_{in}^0}{2A(h^0)h^0\omega^2}$$

The numerical values are often such that $\alpha \ll 2\zeta\omega$. The schedule can then be simplified to

$$K = 2\zeta\omega A(h^0)$$

$$T_i = \frac{2\zeta}{\omega}$$

In this case it is thus sufficient to make the gain proportional to the cross section of the tank. □

Example 9.2 illustrates that it can sometimes be sufficient to measure one or two variables in the process and use them as inputs to the gain schedule. Often, it is not as easy as in Example 9.2 to determine the controller parameters as a function of the measured variables. The design of the controller must then be redone for different working points of the process. Some care must also be exercised if the measured signals are noisy. They may have to be filtered properly before they are used as scheduling variables.

The next example illustrates that gains, delays, and time constants are often inversely proportional to the production rate of the process. This fact can be used to make time scaling.

EXAMPLE 9.3 Concentration control

Consider the concentration control problem in Example 1.5. The process is described by Eq. (1.3). Assume that we are interested in manipulating the concentration in the tank, c, by changing the inlet concentration, c_{in}. For a fixed flow the dynamics can be described by the transfer function

$$G(s) = \frac{1}{1+sT}e^{-s\tau}$$

where

$$T = V_m/q \qquad \tau = V_d/q$$

If $\tau < T$, then it is straightforward to determine a PI controller that performs well when q is constant. However, it is difficult to find universal values of the controller parameters that will work well for wide ranges of q. This is illustrated in Fig. 1.11, which shows the step responses of a fixed-gain controller for varying flows. Since the process has a time delay, it is natural to look for sampled data controllers. Sampling of the model with sampling period $h = V_d/(dq)$, where d is an integer, gives

$$c(kh+h) = ac(kh) + (1-a)u(kh-dh)$$

where

$$a = e^{-qh/V_m} = e^{-V_d/(V_m d)}$$

Notice that the sampled data model has only one parameter, a, that does not depend on q. A constant-gain controller can easily be designed for the sampled data system.

The gain scheduling is realized simply by having a controller with constant parameters, in which the sampling rate is inversely proportional to the flow rate. This will give the same response, independent of the flow, in looking at the sampling instants, but the transients will be scaled in time. Figure 9.5 shows the output concentration and the control signals for three different flows. To

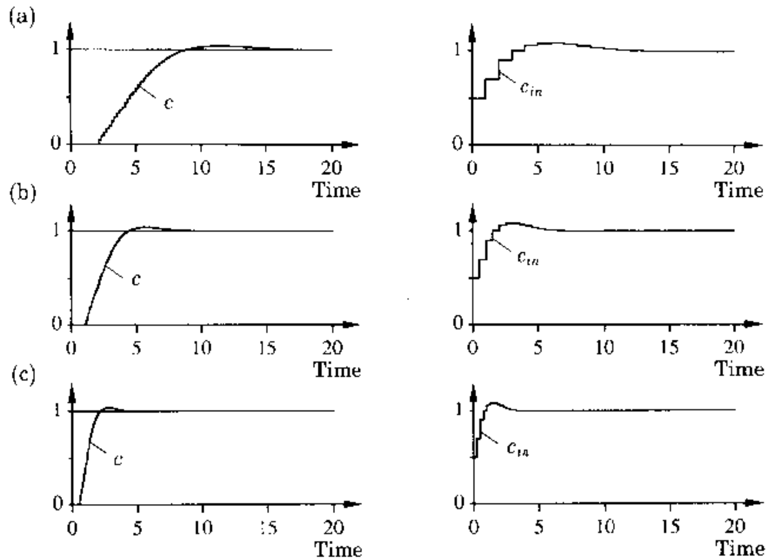

Figure 9.5 Output concentration and control signal when the process in Example 9.3 is controlled by a fixed digital controller but the sampling interval is $h = 1/(2q)$. (a) $q = 0.5$; (b) $q = 1$; (c) $q = 2$.

implement this gain-scheduling controller, it is necessary to measure not only the concentration but also the flow. Errors in the flow measurement will result in jitter in the sampling period. To avoid this, it is necessary to filter the flow measurement.

The Ziegler-Nichols transient response method discussed in Section 8.4 is based on a model with a time delay and a first-order system. Table 8.1 gives

$$K_c = \frac{0.9\tau}{T} = \frac{0.9V_d}{V_m}$$
$$T_i = 3\tau = \frac{3V_d}{q}$$

That is, the integration time is inversely proportional to the flow q. This is the same effect as is obtained with the discrete-time controller when the sampling period is inversely proportional to q. □

In Examples 9.1 and 9.2 it was possible to determine the schedules exactly. The behavior of the closed-loop system does not depend on the operating conditions. In other cases it is possible to obtain only approximate relations for different operating conditions. The design then has to be repeated for several operating conditions to create a table. It is also necessary to interpolate between the values of the table to obtain a smooth behavior of the closed-loop

system. This can lead to extensive calculations and simulations before the full gain schedule is obtained.

The gain schedule is usually obtained through simulations of a process model, but it is also possible to build up the gain table on-line. This might be done by using an auto-tuner or an adaptive controller. The adaptive system is used to get the controller parameters for different operating points. The parameters are then stored for later use when the system returns to the same or a neighboring operating point.

9.4 NONLINEAR TRANSFORMATIONS

It is of great interest to find transformations such that the transformed system is linear and independent of the operating conditions. The process in Example 9.3 is one example in which this can be done by time scaling. The obtained sampled model is independent of the flow because the time is scaled as

$$t_s = \frac{V_d}{q} t$$

This means that the key variable is distance traveled by a particle instead of time. All processes associated with material flows—rolling mills, band transporters, flows in pipes, and so on—have this property.

A system of the form

$$\frac{dx(t)}{dt} = f(x(t)) + g(x(t))\,u(t)$$

can also be transformed into a linear system, provided that all states of the system can be measured and a generalized observability condition holds. (Compare Section 5.10.) The system is first transformed into a fixed linear system. The transformation is usually nonlinear and depends on the states of the process. A controller is then designed for the transformed model, and the control signals of the model are retransformed into the original control signals. The result is a special type of nonlinear controller, which can be interpreted as a gain-scheduling controller. Knowledge about the nonlinearities in the model is built into the controller. The method with nonlinear transformations is illustrated by an example.

EXAMPLE 9.4 Nonlinear transformation of a pendulum

Consider the system

$$\begin{aligned} \frac{dx_1}{dt} &= x_2 \\ \frac{dx_2}{dt} &= -\sin x_1 + u\cos x_1 \\ y &= x_1 \end{aligned} \tag{9.1}$$

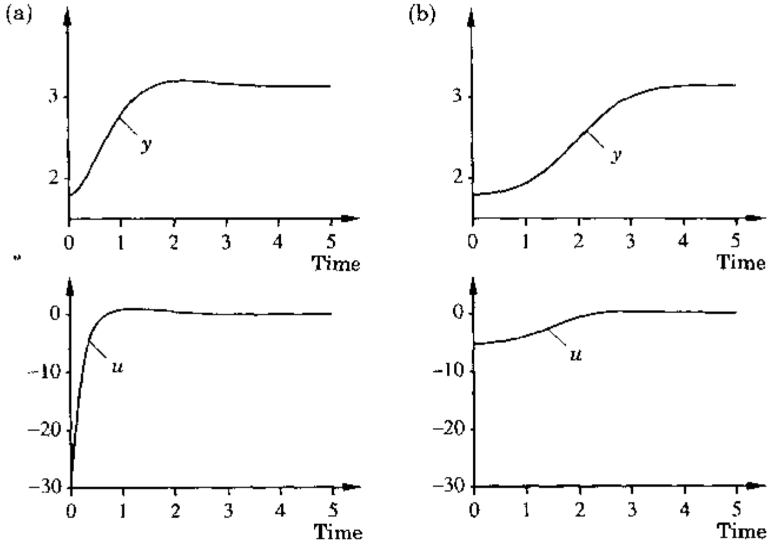

Figure 9.6 The pendulum described by Eqs. (9.1), controlled by (a) the nonlinear controller of Eq. (9.3) and (b) the fixed-gain controller of Eq. (9.4). The desired characteristic equation (Eq. 9.2) is defined by $p_1 = 2.8$ and $p_2 = 4$.

which describes a pendulum, where the acceleration of the pivot point is the input and the output y is the angle from a downward position. Introduce the transformed control signal

$$v(t) = -\sin x_1(t) + u(t)\cos x_1(t)$$

This gives the linear equations

$$\frac{dx}{dt} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} v$$

Assume that x_1 and x_2 are measured, and introduce the control law

$$v(t) = -l_1' x_1(t) - l_2' x_2(t) + m' u_c(t)$$

The transfer function from u_c to y is

$$\frac{m'}{s^2 + l_2' s + l_1'}$$

Let the desired characteristic equation be

$$s^2 + p_1 s + p_2 \tag{9.2}$$

which can be obtained with

$$l_1' = p_2 \qquad l_2' = p_1 \qquad m' = p_2$$

Transformation back to the original control signal gives

$$u(t) = \frac{v(t) + \sin x_1(t)}{\cos x_1(t)} = \frac{1}{\cos x_1(t)} \left(-p_2 x_1(t) - p_1 x_2(t) + p_2 u_c(t) + \sin x_1(t)\right) \tag{9.3}$$

The controller is thus highly nonlinear. Figure 9.6 shows the output and the control signal when the controller of Eq. (9.3) is used and when a fixed-gain controller

$$u(t) = -l_1 x_1(t) - l_2 x_2(t) + m u_c(t) \tag{9.4}$$

is used. The parameters l_1, l_2, and m are chosen to give the characteristic equation (Eqs. 9.2) when the system is linearized around $x_1 = \pi$, that is, the upright position.

Notice that Eq. (9.3) can be used for all angles except for $x_1 = \pm\pi/2$, that is, when the pendulum is horizontal. The magnitude of the control signal increases without bounds when x_1 approaches $\pm\pi/2$. The linearized model is not controllable at this operating point. □

The following example illustrates how to use the method of nonlinear transformations for a second-order system.

EXAMPLE 9.5 Nonlinear transformation of a second-order system

Consider the system

$$\frac{dx_1}{dt} = f_1(x_1, x_2)$$

$$\frac{dx_2}{dt} = f_2(x_1, x_2, u)$$

$$y = x_1$$

Assume that the state variables can be measured and that we want to find a feedback such that the response of the variable x_1 to the command signal is given by the transfer function

$$G(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2} \tag{9.5}$$

Introduce new coordinates z_1 and z_2, defined by

$$z_1 = x_1$$

$$z_2 = \frac{dx_1}{dt} = f_1(x_1, x_2)$$

and the new control signal v, defined by

$$v = F(x_1, x_2, u) = \frac{\partial f_1}{\partial x_1} f_1 + \frac{\partial f_1}{\partial x_2} f_2 \tag{9.6}$$

These transformations result in the linear system

$$\frac{dz_1}{dt} = z_2$$
$$\frac{dz_2}{dt} = v \tag{9.7}$$

It is easily seen that the linear feedback

$$v = \omega^2(u_c - z_1) - 2\zeta\omega z_2 \tag{9.8}$$

gives the desired closed-loop transfer function of Eq. (9.5) from u_c to $z_1 = x_1$ for the linear system of Eqs. (9.7). It remains to transform back to the original variables. It follows from Eqs. (9.6) and (9.8) that

$$F(x_1, x_2, u) = \frac{\partial f_1}{\partial x_1} f_1 + \frac{\partial f_1}{\partial x_2} f_2 = \omega^2(u_c - x_1) - 2\zeta\omega f_1(x_1, x_2)$$

Solving this equation for u gives the desired feedback. It follows from the implicit function theorem that a condition for local solvability is that the partial derivative $\partial F/\partial u$ is different from zero. □

The generalization of Example 9.5 requires a solution to the general problem of transforming a nonlinear system into a linear system by nonlinear feedback. Conditions and examples are given in the references at the end of this chapter. Figure 9.7 shows the general case when the full state is measured. There is a nonlinear transformation

$$u = g_1(x, v)$$
$$z = g_2(x)$$

that makes the relation between v and z linear. A state feedback controller from z is then computed that gives v. The control signal v is then transformed into

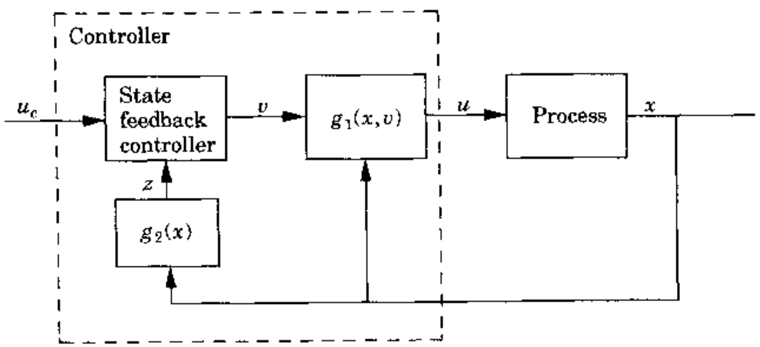

Figure 9.7 Block diagram of a controller based on nonlinear transformation.

the original control signal u. Feedback linearization requires good knowledge about the nonlinearities of the process. Uncertainties will give a transformed system that is not linear, although it may be easier to control than the original system.

A simple version of the problem also occurs in control of industrial robots. In this case the basic equation can be written as

$$J\frac{d^2\varphi}{dt^2} = T_e$$

where J is the moment of inertia, φ is an angle at a joint, and T_e is a torque, which depends on the motor current, the torque angles, and their first two derivatives. The equations are thus in the desired form, and the nonlinear feedback is obtained by determining the currents that give the desired torque. The problem is therefore called the *torque transformation*.

9.5 APPLICATIONS OF GAIN SCHEDULING

Gain scheduling is a very useful method. It requires good knowledge about the process and that some auxiliary variables can be measured. A great advantage with the method is that the controller adapts quickly to changing conditions.

This section contains examples of some cases in which it is advantageous to use gain scheduling in some of the forms that have been presented above. The examples include ship steering, pH control, combustion control, engine control, and flight control.

Ship Steering

Autopilots for ships are normally based on feedback from a heading measurement, using a gyrocompass, to a steering engine, which drives the rudder. It is common practice to use a control law of the PID type with fixed parameters. Although such a controller can be made to work reasonably well, its performance is poor in heavy weather and when the speed of the ship is changed. The reason is that the ship dynamics change with the speed of the ship and that the disturbances change with the weather. There is a growing awareness that autopilots can be improved considerably by taking these changes into account. This is illustrated by analysis of some simple models.

The ship dynamics are obtained by applying Newton's equations to the motion of the ship. For large ships the motion in the vertical plane can be separated from the other motions. It is customary to describe the horizontal motion by using a coordinate system fixed to the ship (see Fig. 9.8). Let V be the total velocity, let u and v be the x and y components of the velocity, and let r be the angular velocity of the ship.

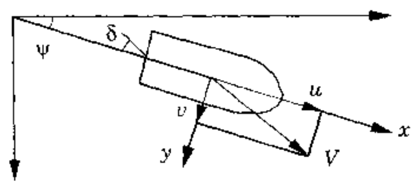

Figure 9.8 Coordinates and notations used to describe the equations of motion of ships.

In normal steering, the ship makes small deviations from a straight-line course. It is thus natural to linearize the equations of motion around the solution $u = u_0$, $v = 0$, $r = 0$, and $\delta = 0$. The natural state variables are the sway velocity v, the turning rate r, and the heading ψ. The following equations are obtained:

$$\begin{aligned}\frac{dv}{dt} &= (u/l)a_{11}v + ua_{12}r + (u^2/l)b_1\delta \\ \frac{dr}{dt} &= (u/l^2)a_{21}v + (u/l)a_{22}r + (u^2/l^2)b_2\delta \\ \frac{d\psi}{dt} &= r\end{aligned} \tag{9.9}$$

where u is the constant forward velocity and l is the length of the ship.

The parameters in the state equation (Eqs. 9.9) are surprisingly constant for different ships and different operating conditions (see Table 9.1). The transfer function from rudder angle δ to heading ψ is easily determined from Eqs. (9.9). The following result is obtained:

$$G(s) = \frac{K(1+sT_3)}{s(1+sT_1)(1+sT_2)} \tag{9.10}$$

where

$$\begin{aligned}K &= K_0u/l \\ T_i &= T_{i0}l/u \qquad i = 1,2,3\end{aligned} \tag{9.11}$$

The parameters K_0 and T_{i0} are also given in Table 9.1. Notice that they may change considerably even if the parameters of the state model do not change much. In many cases the model can be simplified to

$$G(s) = \frac{b}{s(s+a)} \tag{9.12}$$

where

$$\begin{aligned}b &= b_0\left(\frac{u}{l}\right)^2 = b_2\left(\frac{u}{l}\right)^2 \\ a &= a_0\left(\frac{u}{l}\right)\end{aligned} \tag{9.13}$$

Table 9.1 Parameters of models for different ships.

Ship	Mine-sweeper	Cargo	Tanker Full	Tanker Ballast
Length (m)	55	161	350	
a_{11}	−0.86	−0.77	−0.45	−0.43
a_{12}	−0.48	−0.34	0.43	−0.45
a_{21}	5.2	−3.39	−4.1	−1.98
a_{22}	−2.4	−1.63	−0.81	−1.15
b_1	0.18	0.17	0.10	0.14
b_2	−1.4	−1.63	−0.81	−1.15
K_0	2.11	−3.86	0.83	5.88
T_{10}	−8.25	5.66	−2.88	−16.91
T_{20}	0.29	0.38	0.38	0.45
T_{30}	0.65	0.89	1.07	1.43
a_0	−0.14	0.19	−0.28	0.06
b_0	−1.4	−1.63	−0.81	−1.15

This model is called the *Nomoto model*. Its gain b can be expressed approximately as follows:

$$b = c\left(\frac{u}{l}\right)^2\frac{Al}{D} \tag{9.14}$$

where D (in cubic meters) is the displacement, A (in square meters) is the rudder area, and c is a parameter whose value is approximately 0.5. The parameter a will depend on trim, speed, and loading. Its sign may change with the operating conditions.

A ship is influenced by disturbances due to wind, waves, and currents. The effects of these can be described as additional forces. Reasonable models have constant, periodic, and random components. The disturbances due to waves are typically periodic. The period may vary with the speed of the ship and its orientation relative to the waves.

The effects of parameter variations can be seen from the linearized models in Eqs. (9.9), (9.10), and (9.12). First, consider variations in the speed of the ship. It follows from Eqs. (9.11) and (9.13) that the gain is proportional to the square of the velocity and that the time constants are inversely proportional to the velocity. A reduction to half-speed thus reduces the gain to a quarter of its value and doubles the time constants.

The gain is essentially determined by the ratio of the rudder forces to the moment of inertia. Thus the relative water velocity at the rudder is what determines the gain. This velocity is influenced by waves and currents. The relative velocity may decrease drastically when there are large waves coming from behind and the ship is riding on the waves. The relative velocity may be very small or even zero. Controllability is then lost because there is no rudder force. The situation is even worse if the waves are not hitting the ship straight from behind, because the waves will then generate torques that tend to turn the ship.

The ship dynamics are also influenced by other factors. The hydrodynamic forces, and consequently also the parameters a_{ij} and b_j in the linearized model of Eqs. (9.9), depend on trim loading and water depth. This may be seen from Table 9.1, which gives parameters for a tanker under different loading conditions. Some consequences of the parameter variations are illustrated by an example.

EXAMPLE 9.6 Ship steering

Assume that the ship steering dynamics can be approximated by the Nomoto model of Eq. (9.12) and that a controller of PD type with the transfer function

$$G_r(s) = K(1 + sT_d)$$

is used. The loop transfer function is

$$G(s)G_r(s) = \frac{Kb(1 + sT_d)}{s(s + a)}$$

The characteristic equation of the closed-loop system is

$$s^2 + s(a + bKT_d) + bK = 0$$

The relative damping is

$$\zeta = \frac{1}{2}\left(\frac{a}{\sqrt{bK}} + T_d\sqrt{bK}\right)$$

The damping will depend on the speed of the ship. Assume that the model of Eq. (9.12) has the values a_{nom} and b_{nom} at the nominal speed u_{nom}. The variable u_{nom} is the nominal velocity used to design the feedback. Assume that u is the actual constant velocity. Using the speed dependence of a and b given by Eqs. (9.13) gives

$$a = a_{nom}\frac{u}{u_{nom}}$$
$$b = b_{nom}\left(\frac{u}{u_{nom}}\right)^2$$

This gives the damping

$$\zeta = \frac{1}{2}\left(\frac{a_{nom}}{\sqrt{Kb_{nom}}} + \frac{u}{u_{nom}}T_d\sqrt{Kb_{nom}}\right)$$

Consider an unstable tanker with

$$a_{nom} = -0.3$$
$$b_{nom} = 0.8$$
$$K = 2.5$$
$$T_d = 0.86$$

This gives $\zeta = 0.5$ and $\omega = 1.4$ at the nominal velocity. Furthermore,

$$\omega = 1.4u/u_{nom}$$
$$\zeta = -0.11 + 0.61u/u_{nom}$$

The closed-loop characteristic frequency and damping will thus decrease with decreasing velocity. The closed-loop system becomes unstable when the speed of the ship decreases to $u = 0.17u_{nom}$.

By scaling the parameters of the autopilot according to speed, it is possible to obtain closed-loop performance that is less sensitive to speed variations. The scaling of the parameters of the controller depends on the control goal. One design criterion is time invariance; that is, the time response of the ship should always be the same. If true time invariance is desired, the controller gains should be inversely proportional to the square of the speed. Path invariance is another criterion. In this case the path on the map is always the same. The gains should then be inversely proportional to the velocity of the ship. The gains are limited at low speed to avoid large rudder motions. □

pH Control

Control of pH (the concentration of hydrogen ions) is a well-known control problem that presents difficulties due to large variations in process dynamics. The problem is similar to the simple concentration control problem in Example 9.3. The main difficulty arises from a static nonlinearity between pH and concentration. This nonlinearity depends on the substances in the solution and on their concentrations.

The pH number is a measure of the concentration or, more precisely, the activity of hydrogen ions in a solution. It is defined by

$$\mathrm{pH} = -\log[\mathrm{H}^+] \tag{9.15}$$

where $[H^+]$ denotes the concentration of hydrogen ions. The formula (9.15) is, strictly speaking, not correct, since $[H^+]$ has the dimension of concentration, which is measured in the unit M = mol/l. The correct version of Eq. (9.15) is thus $\mathrm{pH} = -\log([\mathrm{H}^+]f_H)$, where f_H is a constant with the dimension liters per mole. The formula of Eq. (9.15) will be used here, however, because it is universally accepted in textbooks of chemistry.

Water molecules are dissociated (split into hydrogen and hydroxyl ions) according to the formula

$$H_2O \rightleftharpoons H^+ + OH^-$$

In chemical equilibrium the concentration of hydrogen H^+ (or rather H_3O^+) and hydroxyl OH^- ions are given by the formula

$$\frac{[H^+][OH^-]}{[H_2O]} = \text{constant} \tag{9.16}$$

Only a small fraction of the water molecules are split into ions. The water activity is practically unity, and we get

$$[H^+][OH^-] = K_w \tag{9.17}$$

where the equilibrium constant K_w has the value $10^{-14}\,[(\text{mol/l})^2]$ at 25°C. The main nonlinearity of the pH control problem will now be discussed.

EXAMPLE 9.7 Titration curve for a strong acid-base pair

Consider neutralization of m_A mol of hydrochloric acid HCl by m_B mol of sodium hydroxide NaOH in a water solution. The following reaction takes place:

$$HCl + NaOH \rightarrow H^+ + OH^- + Na^+ + Cl^-$$

Let the total volume be V. The concentration of chloride ions is then

$$[Cl^-] = x_A = m_A/V$$

and the concentration of sodium ions is given by

$$[Na^+] = x_B = m_B/V$$

because the acid and the base are completely ionized. Since the number of positive ions equals the number of negative ions, it follows that

$$x_A + [OH^-] = x_B + [H^+]$$

The concentration of hydroxyl ions can be related to the hydrogen ion concentration by Eq. (9.17). Hence

$$x = x_B - x_A = [OH^-] - [H^+] = \frac{K_w}{[H^+]} - [H^+] = 10^{\text{pH}-14} - 10^{-\text{pH}} \tag{9.18}$$

Solving for $[H^+]$ gives

$$[H^+] = \sqrt{x^2/4 + K_w} - x/2$$
$$[OH^-] = \sqrt{x^2/4 + K_w} + x/2$$

This gives

$$\text{pH} = f(x) = -\log\left(\sqrt{x^2/4 + K_w} - x/2\right) \tag{9.19}$$

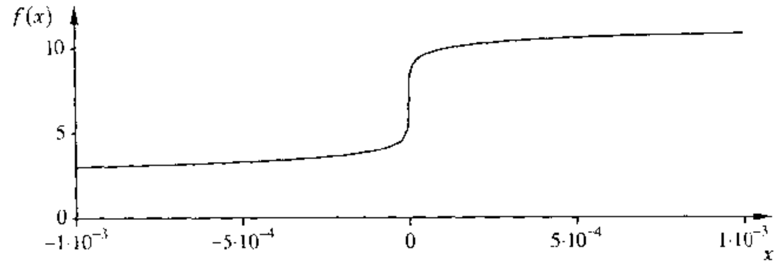

Figure 9.9 Titration curve of Eq. (9.19) for neutralization of a 0.001 M solution of HCl with a 0.001 M solution of NaOH.

The graph of the function f is called the *titration curve*. It is the fundamental nonlinearity for the neutralization problem. An example of the titration curve is shown in Fig. 9.9, which shows that there is considerable variation in the slope of the titration curve. The abscissa of the titration curve in Fig. 9.9 is given in terms of the concentration difference $x_B - x_A$. The x-axis can also be recalibrated into the amount of the reagent.

The derivative of the function f is given by

$$f'(x) = \frac{{}^{10}\log e}{2\sqrt{x^2/4 + K_w}} = \frac{{}^{10}\log e}{10^{\text{pH}-14} + 10^{-\text{pH}}} \tag{9.20}$$

The derivative has its largest value $f' = 2.2 \cdot 10^6$ for pH = 7. It decreases rapidly for larger and smaller values of pH. For pH = 4 and 10 we have $f' = 4.3 \cdot 10^3$. The gain can thus vary by several orders of magnitude. □

Figure 9.9 shows that the pH of a strong acid that is almost neutralized may change very rapidly if only a small amount of base is added. The reason for this is that strong acids and bases are completely dissociated. A weak acid is not completely dissociated, so it can absorb hydrogen ions by converting them to undissociated acid. It can also create hydrogen ions by dissociating acid molecules. This means that a weak acid or a weak base has an ability to resist changes in pH. This property is called *buffering*. The titration curve of a solution that contains weak acids or bases will therefore be less steep than the titration curves of strong acids or bases.

Example 9.7 shows that there will be a severe nonlinearity in the system due to the titration curve. An additional example illustrates the difficulties in controlling such a system.

EXAMPLE 9.8 pH control

Consider the problem of controlling the pH of an acid effluent that is fed to a stirred tank with volume V (in liters) and neutralized with NaOH. Let c_A (in moles per liter) be the concentration of acid in the influent stream, and let q

(in liters per second) be the flow of the effluent. Let c_B (in moles per liter) be the concentration of the reagent. Assume that the reagent concentration is so high that the reagent flow u (in liters per second) is negligible in comparison with q. The system is modeled by a linear dynamic model, which describes the mixing dynamics as if there were no reactions, and a static nonlinear titration curve, which gives pH as a function of the concentrations. Let x_A and x_B be the concentrations of acid and base in the tank if there were no chemical reactions. Mass balances then give

$$\begin{aligned}\frac{dx_A}{dt} &= \frac{q}{V}(c_A - x_A)\\ \frac{dx_B}{dt} &= \frac{u}{V}c_B - \frac{q}{V}x_B\end{aligned} \tag{9.21}$$

The pH is given by Eq. (9.19). It is further assumed that the dynamics of the pH sensor and the pump together can be described by the transfer function

$$G(s) = \frac{1}{(1+sT)^2}$$

A simple calculation indicates the difficulties in the control problem. Assuming proportional control with gain k, the linearized loop transfer function from the error in pH to pH becomes

$$G_0(s) = \frac{c_B k f'}{q(1+sT_m)(1+sT)^2}$$

where T_m is the mixing time constant

$$T_m = V/q$$

and f' is the slope of the titration curve given by Eq. (9.20). The critical gain for stability is

$$k_c = \frac{q}{f' c_B T}(2 + T/T_m)(1 + T/T_m) \approx \frac{2q}{f' c_B T}$$

where the approximation holds for $T \ll T_m$. Since the slope of the titration curve varies drastically with pH, the critical gain will vary accordingly. Some values for different values of the pH of the mixture are:

pH	Critical gain
7	0.009
8	0.046
9	0.46
10	4.6

To make sure that the closed-loop system is stable for small perturbations around an equilibrium of pH = 7, the gain should thus be less than 0.009. A reasonable value of the gain for operation at pH = 8 is $k = 0.01$, but this

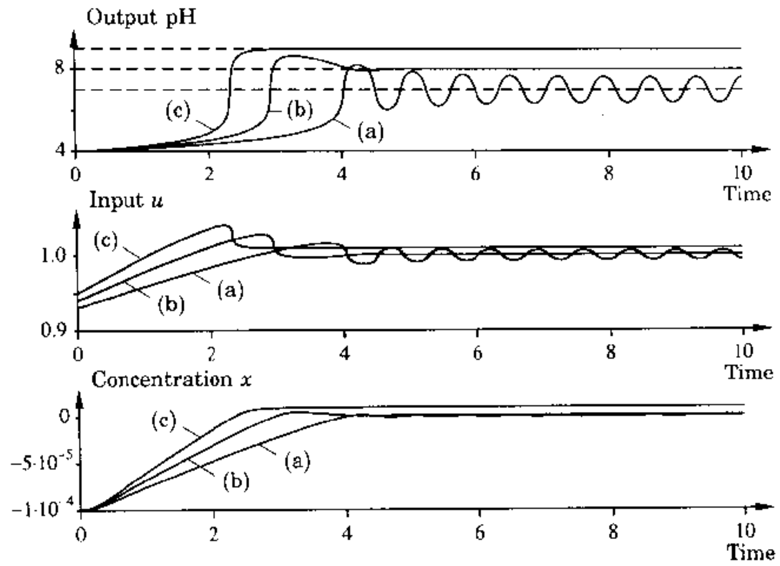

Figure 9.10 Output pH and control signal when the process in Example 9.8 is controlled by using a PI controller when pH_{ref} is (a) 7; (b) 8; (c) 9.

gain will give an unstable system at pH = 7 and is too low for a reasonable response at pH = 9. Figure 9.10 shows PI control with gain 0.01 and reset time 1. The process is started at equilibrium pH = 4. The reference value is then changed to 7, 8, and 9.

The calculations and the simulation illustrate the key problems with pH control. The difficulties are compounded by the presence of time delays and flow variations. One way to get around the problem is to use the concentration x as the output rather than pH. Figure 9.11 shows a possible control scheme in which the measured pH and the reference value of pH are transformed into equivalent concentrations. This means that the variable x is computed for the

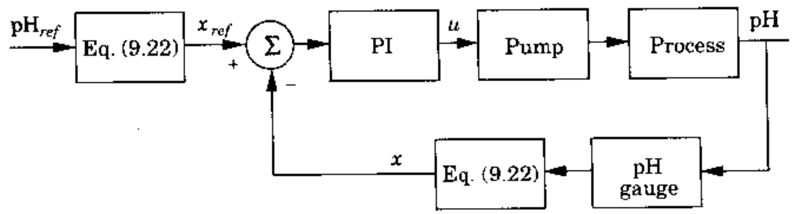

Figure 9.11 Control configuration for the pH control problem in Example 9.8.

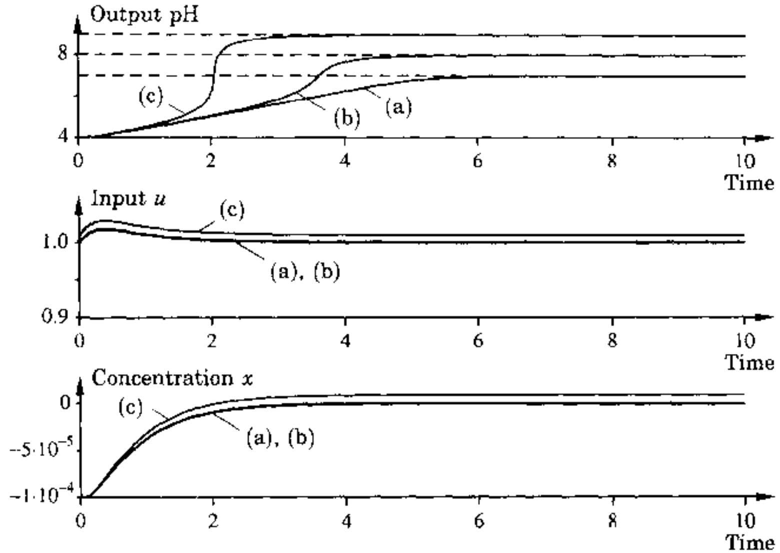

Figure 9.12 The same experiment as in Fig. 9.10, but with the controller structure in Fig. 9.11. The gain of the controller is 1000, and the reset time is 1. (a) $\text{pH}_{ref} = 7$; (b) $\text{pH}_{ref} = 8$; (c) $\text{pH}_{ref} = 9$.

measured pH by the formula

$$x = f^{-1}(\text{pH}) = 10^{\text{pH}-14} - 10^{-\text{pH}} \tag{9.22}$$

The transfer function from u to x is

$$\frac{c_B}{q(1+sT_m)(1+sT)^2}$$

which is independent of the operating point. Figure 9.12 shows the same experiments as in Fig. 9.10, but with the control modification shown in Fig. 9.11. It should be noted that the nonlinear compensation with Eq. (9.22) can be used, since a strong acid-base pair is controlled. The more general problem of mixtures of many weak acids and bases does not have an easy linearizing transformation. It is then necessary to measure the concentrations of the components or to make an on-line measurement of the titration curve. Some other form of adaptation can then be reasonable. □

Combustion Control

In combustion control of a boiler it is important to adjust the oxygen content of the flue gases. The flow of combustion air depends on the burn rate in the

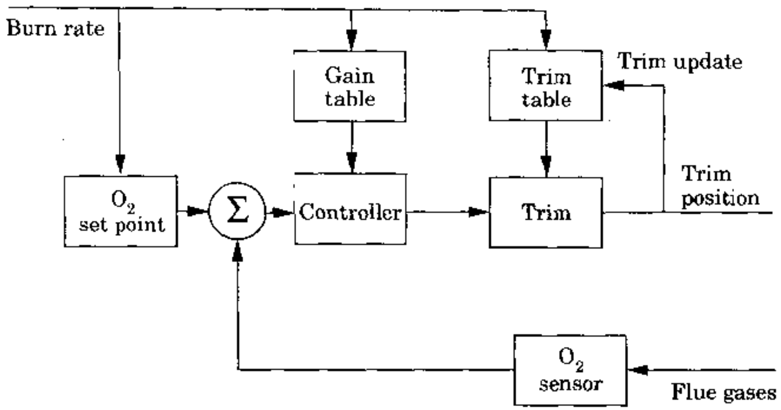

Figure 9.13 Adaptive feedforward and gain scheduling in an oxygen trim controller.

boiler. The measurement signal is the oxygen content in the exhaust stack, and the control signal is the trim position, which controls the flow of combustion air. There is a significant time delay between the input to the burner and the oxygen sensor in the exhaust stack. With a conventional controller there is then a loss of efficiency before the correct trim position is reached after a change in the burn rate. One configuration based on adaptive feedforward and gain scheduling is shown in Fig. 9.13. The working range of the boiler is divided into regions. For each region there is a memory (digital integrator). All integrators are zero initially. When the boiler starts to operate, the trim control will adjust the oxygen setpoint. When the setpoint level is achieved, the appropriate integrator is set to the correct trim position. A trim profile will be built up as the boiler works over its range. When the boiler returns to a position at which the integrator is set, the stored trim value is instantly fed to the trim drive actuator, thus eliminating the lag from the control loop. If the fuel changes, the trim profile is updated automatically. The controller thus works with an adaptive feedforward compensation from the burn rate. There is also a gain scheduling of the loop gain of the controller to get tight control under all firing conditions. This gain schedule is built up in commissioning the controller.

Fuel-Air Control in a Car Engine

A schematic drawing of a microcomputer control system for a car engine is shown in Fig. 9.14. The accelerator is connected to the throttle valve. The fuel injection is governed by a table lookup controller. The control variable, which

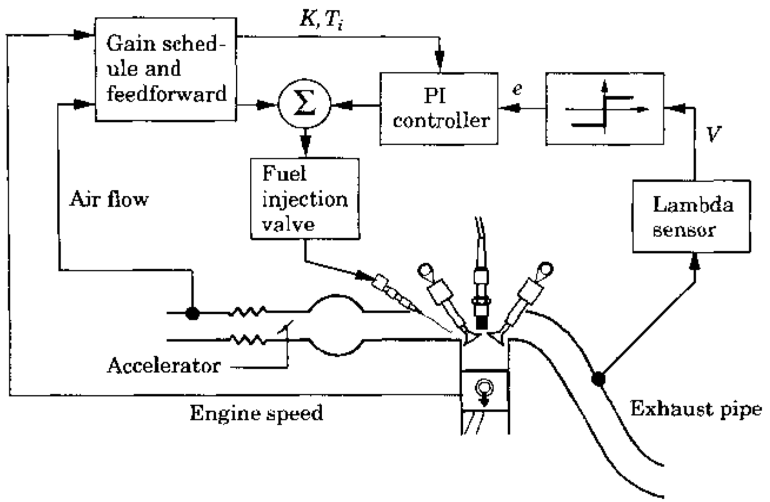

Figure 9.14 Schematic diagram of a microcomputer engine control system.

is the opening time for the fuel injection valve, is controlled by a combination of feedforward and feedback. The feedforward signal is a nonlinear function of engine speed and load. The load is represented by the air flow, which can be measured by using a hot wire anemometer. In one common system the table has 16 × 16 entries with linear interpolation. There is also feedback in the system from an exhaust oxygen sensor. The fuel-air ratio is measured by using a zirconium oxide catalytic sensor called the *lambda sond*. This sensor gives

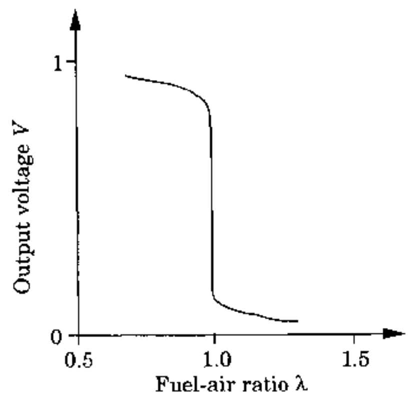

Figure 9.15 The characteristic of a lambda sond.

an output that changes drastically when the fuel-air ratio is 1. A typical sensor characteristic is shown in Fig. 9.15. The lambda sond is positioned after the exhaust manifold in an excess oxygen environment, where the exhaust gas from all the cylinders is mixed. This creates a delay in the feedback loop. Notice the feedforward path via the table discussed earlier in the paragraph. The feedback has a special form; continuous control cannot be used because of the strongly nonlinear characteristics of the lambda sond. The error signal is formed by normalizing the output of the lambda sensor as follows:

$$e = \begin{cases} 1 & \text{if } V > 0.5 \\ -1 & \text{if } V \le 0.5 \end{cases}$$

The error signal is thus positive if the fuel-air ratio is low (lean mixture) and negative when the ratio is high (rich mixture). The error signal is sent to a PI controller whose gain and integration time are set from the scheduling table. The values are set on the basis of load (air flow) and engine speed. The gain

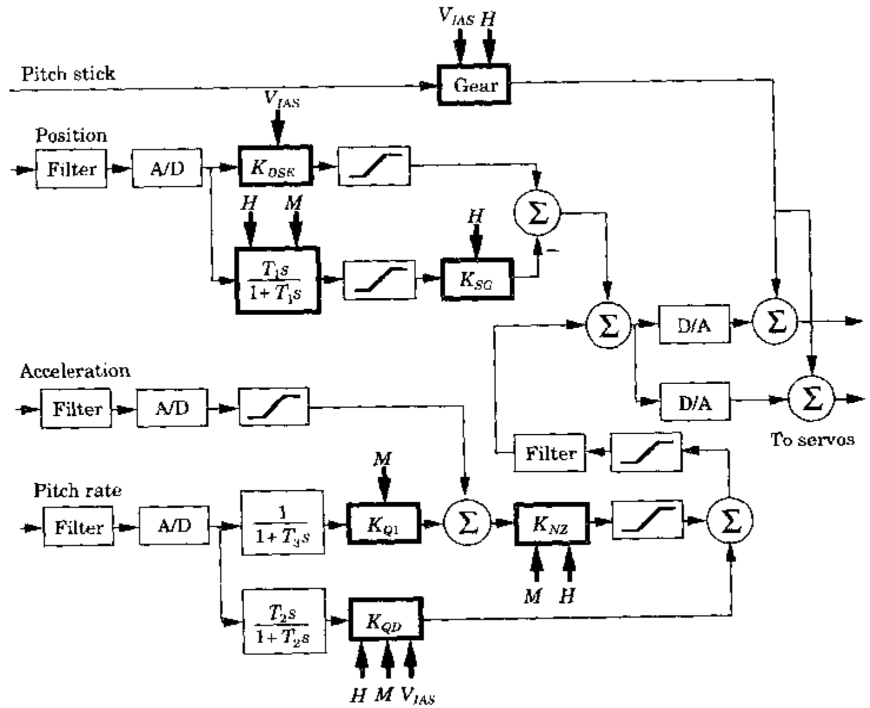

Figure 9.16 Simplified block diagram of the pitch control of the autopilot for a supersonic aircraft. The highlighted blocks show the parts of the autopilot where gain scheduling is used.

schedule is implemented simply by adding entries for the gain and integration time to the table used for feedforward of the nominal control variable. Because of the relay characteristic, there will be an oscillation in the fuel-air ratio. This is beneficial, because the catalytic sensor needs a variation to operate properly. The amplitude and the frequency of the oscillation are determined by the parameters of the controller.

Flight Control Systems

Figure 9.16 shows a block diagram of the pitch channel of a flight control system for a supersonic aircraft. The pich stick signal is the command signal from the pilot. Position, acceleration, and pitch rate are feedback signals. There are three scheduling variables: height H, indicated airspeed V_{IAS}, and Mach number M. The parameters of the controller that are scheduled are drawn as boxes; the arrows indicate the scheduling variables. The schedule for the gain K_{QD} is given by

$$K_{QD} = K_{QD_{IAS}} + (K_{QD_H} - K_{QD_{IAS}})MF$$

where $K_{QD_{IAS}}$ is a function of indicated airspeed V_{IAS} (shown in Fig. 9.17) and K_{QD_H} is a function of height (also shown in Fig. 9.17). The variable MF is given by

$$MF = \frac{1}{s+1} K_{MF}$$

where K_{MF} is a function of the Mach number and s is the Laplace transform variable.

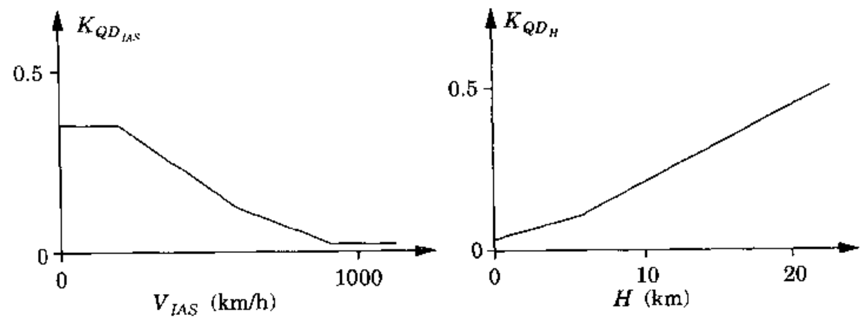

Figure 9.17 Scheduling functions. The function $K_{QD_{IAS}}$ is also different for different flight modes.

9.6 CONCLUSIONS

Gain scheduling is a good way to compensate for known nonlinearities. With such a scheme the controller reacts quickly to changing conditions. One drawback of the method is that the design may be time-consuming if it is not possible to use nonlinear transformations or auto-tuning. Another drawback is that the controller parameters are changed in open loop, without feedback from the performance of the closed-loop system. This makes the method impossible to use if the dynamics of the process or the disturbances are not known accurately enough.

Example 9.3 and the ship steering example in Section 9.5 show that it is often useful to introduce normalized variables. The processes then become constant in the new variables, and the gain scheduling of the controllers is easily derived.

PROBLEMS

9.1 Simulate the tank system in Example 9.2. Let the tank area vary as

$$A(h) = A_0 + h^2$$

Further assume that $a = 0.1A_0$.

(a) Study the behavior of the closed-loop system when the full gain schedule is used and when the modified gain schedule is used.

(b) Study the sensitivity of the system to changes in the parameters of the process.

(c) Study the sensitivity of the closed-loop system to noise in the measurement of the level.

9.2 Consider the concentration control problem in Example 9.3. Design a fixed sampled-data controller with a fixed sampling period for the system. Compare it with a controller based on the time-scaled model.

9.3 A model of a ship is given in Section 9.5. Show that the two scalings suggested in Example 9.6 correspond to the time invariance and the path invariance behavior of the ship.

9.4 The simulations in Example 9.8 are done by using the model of Eqs. (9.21) and (9.19) with $q = 1000$, $V = 1000$, $T = 0.1$, and $K_w = 10^{-14}$. The controller is a PI controller with gain 0.01 and reset time 1. Verify the simulations in Fig. 9.10 and Fig. 9.12.

9.5 Consider the ship steering problem in Example 9.6. Simulate the closed-loop system, and determine the sensitivity with respect to the speed of the ship.

9.6 The controller in Example 9.3 gives a control that is equal when measured in terms of the number of sampling intervals but not when measured in terms of time. Suggest and test possibilities to get the same time responses independent of the flow through the tank.

REFERENCES

The use of gain scheduling in aircraft control is discussed in:

Stein, G., 1980. "Adaptive flight control: A pragmatic view." In *Applications of Adaptive Control*, eds. K. S. Narendra and R. V. Monopoli. New York: Academic Press.

A typical application of gain scheduling and compensation of nonlinearities in the process industry is given in:

Whatley, M. J., and D. C. Pott, 1984. "Adaptive gain improves reactor control." *Hydrocarbon Processing* May: 75–78.

Nonlinear transformations in a general context were originally discussed by using geometric control theory in:

Krener, A. J., 1973. "On the equivalence of control systems and the linearization of nonlinear systems." *SIAM J. Control* **11**: 670.

Brockett, R. W., 1978. "Feedback invariants for nonlinear systems." *Preprint 7th IFAC World Congress*, pp. 1115–1120. Helsinki, Finland.

Necessary and sufficient conditions under which transformations from nonlinear to linear systems exist are given in:

Su, R., 1982. "On the linear equivalents of nonlinear systems." *Systems & Control Letters* **2**: 48.

Hunt, L. R., R. Su, and G. Meyer, 1983. "Design for multiinput systems." In *Differential Geometric Control Theory Conference*, eds. R. W. Brockett, R. S. Millman, and H. J. Sussman, pp. 268–298. Boston: Birkhauser.

Linearizing control is discussed, for instance, in:

Isidori, A., 1989. *Nonlinear Control Systems: An Introduction* (2nd ed.). Berlin: Springer-Verlag.

Nijmeijer, H., and A. J. van der Schaft, 1991. *Nonlinear Dynamical Control Systems*. New York: Springer-Verlag.

A neat application for design of a flight control system for a helicopter was made by:

Meyer, G., R. Su, and L. R. Hunt, 1984. "Application of nonlinear transformations to automatic flight control." *Automatica* **20**: 103–107.

Applications of the same idea in simpler setting are given in:

Orava, P. J., and A. J. Niemi, 1974. "State model and stability analysis of a pH control process." *Int. J. Control* **20**: 557–567.

Källström, C. G., K. J. Åström, N. E. Thorell, J. Eriksson, and L. Sten, 1979. "Adaptive autopilots for tankers." *Automatica* **20**: 241–254.

Niemi, A. J., 1981. "Invariant control of variable flow processes." *Proceedings of the 8th IFAC World Congress*, pp. 2687–2692. Kyoto, Japan.

CHAPTER 10

ROBUST AND SELF-OSCILLATING SYSTEMS

10.1 WHY NOT ADAPTIVE CONTROL?

In previous chapters we showed that adaptive control can be very useful and can give good closed-loop performance. However, that does not mean that adaptive control is the universal tool that should always be used. A control engineer should be equipped with a variety of tools and the knowledge of how to use them. A good guideline is to use the simplest control algorithm that satisfies the specifications. Robust high-gain control should definitely be considered as alternatives to adaptive control algorithms. Section 10.2 treats robust high-gain control. The self-oscillating adaptive system (SOAS) is presented in Section 10.3. This is a special class of adaptive systems with strong ties to high-gain control and auto-tuning. Relay feedback is a key ingredient of the SOAS. Another class of switching systems, variable-structure systems, is discussed in Section 10.4. Variable-structure systems have been developed mainly in the Soviet Union and can be regarded as a generalization of the SOAS. Conclusions are given in Section 10.5.

10.2 ROBUST HIGH-GAIN FEEDBACK CONTROL

Some design methods deal explicitly with process uncertainties. One powerful method has been developed by Horowitz. This procedure, which has its origin in Bode's classical work on feedback amplifiers, is based on several ideas. The specifications are expressed in terms of the transfer function from command

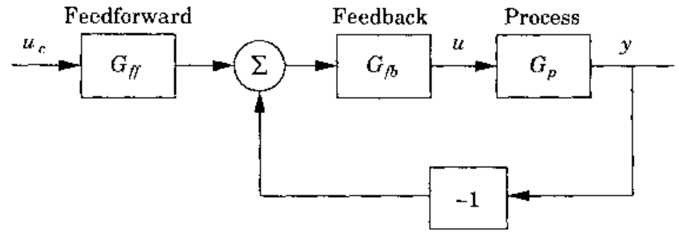

Figure 10.1 A two-degree-of-freedom system.

signal to process output. The plant is characterized by its nominal transfer function. For each frequency it is also assumed that the process uncertainty is known in terms of variations in amplitude and phase. A solution is determined in terms of a controller with a feedback G_{fb} and a feedforward G_{ff}, as shown in Fig. 10.1. Such a configuration is called a *two-degree-of-freedom* system because there are two transfer functions to be determined.

Several other design methods can be used to design robust controllers. One technique is based on LQG design. By adjusting the weighting matrices in the LQG problem, a *loop transfer recovery (LTR)* is achieved. This design procedure can cope with phase uncertainty at high frequencies. The key idea is to keep the loop gain less than 1 at high frequencies, where the phase error is large.

In Horowitz's procedure the feedback transfer function G_{fb} is first determined such that the closed-loop uncertainty is within the specified limits. The nominal value of the transfer function is then modified by the feedforward compensation G_{ff}. The method is based on graphical constructions using the Nichols chart. It gives a high-order linear compensator that can cope with the specified plant uncertainty. The procedure attempts to keep the loop gain as low as possible. A key idea in the Horowitz design method is the observation that a system in which the Nyquist curve is close to a straight line through the origin can tolerate a significant change of gain. The response time will change with the gain, but the shape of the response will remain invariant. For minimum-phase systems with a known pole excess it is always possible to find a frequency range in which the phase is constant. By proper compensation it is then possible to obtain a loop gain at which the Nyquist curve is close to a straight-line segment. The assumption that the pole excess is known implies that the phase of the system is known for high frequencies. This is not always a realistic assumption. The Horowitz design method was originally developed for structured perturbations but has also been extended to unstructured uncertainties.

The main step in the procedure is to determine the tolerances for the gain in the closed-loop transfer function. The plant uncertainties are specified as gain and phase variations of the plant transfer function at different frequencies. The given tolerances and uncertainties are used to calculate constraints for the open-loop transfer function. The feedback compensator G_{fb} is then designed such that the compensated open-loop system satisfies the tolerances. This is usually an iterative procedure, which can conveniently be done graphically by using a Nichols chart. Finally, the prefilter G_{ff} is designed such that the closed-loop specifications are fulfilled. This may be done by using the Bode diagram.

The major drawback of the method is that it is impossible to know *a priori* whether the desired closed-loop specifications are attainable. It is thus a trial-and-error method, but the iterations give the designer insight into the tradeoffs between different specifications, such as closed-loop sensitivity, complexity of the controller, and measurement noise amplification.

EXAMPLE 10.1 An industrial robot arm

A simple model of a robot arm is used in this example. The transfer function from the control input (motor current I) to measurement output (motor angular velocity ω) is

$$G_p(s) = \frac{k_m(J_a s^2 + ds + k)}{J_a J_m s^3 + d(J_a + J_m)s^2 + k(J_a + J_m)s}$$

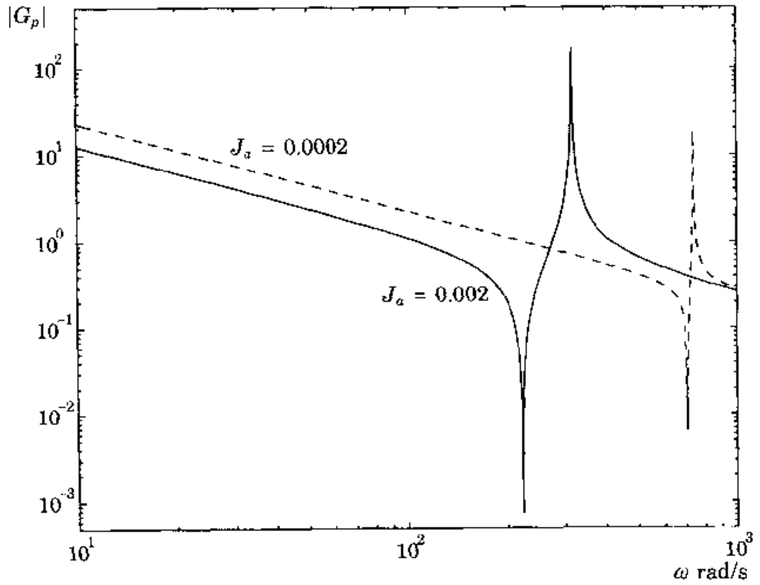

Figure 10.2 Bode plots for the robot arm in Example 10.1 for $J_a = 0.0002$ and $J_a = 0.002$.

with $J_a \in [0.0002, 0.002]$, $J_m = 0.002$, $d = 0.0001$, $k = 100$, and $k_m = 0.5$. The moment of inertia J_a of the robot arm varies with the arm angle. Bode plots of the plant gain for the extreme values of the arm inertia J_a are given in Fig. 10.2. The purpose of the control system is to control the angular velocity step responses at various arm angles. The aim is to get a closed-loop system with a bandwidth between 15 and 40 Hz. The disturbance rejection specification has been set to 6 dB. A feedback compensator that satisfies the specifications is

$$G_{fb}(s) = \frac{125(1 + s/50)(1 + s/300)}{s(1 + s/800)(1 + s/5000)}$$

This compensator is essentially a PI controller with a lead filter. The final prefilter has the transfer function

$$G_{ff}(s) = \frac{1 + s/1000}{(1 + s/26)(1 + s/200)(1 + s/200)}$$

Simulated responses are shown in Figs. 10.3 and 10.4.

To make a comparison, an adaptive controller is also designed for the process. In this particular problem the essential uncertainty is in one parameter only, the moment of inertia. It is then natural to try to make a special adaptive design in which only this parameter is estimated.

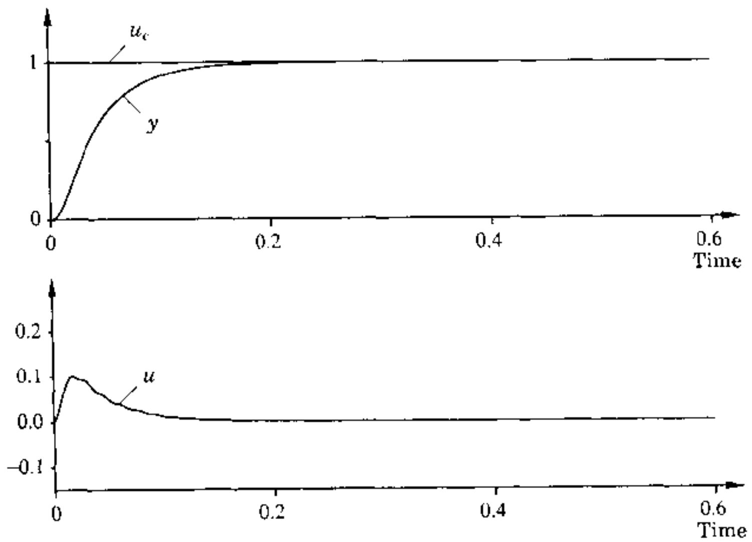

Figure 10.3 Simulation of the step response with the arm inertia $J_a = 0.0002$ for the robust system.

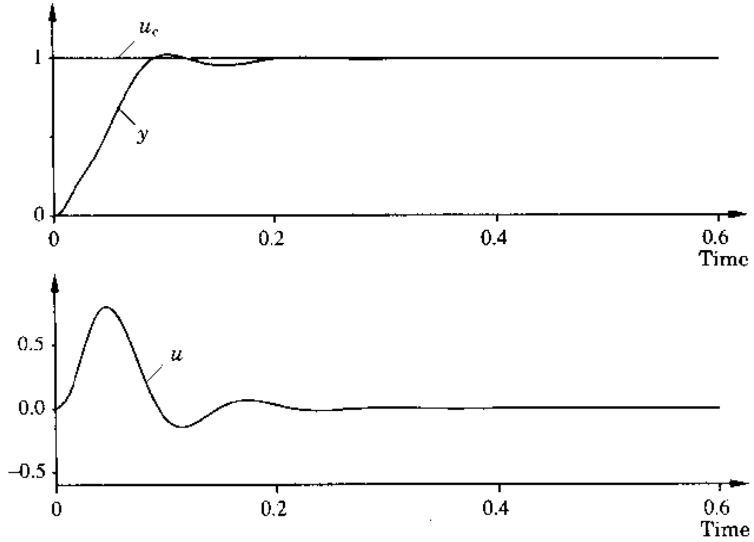

Figure 10.4 Simulation of the step response with the arm inertia $J_a = 0.002$ for the robust system.

The adaptive controller is designed on the basis of a simplified model. If we neglect the elasticity in the robot arm, the system can be described by

$$J\frac{d\omega}{dt} = k_m I \tag{10.1}$$

where $J = J_a + J_m$ is the total moment of inertia and k_m the current gain of the motor. The plant of Eq. (10.1) can be controlled adequately with a PI controller. The controller parameters can be chosen to be

$$K = \frac{2\zeta_0\omega_0 J}{k_m}$$

$$T_i = \frac{2\zeta_0}{\omega_0}$$

This gives the following characteristic equation for the closed-loop system:

$$s^2 + 2\zeta_0\omega_0 s + \omega_0^2 = 0$$

The controller parameters are thus related to the model by simple equations. Notice that the integration time T_i does not depend on the moment of inertia of the robot arm and that the controller gain K should be proportional to the moment of inertia.

A root-locus calculation indicates that the design based on the simplified model will work well if

$$\omega_0 < \omega_{crit} = \zeta_0\left(\frac{kJ_m}{J_a^2}\right)^{1/2}$$

The most critical case occurs for $J_a = 0.002$. It implies that ω_0 must be less than 200 rad/s.

The fact that the design is based on a simplified model limits the closed-loop bandwidth. A fast response to command signals can still be obtained by use of feedforward compensation. For this purpose, let the desired response to angular velocity commands be given by

$$G_m(s) = \frac{\omega_m^2}{s^2 + 2\zeta_0\omega_m s + \omega_m^2}$$

The feedforward controller can now be designed such that the closed-loop system gets the desired response.

An adaptive system can be obtained simply by estimating the total moment of inertia by applying recursive least squares to the model of Eq. (10.1) and feeding the estimate into the above design equation. To estimate the parameters of the continuous-time model of Eq. (10.1), it is necessary to introduce

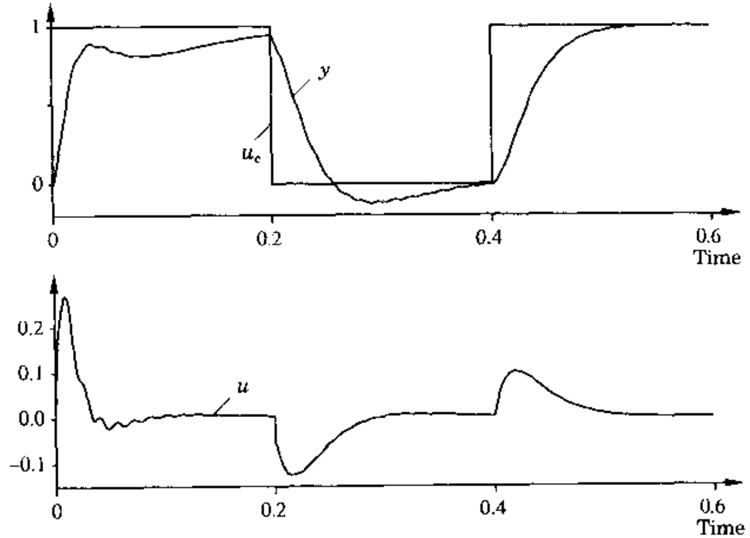

Figure 10.5 Simulation of the tailored adaptive systems response with the arm inertia $J_a = 0.0002$. The controller is initially tuned for $J_a = 0.002$.

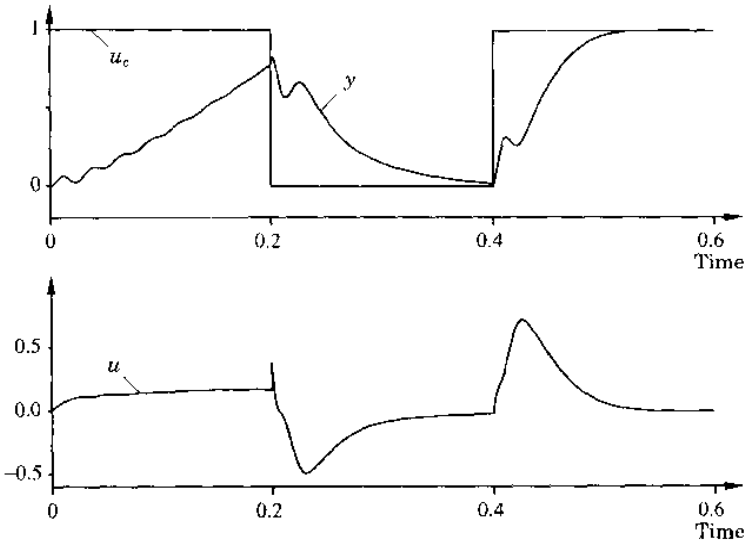

Figure 10.6 Simulation of the tailored adaptive systems response with the arm inertia $J_a = 0.002$. The controller is initially tuned for $J_a = 0.0002$.

filtering. This is done by integrating Eq. (10.1) over the time interval $(t, t+h)$:

$$\omega(t+h) - \omega(t) = \frac{k_m}{J} \int_t^{t+h} I(s)\, ds$$

A least-squares estimator of J is easily constructed from this equation. This estimate is then used in the PI control law. Simulations of the system are shown in Figs. 10.5 and 10.6. The parameter h was chosen to be 0.1 s. The figures show that the system adapts to a good response after two transients. Notice the magnitudes of the control signal for the cases of low and high inertia.

The controller structures for the robust and adaptive cases are quite similar by design. The feedback part of the robust controller is essentially a PI controller with a lead-lag filter. The parameters are $K = 2.5$ and $T_i = 0.02$. The lead-lag filter increases the controller gain to 6.7 at frequencies around 500 rad/s. The feedback part of the adaptive controller is also a PI controller, but the parameters are adjustable. They range from $K = 0.15$ and $T_i = 0.07$ for $J_a = 0.0002$ to $K = 1.05$ and $T_i = 0.07$ for $J_a = 0.002$. The feedback gain in the adaptive controller is thus 40 times smaller than the gain of the robust controller. This means that the effects of measurement noise are also much smaller for the adaptive controller. Both systems are designed to give the same response time to command signals. Notice, however, that feedforward is used in very different ways in the two systems. In the robust design, it is used

to decrease the response time to command signals; in the adaptive design, it is used to increase the response time. The reason is that the bandwidth of the closed inner loop is large in the robust design, to take care of the plant variations, whereas the adaptive design allows a low closed-loop bandwidth, since the uncertainty is eliminated. The responses of the adaptive system are better over the full parameter range when the parameters are adapted, but it will take some time for the parameters to adapt. The robust controller will have a better response when the parameters of the process are changing rapidly from one constant value to another. □

Comparison between Robust and Adaptive Control

The robust design method will generally give systems that respond more quickly when the parameters change, but it is important that the range of parameter variation be known. The adaptive controller responds more slowly but can generally handle larger parameter variations. The adaptive controller will give better responses to command signals and load variations when controller parameters have converged, provided that the model structure is sufficiently correct. The controllers designed by Horowitz's method will generally have high-loop gains, which make them more sensitive to noise.

10.3 SELF-OSCILLATING ADAPTIVE SYSTEMS

A system that is insensitive to parameter variations can be obtained by using a two-degree-of-freedom configuration with a high-gain feedback and a feedforward compensator (compare Section 10.2). This section introduces an adaptive technique to keep the gain in the feedback loop high by using a relay feedback. Relays combine the properties of high gain and inexpensive implementations. However, relays often introduce oscillations into the system.

The idea of the self-oscillating adaptive system (SOAS) originated in work at Honeywell on adaptive flight control in the late 1950s. The inspiration came from work on nonlinear systems by Flügge-Lotz at Stanford. Systems based on the idea were flight-tested in the F-94C, the F-101, and the X-15 aircraft. (See Fig. 1.2.) The idea has also been applied in process control, but the SOAS has not found widespread use. One reason is that substantial modifications of the basic scheme are necessary to make the systems work well. A characteristic feature of the SOAS is that there is a limit cycle oscillation. The system thus represents a type of adaptive control in which there are intentional perturbations, which excite the system all the time. The SOAS is one of the simplest systems with this property. The SOAS is based on three useful ideas: model-following, automatic generation of test signals, and use of a relay with a dither signal as a variable gain. The key result is that the loop gain is automatically adjusted to give an amplitude margin $A_m = 2$.

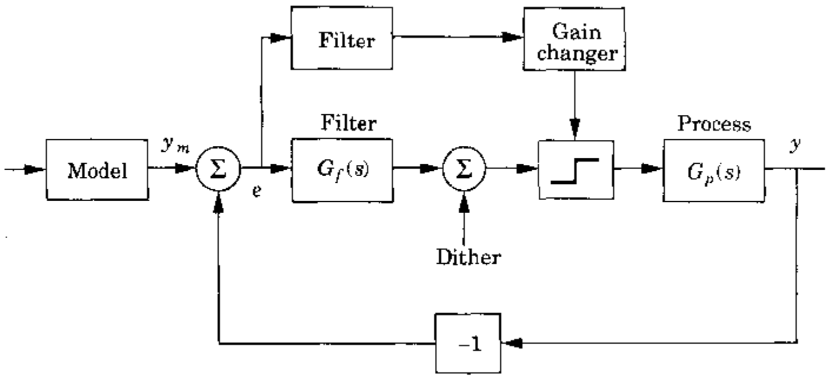

Figure 10.7 Block diagram of a self-oscillating adaptive system (SOAS).

Principles of the SOAS

Since we want to emphasize the ideas, we will limit the discussion to the basic version of the system. A block diagram of an SOAS is shown in Fig. 10.7. This is a two-degree-of-freedom system. There is a high-gain feedback loop around the process. The desired response to command signals is obtained by the reference model. Ideally, the high-gain loop will make the process output y follow the model output y_m. The response of the closed-loop system will be relatively insensitive to the variations in process dynamics because of the high loop gain. The system is thus a typical model-following design. The special feature is that the high-gain loop is nonlinear. The high-gain loop is supplemented with a feedforward model and a device to change the gain of the relay. The model determines the closed-loop response and the gain changer limits the amplitude of the limit cycle oscillation.

The High-Gain Loop

The feedback compensator contains a lead filter $G_f(s)$ and a relay. The relay is motivated by the desire to have as high a gain as possible. Because of the relay, there will be a limit cycle, whose amplitude is kept at a tolerable limit by adjusting the relay amplitude by a separate feedback loop. The relay gives a high gain for small inputs, and the gain decreases with increasing input amplitude. The key difficulty in the design of an SOAS is to find a suitable compromise between the limit cycle amplitude and the response speed. A low relay amplitude gives a limit cycle with a low amplitude but also a slow response speed. A large relay amplitude gives a rapid response but also a large amplitude of the limit cycle oscillation. The relations can to some extent be influenced by the lead filter.

Properties of the Basic SOAS

The limit cycle in a system with relay feedback was discussed in Section 8.6. This will now be used to analyze the self-oscillating adaptive system. Consider the system shown in Fig. 10.7 without the gain changer.

The relay is used to introduce a limit cycle oscillation in the system. The period and the amplitude of the oscillation can be determined by the methods discussed in Section 8.6. When the reference signal is changed, or when there are disturbances, there will also be other signals in the system, which will be superimposed on the limit cycle oscillations. The signals that appear in the system will thus be of the form

$$s(t) = a \sin \omega t + b(t)$$

where $a \sin \omega t$ denotes the limit cycle oscillation. The key to understanding the SOAS is to find out how signals of this type propagate in the system. It is straightforward to determine the transmission of the signal through the linear subsystems; the signal propagation through the relay is the main difficulty. This analysis will be simplified considerably if it is assumed that $b(t)$ varies much more slowly than $\sin \omega t$. Furthermore, assume that $b(t)$ is smaller than a. This should be true at least in steady state, since $b(t)$ is the difference between the model output and the process output.

The Dual-Input Describing Function

It is assumed that $b(t)$ varies so slowly that it can be approximated by a constant. The input signal to the relay is thus of the form

$$u(t) = a \sin \omega t + b$$

The relay input and output are shown in Fig. 10.8. The relay output can be expanded in a Fourier series

$$y(t) = bN_B + aN_A \sin \omega t + aN_{A_2} \sin 2\omega t + \cdots \tag{10.2}$$

where the numbers N_A and N_B are given by

$$\begin{aligned} N_B &= \frac{1}{2\pi b}\int_0^{2\pi} y(t)\,dt = \frac{d(\pi+\alpha) - d(\pi - 2\alpha) + \alpha d}{2\pi b} \\ &= \frac{4\alpha d}{2\pi b} = \frac{2\alpha d}{\pi b} = \frac{2d}{\pi b}\sin^{-1}\left(\frac{b}{a}\right) \end{aligned}$$

$$\begin{aligned} N_A &= \frac{1}{\pi a}\int_0^{2\pi} y(t)\sin\omega t\,dt = \frac{2d}{\pi a}\int_\alpha^{\pi-\alpha} \sin \omega t\,dt \\ &= \frac{4d}{\pi a}\cos\alpha = \frac{4d}{\pi a}\sqrt{1-(b/a)^2} \end{aligned}$$

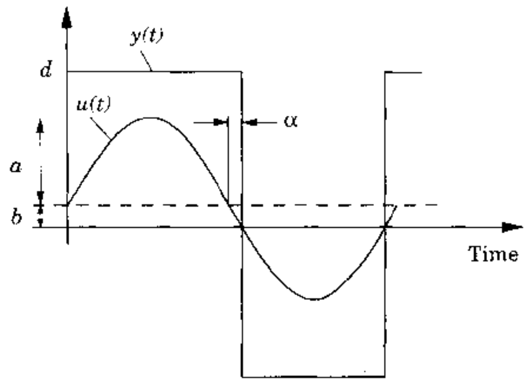

Figure 10.8 Relay inputs and outputs.

Small values of b/a give the approximations

$$N_A \approx \frac{4d}{\pi a} \qquad N_B \approx \frac{2d}{\pi a}$$

Notice that

$$N_A \approx 2N_B \tag{10.3}$$

The transmission of the constant level b and of the first harmonic $\sin \omega t$ are thus characterized by the equivalent gains N_B and N_A. Since the linear parts will normally attenuate high frequencies more than low frequencies, a reasonable approximation is often obtained by considering only the constant part and the first harmonic. The number N_B, which describes the propagation of a constant signal, is called the *dual-input describing function*, by analogy with the ordinary describing function that describes the propagation of sinusoids through static nonlinearities. Notice that the describing function N_B depends on a (the amplitude of the sinusoidal oscillation). This dependence is the key to understanding how the SOAS works. The dual-input describing function can be used to characterize the transmission of slowly varying signals. A detailed analysis of the accuracy of the approximation is fairly complicated. Let it therefore suffice to mention some rules of thumb for using the approximation. The ratio a/b should be greater than 3, and the ratio of the limit cycle frequency to the signal frequency should also be greater than 3. It is strongly recommended that the analysis be supplemented by simulation.

Main Result

The tools for explaining how the SOAS works are now available. Consider the system in Fig. 10.7. From Section 8.5 the period of the limit cycle is given by

Eqs. (8.8) when the describing function method is used. The amplitude of the limit cycle at the relay input is also given by Eqs. (8.8):

$$N_A |G(i\omega_u)| = 1 \tag{10.4}$$

The transmission of a sinusoidal signal through a relay can thus be approximately described by an equivalent gain, which is inversely proportional to the signal amplitude at the relay input. The amplitude thus automatically adjusts so that the loop gain is unity at the frequency ω_u.

Now consider the propagation of slowly varying signals superimposed on the limit cycle oscillations. The propagation of the signals through the linear parts of the system can be described by the transfer function $G(s)$. If the signals vary slowly in comparison with the limit cycle oscillations, the propagation through the relay is approximately described by the dual-input describing function N_B. The propagation of slowly varying signals is thus approximately described by the loop transfer function

$$G_0(s) = N_B(a)G(s)$$

It follows from Eqs. (10.3) and (10.4) that

$$|G_0(i\omega_u)| = N_B(a)|G(i\omega_u)| = \frac{1}{2} N_A |G(i\omega_u)| = 0.5$$

We thus obtain the following important result, which describes the operation of the SOAS.

RESULT 10.1 Amplitude margin of the SOAS

The SOAS automatically adjusts itself so that the response to reference signals is approximately described by the closed-loop transfer function

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

where the gain k is such that the amplitude margin is 2. □

This result explains the adaptive properties of the SOAS. The result can also be stated in the following way: The relay acts as a variable gain. The magnitude of the gain depends on the amplitude of the sinusoidal signal at the relay input. This gain is automatically set by the limit cycle oscillation to such a value that the loop gain becomes 0.5 at the frequency of the limit cycle.

The result is illustrated by an example.

EXAMPLE 10.2 A basic SOAS

Assume that the linear parts are characterized by the transfer function

$$G(s) = \frac{K\alpha}{s(s+1)(s+\alpha)}$$

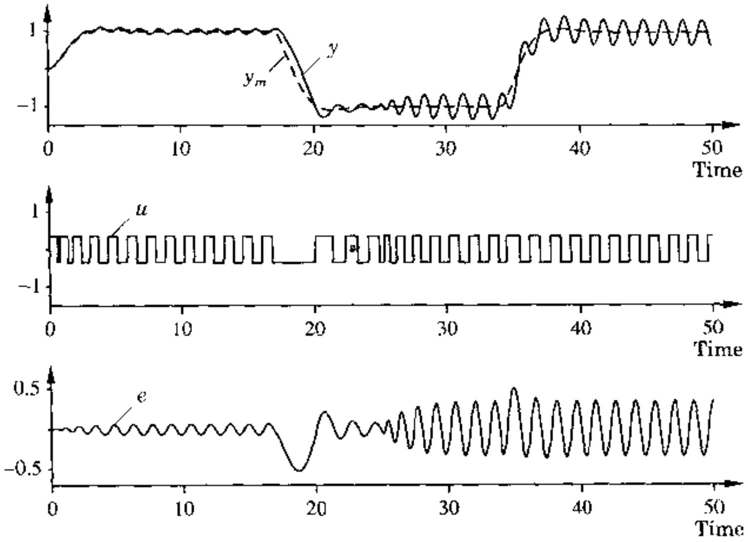

Figure 10.9 Simulation of an SOAS applied to the system in Example 10.2. The dashed line shows the desired response y_m.

From Example 8.1 the period of the limit cycle is approximately given by

$$\omega_u = \sqrt{\alpha}$$

The magnitude of the transfer function at this frequency is

$$|G(i\omega_u)| = \frac{K}{\alpha + 1}$$

If the relay amplitude is d, it follows that the amplitude of the limit cycle oscillation at the relay input is approximately given by

$$e_0 = \frac{Kd}{1 + \alpha}$$

The limit cycle amplitude is thus inversely proportional to α. A simulation of the system is shown in Fig. 10.9. The feedforward transfer function is a second-order system with the damping 0.7 and the natural frequency 1 rad/s. The nominal values of the parameters are $K = 3$, $d = 0.35$, and $\alpha = 20$. The approximate analysis gives a limit cycle with period $T = 1.4$ and amplitude 0.05. The process gain is suddenly increased by a factor of 5 at $t = 25$. Notice the rapid adaptation. However, the amplitude of the oscillation will also increase by a factor of 5. If the value of d is chosen such that the error would be 0.05 for the higher value of K, then the system becomes too slow for small K. □

Design of an SOAS

The self-oscillating adaptive system is a simple nonlinear feedback system that is capable of adapting rapidly to gain variations. The system has a continuous limit cycle oscillation. This is not suitable when valves or other mechanical parts are used as actuators. However, an SOAS may conveniently be used with thyristors as actuators. The presence of the limit cycle oscillation may also cause other inconveniences. Since the system will automatically adjust to an amplitude margin $A_m = 2$, it is also necessary that the characteristics of the process be such that this design principle gives suitable closed-loop properties. The key problem in the design of the SOAS is the compromise between the limit cycle amplitude and the response speed. This compromise is influenced by the selection of the linear compensator, $G_f(s)$, and of the relay amplitude. (Compare Fig. 10.7.) The design for an SOAS can be described by the following procedure.

Step 1: The relay amplitude is first determined such that the desired control authority (tracking rate, force, speed, etc.) is obtained. This can be estimated by analyzing the response of the process to constant control signals.

Step 2: When the relay amplitude is specified, the desired limit cycle frequency can be determined from the condition

$$d\,|G_p(i\omega_u)| = e_0$$

where e_0 is the tolerable limit cycle amplitude in the error signal and $G_p(s)$ is the transfer function of the process. It is necessary to check that the frequency obtained is reasonable. For example, the frequency ω_u may become so high that the process dynamics become uncertain.

Step 3: The final step is to determine the transfer function G_f of the linear compensator such that

$$\arg G_f(i\omega_u) + \arg G_p(i\omega_u) = -\pi$$

A large phase lead may be necessary, but this may not be realizable because of noise sensitivity.

Step 4: Check that the linear closed-loop system with the loop gain $G_0 = KG_fG_p$ will work well when the gain K is adjusted so that the amplitude margin is 2. If this is not the case, the compensator G_f must be modified. □

Notice that it is necessary to have an estimate of the magnitude of the process transfer function in Steps 1 and 2. Knowledge of the phase curve of the process transfer function is necessary in the third step. Also notice that it may not be possible to resolve the compromises in all steps. It is then necessary to add additional loops for changing the gain.

Gain Changers

External feedback loops, which adjust the relay amplitude, may be used to resolve the compromise between a high tracking rate and a small limit cycle amplitude. The so-called up-logic used in the first SOAS can be described as follows:

$$d = \begin{cases} d_1 & \text{if } |e| > e_l \\ d_2 + (d_1 - d_2)e^{-(t-t_0)/T} & \text{if } |e| < e_l \end{cases}$$

The time t_0 is the last time that $|e| < e_l$. The relay amplitude is thus increased to d_1 when the error exceeds a limit e_l. The relay amplitude then decreases to a lower level d_2 when the error is less than e_l. This gain changer increases the relay amplitude and the response rate when large reference signals are applied.

Another type of gain changer has been used to control the amplitude of the limit cycle. The limit cycle amplitude at the process output is measured by a band-pass filter and a rectifier. The relay amplitude is then adjusted to keep the limit cycle amplitude constant at the process output.

Dither Signals

In some applications it is desirable to avoid the limit cycle. One idea that has been used successfully is to introduce a variable gain after the relay. The gain is adjusted so that the limit cycle vanishes. In the early applications it was difficult to implement multiplications. A trick that was used to implement the multiplication is illustrated in Fig. 10.10. A high-frequency triangular wave is added to the signal before the relay. With low-pass filtering, the average effect

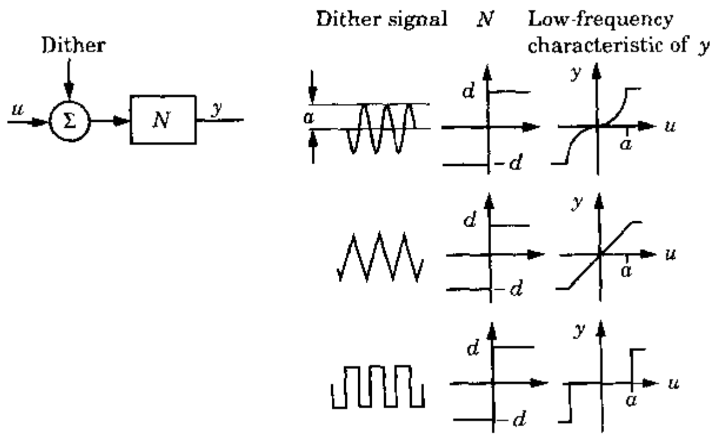

Figure 10.10 The principle of using a dither signal.

of the additive triangular signal is the same as multiplication by a constant. The constant is inversely proportional to the amplitude of the triangular wave. The triangular wave is called a *dither signal*. Use of a dither signal is an illustration of the idea that an oscillation may be quenched by another high-frequency oscillation.

EXAMPLE 10.3 SOAS with lead network and gain changer

The relay control in Example 10.2 gave an error amplitude of about $e_0 = 0.03$. Assume that we want to decrease the amplitude by a factor of 3 while maintaining $d = 0.35$. This gives a new oscillation frequency ω_u' such that

$$d\,|G(i\omega_u')| = 0.01$$

or $\omega_u' = 10$ rad/s. To get this oscillating frequency, a lead network G_f is added such that

$$\arg G_f(i\omega_u') + \arg G_p(i\omega_u') = -\pi$$

Figure 10.11 shows a simulation of the system in Example 10.2 with the compensation network

$$G_f(s) = 1.2\,\frac{s+5}{s+15}$$

As in Fig. 10.9, the gain is increased by a factor of 5 at $t = 25$. It is seen that the lead network decreases the amplitude of the oscillation while maintaining

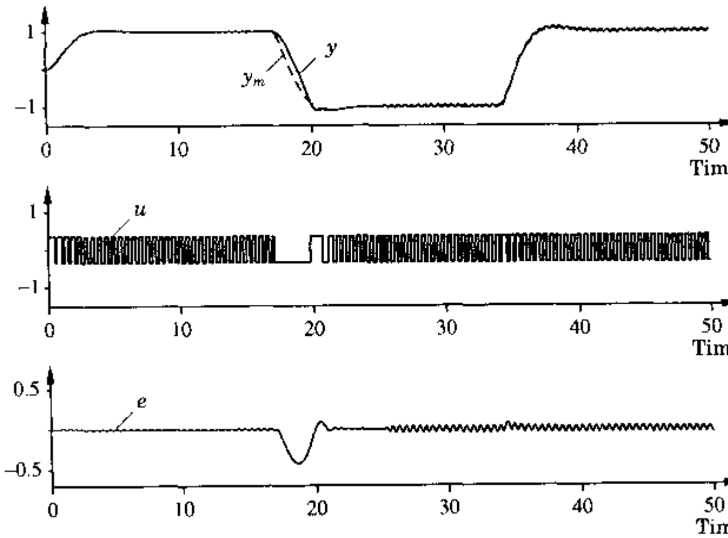

Figure 10.11 Simulation of the system in Example 10.3 using an SOAS with a lead network. The dashed line shows the desired response y_m.

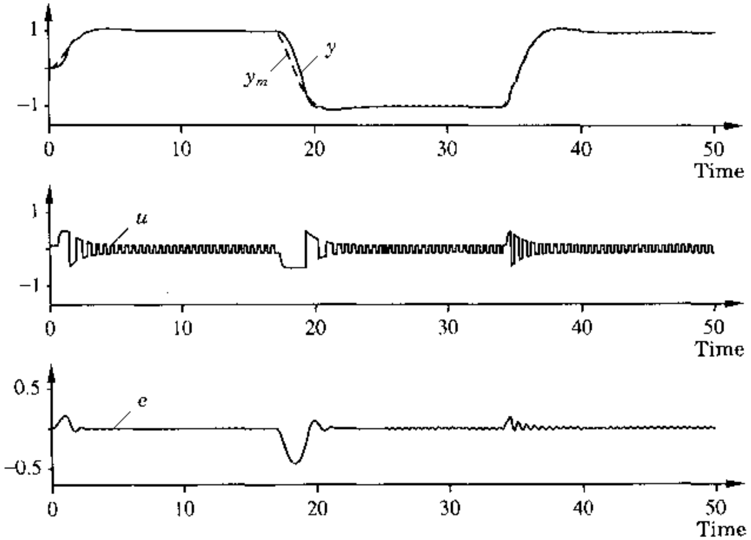

Figure 10.12 Simulation of the system in Example 10.3 using an SOAS with a lead network and a gain changer. The dashed line shows the desired response y_m.

the response speed. To speed up the response, we can introduce the up-logic for the gain. Figure 10.12 shows a simulation in which $d_1 = 0.5$, $d_2 = 0.1$, and $e_l = 0.1$. The error signal is decreased, but there is still an oscillation. The behavior of the closed-loop system can be sensitive to the choice of the parameters in the gain changer. Too large a value of d_1 will cause the error to be larger than e_0, and there will be no decrease in d nor in the amplitude of the error. The oscillation can be quenched by adding a dither signal at the input of the process. □

The examples show how the properties of the SOAS can be changed by using lead filters, gain changers, and dither signals.

Externally Excited Adaptive Systems

A system that is closely related to the SOAS is obtained by injecting a high-frequency sinusoid to measure the gain of the process and to set the controller gain. Such a system is called an *externally excited adaptive system* (EEAS) and gives the designer more freedom than the SOAS because the frequency of the excitation can be chosen more easily. This system is used for track-keeping in compact disc players. The main source for the parameter variation is a gain variation in the laser diode system.

Summary

The basic SOAS is simple to implement and can cope with large gain changes in the process. Result 10.1 shows that the SOAS will automatically adjust itself so that the amplitude margin is 2. However, the limit cycle in the SOAS is noticeable and can be disturbing. The introduction of lead network, gain changer, and dither can decrease the amplitude of the oscillation. The EEAS is a similar system in which a high-frequency signal is introduced externally.

10.4 VARIABLE-STRUCTURE SYSTEMS

In Section 10.2 we showed how fixed robust controllers can be obtained by increasing the complexity of the controller. Another way to obtain a robust controller is to use a special version of on-off control called a *variable-structure system (VSS)*. The key idea is to apply strong control action when the system deviates from the desired behavior. The name "variable-structure" alludes to the fact that the controller structure may be changed.

Sliding Modes

One way to change the structure of the system is to use different controllers in different parts of the state space of the system. Consider the case in which the control law switches on the surface

$$\sigma(x) = 0$$

Assume that the closed-loop system is described by

$$\frac{dx}{dt} = \begin{cases} f^+(x) & \sigma(x) > 0 \\ f^-(x) & \sigma(x) < 0 \end{cases} \tag{10.5}$$

Two situations may occur, which for the two-dimensional case are shown in Fig. 10.13. In Case (a) the trajectories will pass the switching curve and continue into the other region. However, the dynamics are different in the two regions. In Case (b) the vector fields will drive the state toward the surface $\sigma(x) = 0$. The control will change rapidly from one value to another on the switching surface. This is called *chattering*. The net effect is that the state will move toward the surface $\sigma(x) = 0$ and then slide along the surface. This is called *sliding mode*. This sliding motion can be described as follows: Let f_n denote the projection of f on the normal of the surface $\sigma(x) = 0$. Introduce a number α such that

$$\alpha f_n^+ + (1-\alpha) f_n^- = 0$$

The sliding motion is then given by

$$\frac{dx}{dt} = \alpha f^+ + (1-\alpha) f^-$$

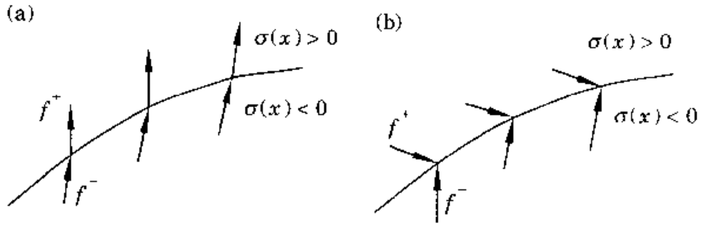

Figure 10.13 The trajectories of Eqs. (10.5) at the switching surface. (a) No sliding mode; (b) sliding mode.

This was formally shown by A. F. Filippov in the early 1960s. The control law giving the sliding motion is sometimes called the equivalent control law. If the switching is not ideal, then the trajectory will move back and forth over the switching surface. This is the case, for instance, when there is hysteresis in the switching. However, on average the motion will be along the switching surface.

Stability and Robustness

In a variable-structure system we attempt to find a switching surface such that the closed-loop system behaves as desired. We now construct a variable-structure controller. For this purpose we assume that the system that we want to control is described by the nonlinear equation

$$\frac{d^n y}{dt^n} = f_1\left(y, \frac{dy}{dt}, \ldots, \frac{d^{n-1}y}{dt^{n-1}}\right) + g_1\left(y, \frac{dy}{dt}, \ldots, \frac{d^{n-1}y}{dt^{n-1}}\right) u$$

If we introduce the state

$$x = \begin{pmatrix} \dfrac{d^{n-1}y}{dt^{n-1}} & \dfrac{d^{n-2}y}{dt^{n-2}} & \ldots & \dfrac{dy}{dt} & y \end{pmatrix}^T \tag{10.6}$$

the system can be written as

$$\begin{aligned}\frac{dx}{dt} &= \begin{pmatrix} 0 & 0 & \ldots & 0 & 0 \\ 1 & 0 & \ldots & 0 & 0 \\ \vdots & & & \vdots & \\ 0 & 0 & \ldots & 1 & 0 \end{pmatrix} x + \begin{pmatrix} f_1(x) + g_1(x)u \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\ &= f(x) + g(x)u \\ y &= \begin{pmatrix} 0 & 0 & \ldots & 0 & 1 \end{pmatrix} x \end{aligned} \tag{10.7}$$

where $f(x)$ and $g(x)$ are vectors. The system is nonlinear but is affine in the control signal. Further, it is assumed that all the states can be measured and

that the state vector has the special form given in Eq. (10.6). For simplicity it is assumed that the purpose of the control is to find a controller such that $x = 0$ is an asymptotically stable solution. The problem with constant reference signals is considered in Problem 10.14 at the end of the chapter.

There are three important questions that must be answered for VSS:

- Will the trajectories starting at any point hit the switching line?
- Is there a sliding mode?
- Is the sliding mode stable?

There are partial answers to these questions in the literature on VSS. For the special type of system defined by Eqs. (10.7) it is easy to derive a controller that makes the sliding mode stable.

Let the switching surface be

$$\sigma(x) = p_1 x_1 + p_2 x_2 + \cdots + p_n x_n = p^T x = 0 \tag{10.8}$$

Using the definition of the state vector, we find that

$$\sigma(x) = p_1 y^{(n-1)} + p_2 y^{(n-2)} + \cdots + p_n y = 0$$

The dynamic behavior on the sliding surface can be specified by a proper choice of the numbers p_i. The motion is determined by a differential equation of order $n - 1$. It will be stable if the polynomial

$$P(s) = p_1 s^{n-1} + p_2 s^{n-2} + \cdots + p_n \tag{10.9}$$

has all its roots in the left-half plane.

To determine a control law that keeps the system on $\sigma(x) = 0$, we introduce the Lyapunov function $V(x) = \sigma^2(x)/2$. The time derivative of V is given by

$$\begin{aligned}\frac{dV}{dt} &= \sigma(x)\dot{\sigma}(x) = x^T p p^T \dot{x} \\ &= x^T p \left(p^T f(x) + p^T g(x) u(t)\right)\end{aligned}$$

Choose the control law

$$u(t) = -\frac{p^T f}{p^T g} - \frac{\mu}{p^T g}\,\mathrm{sign}(\sigma(x)) \tag{10.10}$$

Then

$$\frac{dV}{dt} = -\mu\sigma(x)\,\mathrm{sign}(\sigma(x)) \tag{10.11}$$

which is negative definite. This implies that $\sigma(x) = 0$ is asymptotically stable. Notice that there is a discontinuity in the control signal when the switching surface is passed.

Assume that the system has initial values such that $\sigma(x) = \sigma_0 > 0$, and let t_σ be the time when the switching surface is reached the first time. From Eq. (10.11) we find that

$$\dot{\sigma}(x) = -\mu$$

Integrating this equation from 0 to t_σ gives

$$0 - \sigma_0 = -\mu(t_\sigma - 0)$$

which gives $t_\sigma = \sigma_0/\mu$. Using the same arguments for $\sigma_0 < 0$ shows that $t_\sigma = |\sigma_0|/\mu$. With the control law given by Eq. (10.10) the state will thus reach the switching surface in finite time. The subspace $\sigma(x) = 0$ is asymptotically stable, and the state will stay on the switching surface once it is reached. The motion along the surface is determined by Eq. (10.9).

Uncertainties in f and g can be handled if μ is sufficiently large. Assume that the design of the control law is based on the approximate values $\hat{f}$ and $\hat{g}$ instead of the true ones. Then

$$\frac{dV}{dt} = \sigma\left(\frac{p^T\left(f\hat{g}^T - \hat{f}g^T\right)p}{p^T\hat{g}} - \mu\frac{p^Tg}{p^T\hat{g}}\,\mathrm{sign}(\sigma)\right)$$

The right-hand side is negative if μ is sufficiently large, provided that $p^T\hat{g}$ and p^Tg have the same sign. The system will thus be insensitive to uncertainties in the process model.

One way to design a variable-structure system is to first transform the system to the form given by Eqs. (10.7). This is possible for controllable linear systems without zeros and for some classes of nonlinear systems. A stable switching surface in the transformed variables is then determined. The system and the switching criteria can then be transformed back to the original state variables. Notice, however, that all states must be measured.

Smooth Control Laws

The control law (10.10) has the drawback that the relay chatters. One way to avoid this is to make the relay characteristics smoother. To do this, introduce a boundary layer around the switching surface

$$B(t) = \left\{x(t)\,\middle|\, |\sigma(x(t))| \le \varepsilon\right\} \qquad \varepsilon > 0$$

The parameter ε can be interpreted as a measure of the thickness of the boundary layer. The sign function in Eq. (10.10) is now replaced by the saturation function

$$\mathrm{sat}(\sigma, \varepsilon) = \begin{cases} 1 & \sigma > \varepsilon \\ \sigma/\varepsilon & -\varepsilon \le \sigma \le \varepsilon \\ -1 & \sigma < -\varepsilon \end{cases}$$

The control law is then

$$u(t) = \frac{p^Tf}{p^Tg} - \frac{\mu}{p^Tg}\,\mathrm{sat}(\sigma(x), \varepsilon) \qquad (10.12)$$

The width of the boundary layer will influence the tracking performance and the bandwidth of the closed-loop system.

EXAMPLE 10.4 Second-order VSS

Consider the unstable system

$$\frac{dx}{dt} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u = Ax + Bu$$

$$y = \begin{pmatrix} 0 & 1 \end{pmatrix} x$$

which has the transfer function

$$G(s) = \frac{1}{s(s-1)}$$

To design a variable-structure controller we determine the closed-loop dynamics by choosing the switching line

$$\sigma(x) = p_1x_1 + p_2x_2 = x_1 + x_2$$

Along the sliding line $\sigma = 0$ we have

$$\sigma(x) = x_1 + x_2 = \frac{dy}{dt} + y = 0$$

Since the system is in controllable form, the closed-loop behavior is independent of the system parameters at the sliding mode. The sliding mode controller

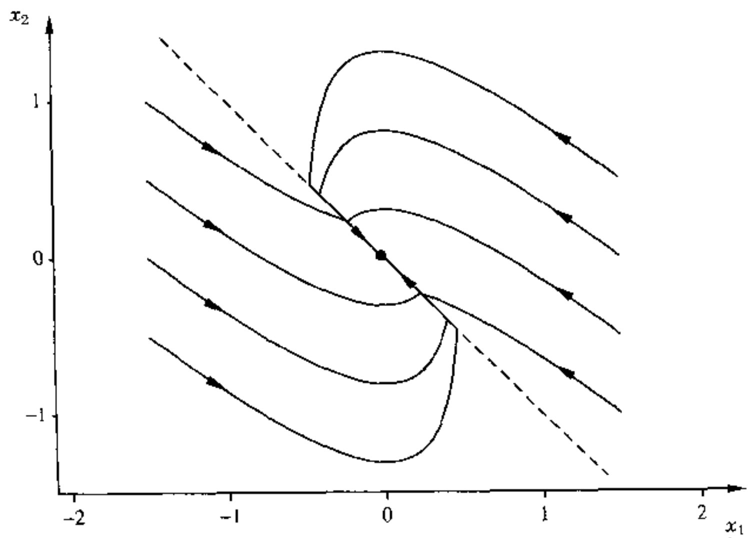

Figure 10.14 Phase portrait of the system in Example 10.4. The dashed line shows $\sigma(x) = 0$.

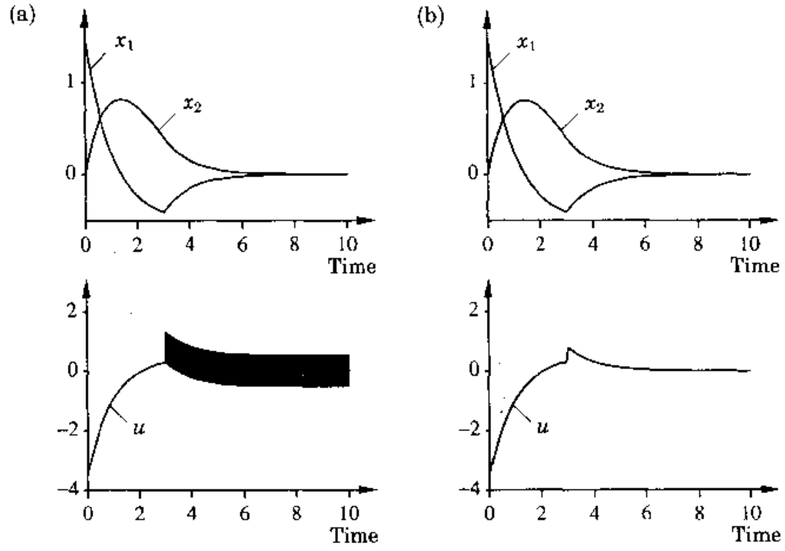

Figure 10.15 The states and the output as a function of time in Example 10.4. The initial conditions are $x_1(0) = 1.5$ and $x_2(0) = 0$. The controllers are (a) Eq. (10.10) with $\mu = 0.5$; (b) Eq. (10.12) with $\mu = 0.5$ and $\varepsilon = 0.01$.

(10.10) is now

$$u(t) = -\frac{p^T Ax}{p^T B} - \mu \text{sign}(\sigma(x))$$
$$= -\begin{pmatrix} 2 & 0 \end{pmatrix} x(t) - \mu \text{sign}(\sigma(x))$$

The phase plane of the system is shown in Fig. 10.14 when $\mu = 0.5$. The input and the output for one initial value are shown in Fig. 10.15(a). The trajectories hit the switching line $\sigma = 0$ and stay on it. This implies that the control signal will chatter. Using the control law (10.12),

$$u(t) = -\begin{pmatrix} 2 & 0 \end{pmatrix} x(t) - \mu \text{sat}(\sigma(x), \varepsilon)$$

with $\varepsilon = 0.01$, gives the behavior shown in Fig. 10.15(b). The control signal is now smooth, but the differences in the state trajectories are negligible. □

Summary

Variable-structure systems are related to the self-oscillating adaptive systems (SOAS). In variable-structure systems we want the system to get into a sliding mode to obtain insensitivity to parameter variations. The control signal of

variable-structure systems will chatter in the sliding mode. The chatter can be avoided by smoothing the relay characteristics. The amplitude of the control signal is determined by the magnitude of the state variables or the error. With this modification the variable-structure system can be regarded as an SOAS in which the relay amplitude depends on the states. The switching condition is a linear function of the error in the SOAS, while in variable-structure systems it is a nonlinear function of the states.

The theory on VSS can be extended to controllers, in which the feedback is done from a reduced number of state variables. However, the conditions will become more complex than those discussed in this section. There will be more constraining conditions on the choice of the switching plane. Since the conditions for the existence of a sliding mode depend on the process and the switching plane, there have been attempts to make adaptive VSS by adaptation on $\sigma(x)$.

One main drawback of variable-structure systems is the problem of choosing the switching plane. It also requires measurement of all state variables. Another drawback is the chatter in the control signal in the sliding mode.

10.5 CONCLUSIONS

Robust high-gain control can be very effective for systems with structured parameter variations, where the range of the variations is known. If the parameter bounds are uncertain, high-gain design methods will lead to a complex and conservative design. Relay feedback is an extreme form of high-gain systems. In this chapter we have described different ways to use relay feedback to obtain systems that are insensitive to parameter variations. Self-oscillating adaptive systems and variable-structure systems are two applications of this idea. The SOAS can be designed to work quite well, but it requires engineering effort and some knowledge of the process to get a satisfactory performance of the closed-loop system. These drawbacks have resulted in lack of interest in the SOAS. However, the ideas behind SOAS have become useful in connection with the auto-tuning of simple controllers, as discussed in Chapter 8.

PROBLEMS

10.1 Determine whether each of the following plants can be stabilized by a linear fixed-parameter compensator when $a \in [-1, 1]$:
(a) a/s; (b) $1/(s+a)$; (c) $1/(1+as)$; (d) $a/(1+s)$;
(e) $a/(1-s)$.

10.2 Consider the process

$$G_p(s) = e^{-sT} \qquad T \in [0, 1]$$

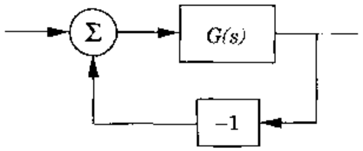

Figure 10.16 The system in Problem 10.3.

(a) Show that the process can be controlled by a controller of the structure in Fig. 10.1 with

$$G_{fb}(s) = \frac{0.6(1 + s/1.3)}{s(1 + s/2)}$$

$$G_{ff}(s) = \frac{1 + s}{1 + s/3}$$

(b) Simulate the behavior for changes in the command signal and step disturbances at the output.

(c) Discuss how to make a self-tuning regulator based on pole placement for the process.

10.3 Consider the linear closed-loop system shown in Fig. 10.16 with the same $G(s)$ as in Example 10.2 and with $\alpha = 20$. Determine the gain K so that the amplitude margin is $A_m = 2$. Simulate the system and determine its step response. Compare this with the step response of the corresponding SOAS in Example 10.2.

10.4 Consider a linear plant with the transfer function

$$G(s) = \frac{k}{s(s+1)^2}$$

where the gain k may vary in the range $0.1 \le k \le 10$. Determine the relay amplitude d and a suitable lead network so that the limit cycle amplitude at the process output is less than 0.05 and the rise time to a step of unit amplitude is never less than 0.5. Simulate the resulting design and verify the results.

10.5 Consider the system in Example 10.2. Experiment with a gain changer of the up-logic type. Investigate how a dither signal will influence the performance of the closed-loop system.

10.6 Consider the system in Problem 10.4. Design a gain changer that keeps the limit cycle amplitude at 0.01 for the whole operating range.

10.7 Consider a system with the transfer function

$$G(s) = \frac{k}{s+1}$$

where the gain k may change in the range 0.1 to 10. Design a servo using the SOAS principle so that the closed-loop transfer function is

$$G(s) = \frac{1}{s^2 + s + 1}$$

independent of the process gain.

10.8 Consider the system in Example 10.4 and assume that the controller (10.10) is used with $\mu = 0.5$. Assume that the process is changed such that

$$\frac{dx_1}{dt} = ax_1 + bu$$

Determine values of a and b such that the closed-loop system is still asymptotically stable.

10.9 Assume that the nth-order single-input, single-output system

$$\frac{dx}{dt} = Ax + Bu$$

is in companion form and that the control law is of the form

$$u = -\sum_{i=1}^{n} l_i x_i - \mu \operatorname{sign}(\sigma(x))$$

Derive the necessary and sufficient conditions for the existence of a sliding mode. When will the sliding mode be stable?

10.10 Consider the process in Problem 1.9. Design a robust controller for the system. Investigate the disturbance rejection of the closed-loop system.

10.11 Consider the process in Problem 1.10. Design a robust controller for the system. Investigate the disturbance rejection of the closed-loop system.

10.12 Design an SOAS for the system in Problem 1.9, and investigate its properties.

10.13 Design an SOAS for the system in Problem 1.10, and investigate its properties.

10.14 Consider the process in Eqs. (10.7). Assume that the reference value is constant y_r. The desired state is then

$$x_d = \begin{pmatrix} 0 & \dots & 0 & y_r \end{pmatrix}$$

Determine a controller such that $x = x_d$ is an asymptotically stable solution. (*Hint:* Introduce the state error $\tilde{x} = x - x_d$, and consider the Lyapunov function $V(\tilde{x}) = \sigma^2(\tilde{x})/2$.)

REFERENCES

The robust high-gain design is closely related to early ideas on feedback amplifiers. See:

Bode, H. W., 1945. *Network Analysis and Feedback Amplifier Design.* New York: Van Nostrand.

The basic robust design method for SISO systems for specifications in the frequency domain is discussed in:

Horowitz, I. M., and M. Sidi, 1972. "Synthesis of feedback systems with large plant ignorance for prescribed time-domain tolerances." *Int. J. Control* **16**: 287–309.

Horowitz, I. M., 1973. "Optimum loop transfer function in single-loop minimum-phase feedback systems." *Int. J. Control* **18**: 97–113.

Horowitz, I. M., and U. Shaked, 1975. "Superiority of transfer function over state-variable methods in linear time-invariant feedback system design." *IEEE Trans. Automat. Contr.* **AC-20**: 84–97.

Horowitz, I. M., and M. Sidi, 1978. "Optimum synthesis for non-minimum phase feedback systems with plant uncertainty." *Int. J. Control* **27**: 361–386.

This last paper presents criteria for determining whether a given set of performance specifications are achievable, and, if so, a synthesis procedure is included for deriving the optimum design, which is defined as that with an effectively minimum-loop transmission bandwidth. The theory on QFT is summarized in:

Horowitz, I. M., 1993. *Quantitative Feedback Design Theory (QFT),* Vol 1. Boulder, Colo.: QFT Publications.

Robust design of processes with unstructured uncertainties is treated in:

Doyle, J. C., and G. Stein, 1981. "Multivariable feedback design: Concepts for a classical/modern synthesis." *IEEE Trans. Automat. Contr.* **AC-26**: 4–16.

Morari, M., and J. C. Doyle, 1986. "A unifying framework for control system design under uncertainty and its implications for chemical process control." In *Chemical Process Control: CPCIII,* eds. M. Morari and T. McAvoy, Proceedings of the 3rd International Conference on Chemical Process Control. New York: Elsevier.

Research on relay systems was very active in the 1950s and 1960s. An authoritative treatment by one of the key contributors is:

Tsypkin, Y. Z., 1984. *Relay Control Systems.* Cambridge, U.K.: Cambridge University Press.

The method of harmonic balance and describing function is extensively treated in:

Gelb, A., and W. E. Vander Velde, 1968. *Multiple-Input Describing Functions and Nonlinear System Design,* pp. 18, 273, 308, 317, and 588. New York: McGraw-Hill.

This book also contains applications of SOAS. The background of SOAS and more about the design rules can be found in:

Lozier, J. C., 1950. "Carrier-controlled relay servos." *Elec. Eng.* **69**: 1052–1056.

Schuck, O. H., 1959. "Honeywell's history and philosophy in the adaptive control field." In *Proceedings of the Self Adaptive Flight Control Symposium,* ed. P. C. Gregory. Wright Patterson AFB, Ohio: Wright Air Development Center.

Horowitz, I. M., 1964. "Comparison of linear feedback systems with self-oscillating adaptive systems." *IEEE Trans. Automat. Contr.* **AC-9**: 386–392.

Horowitz, I. M., J. W. Smay, and A. Shapiro, 1974. "A Synthesis Theory for Self-Oscillating Adaptive Systems (SOAS)." *Automatica* **10**: 381–392.

A modified version of the SOAS was flight-tested extensively in the experimental rocket plane X-15. Experiences from that are summarized in:

Thompson, M. O., and J. R. Welsh, 1970. "Flight test experience with adaptive control systems." *Proceedings of the Agard Conference on Advanced Control Systems Concepts,* vol. 58, pp. 141–147. Agard, Neuilly-sur-Seine, France.

The general conditions for stability of the limit cycle in relay systems are still unknown. Some guidance is given by the stability conditions in:

Åström, K. J., and T. Hägglund, 1984. "Automatic tuning of simple regulators." *Proceedings of the IFAC 9th World Congress.* Vol. 3, pp. 267–272. Budapest.

Additional results on relay oscillations are found in:

Amsle, B. E., and R. E. Gorozdos, 1959. "On the analysis of bi-stable control systems." *IEEE Trans. Automat. Contr.* **AC-4**: 46–58.

Gille, J. C., M. J. Pelegrin, and P. Decaulne, 1959. *Feedback Control Systems.* New York: McGraw-Hill.

Atherton, D. P., 1975. *Nonlinear Control Engineering: Describing Function Analysis and Design.* London: Van Nostrand Reinhold.

Atherton, D. P., 1982. "Limit cycles in relay systems." *Electronics Letters* **1**(21).

A procedure for designing externally excited adaptive systems (EEAS) is given in:

Horowitz, I. M., J. W. Smay, and A. Shapiro, 1957. "A synthesis theory for the externally excited adaptive system (EEAS)." *IEEE Trans. Automat. Contr.* **AC-2**: 101–107.

Variable-structure systems are treated in:

Emelyanov, S. V., 1967. *Variable Structure Control Systems.* Munich: Oldenburger Verlag.

Itkis, U., 1976. *Control Systems of Variable Structure.* New York: Halsted Press, Wiley.

Utkin, V. I., 1977. "Variable structure systems with sliding modes." *IEEE Trans. Automat. Contr.* **AC-22**: 212–222.

Slotine, J.-J. E., and W. Li, 1991. *Applied Nonlinear Control.* Englewood Cliffs, N.J.: Prentice-Hall.

The sufficient conditions of the existence of a sliding mode is found in:

Gough, N. E., Z. M. Ismail, and R. E. King, 1984. "Analysis of variable structure systems with sliding modes." *Int. J. Systems Sci.* **15**(4): 401–409.

Adaptive variable-structure systems are discussed in:

Young, K.-K. D., 1978. "Design of variable structure model-following control systems." *IEEE Trans. Automat. Contr.* **AC-23**: 1079–1085.

Zinober, A. S. I., 1981. "Adaptive variable structure systems." In *Proceedings of the Third IMA Conference on Control Theory*, eds. J. E. Marshall, W. D. Collins, C. J. Harris, and D. H. Owens. London: Academic Press.

An overview and a presentation of several industrial applications of variable-structure systems are given in the survey paper:

Utkin, V. I., 1987. "Discontinuous control systems: State of the art in theory and applications." *Preprints of the IFAC 10th World Congress.* Vol. 1, pp. 75–94. Munich.

CHAPTER 11

PRACTICAL ISSUES AND IMPLEMENTATION

11.1 INTRODUCTION

The previous chapters were devoted mainly to development of algorithms and analysis of adaptive systems. In this chapter we discuss practical implementation of adaptive controllers. The presentation will be guided by theoretical considerations, but since the issues are quite complicated, theory can cover only part of the problems. Several issues will therefore be solved in an ad hoc manner and verified by extensive experimentation and simulation.

Since an ordinary digital controller is an integral part of an adaptive controller, it is essential to master implementation of conventional controllers. Some aspects of this are covered in Section 11.2. The discussion includes computational delay, sampling, prefiltering and postfiltering, and integrator windup. Automatic design of a controller is another important part of an adaptive controller. In this book we have mostly used simple design methods based on pole placement. In Section 11.3 we discuss how this design method can be modified in several ways to accommodate more complex specifications and to make it more robust. The design technique requires the solution of a Diophantine equation. Efficient numerical methods for doing this are discussed in Section 11.4.

Parameter estimation is another important part of an adaptive controller. Implementation of estimators is discussed in Section 11.5. This includes choice of model structure, data filters, and excitation. The ability of an adaptive controller to track time-varying parameters is an important issue. There are several ways to do this. Two techniques, exponential forgetting and covariance resetting, are discussed in detail. It turns out that exponential forgetting in combination with poor excitation can give rise to an undesirable effect called covariance windup. This phenomenon is discussed in detail together with several ways of avoiding it. The techniques discussed include constant trace algorithms, directional forgetting, and leakage. A technique to make the estimator less sensitive to outliers is also discussed in Section 11.5. It is important to have numerically efficient methods for recursive estimation. Square root algorithms, which are numerically superior to the conventional algorithms, are discussed in Section 11.6.

In Section 11.7 we discuss the interaction between estimation and control. We show that difficulties can arise if integral action is implemented inappropriately. We also show how the criteria for control and estimation can be made compatible by appropriate choices of the data filter and the experimental conditions.

Some prototype algorithms are given in Section 11.8. To implement a control system successfully, it is necessary to consider all situations that may occur in practice. Conventional controllers typically have two operating modes: manual and automatic. Adaptive controllers have many more modes. This is discussed briefly in Section 11.9, which covers startup, shut-down, and switching between different modes. Supervision of adaptive algorithms is also discussed in that section.

11.2 CONTROLLER IMPLEMENTATION

An ordinary controller is an important part of an adaptive controller. Compare with the block diagram in Fig. 1.1. It is, of course, important that the controller be implemented in a good way. Implementation of digital controllers is treated in books on digital control. Some aspects are summarized in this section.

Computational Delay

Because the analog-to-digital (A-D) and digital-to-analog (D-A) conversions and the computations take time, there will always be a delay between the measurement and the time the control signal is applied to the process. This delay, which is called the *computational delay*, depends on how the control law is implemented in the computer. Two ways are illustrated in Fig. 11.1. In Case (a) the measured variable at time t_k is used to compute the control signal applied at time t_{k+1}. In Case (b) the control signal is applied as soon as the computations are finished. The disadvantage of Case (a) is that the control action is delayed unnecessarily. In Case (b) the disadvantage is that the time delay may change, depending on the load on the computer or changes in the program. In both cases it can be necessary to include the computational delay in the design of the controller.

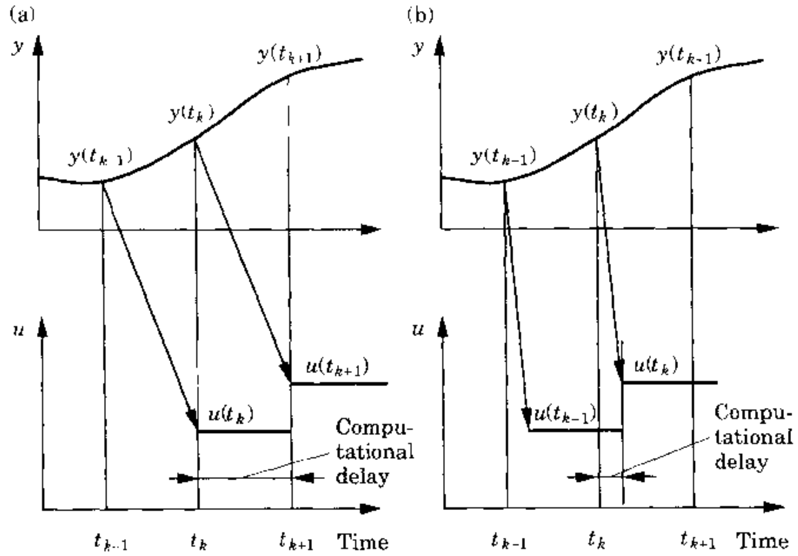

Figure 11.1 Two ways to synchronize inputs and outputs. (a) The signals measured at time t_k are used to compute the control signal applied at t_{k+1}. (b) The control signal is applied as soon as it is computed.

In Case (b) it is desirable to make the delay as small as possible. This can be done by performing as few operations as possible between the A-D and D-A conversions. Assume that the regulator has the form

$$\begin{aligned} u(t) + r_1 u(t-1) + \ldots + r_k u(t-k) \\ = t_0 u_c(t) + \ldots + t_m u_c(t-m) - s_0 y(t) - \ldots - s_l y(t-l) \end{aligned}$$

This equation can be written as

$$u(t) = t_0 u_c(t) - s_0 y(t) + u'(t-1)$$

where

$$\begin{aligned} u'(t-1) = t_1 u_c(t-1) + \ldots + t_m u_c(t-m) - s_1 y(t-1) - \ldots - s_l y(t-l) \\ - r_1 u(t-1) - \ldots - r_k u(t-k) \end{aligned}$$

Notice that the signal $u'(t-1)$ contains information that is available at time $t-1$. An implementation of the control algorithm that exploits this to make the computational delay as small as possible is the following:

1. Perform A-D conversion of $y(t)$ and $u_c(t)$.
2. Compute $u(t) = t_0 u_c(t) - s_0 y(t) + u'(t-1)$.

3. Perform D-A conversion of $u(t)$.
4. Compute

$$u'(t) = t_1 u_c(t) + \ldots + t_m u_c(t-m+1) - s_1 y(t) - \ldots - s_l y(t-l+1)$$
$$- r_1 u(t) - \ldots - r_k u(t-k+1)$$

The computational delay can thus be significantly reduced by a proper implementation of the controller. In a proper implementation the delay can be reduced. Apart from the two multiplications and the two additions, $u(t)$ must be tested for limitations and anti-reset windup must be done in a proper way. Since the computational delay appears in the same way as a time delay in the process dynamics, it is important to take it into account in designing a control system. A common rule of thumb is that the time delay can be neglected if it is less than 10% of the sampling period. For high-performance systems it should always be taken into account. Since the time delay is not known until the algorithm has been coded, the control design may have to be repeated. For an adaptive system it is important that the model structure is chosen so that the computational delay can be accommodated. In multitasking systems it may also happen that the computational delay varies with time.

Sampling and Pre- and Postfiltering

The choice of sampling rate is an important issue in digital control. The sampling rate influences many properties of a system such as following of command signals, rejection of load disturbances and measurement noise, and sensitivity to unmodeled dynamics. Selection of sampling rates is thus an essential design issue.

One rule of thumb that is useful for deterministic design methods is to let the sampling interval h be chosen such that

$$\omega_o h \approx 0.2 - 0.6$$

where ω_o is the natural frequency of the dominating poles of the closed-loop system. This corresponds to 12–60 samples per undamped natural period. The sampling frequency is $\omega_s = 2\pi/h$.

In all digital systems it is important that signals are filtered before they are sampled. All components of the signal with frequencies above the Nyquist frequency, $\omega_N = \omega_s/2 = \pi/h$ should be eliminated. If this is not done, a signal component with frequencies $\omega > \omega_N$ will appear as low-frequency components with the frequency

$$\omega_a = |((\omega + \omega_N) \bmod \omega_s) - \omega_N|$$

This phenomenon is called *aliasing*, and the prefilters introduced before a sampler are called anti-aliasing filters. Suitable choices of anti-aliasing filters

Table 11.1 Damping and natural frequency of second-, fourth-, and sixth-order Butterworth, ITAE, and Bessel filters. The filters have the bandwidth ω_B.

Order	Butterworth		ITAE		Bessel	
	ω/ω_B	ζ	ω/ω_B	ζ	ω/ω_B	ζ
2	1	0.71	1	0.71	1.27	0.87
4	1	0.38	1.48	0.32	1.59	0.62
	1	0.92	0.83	0.83	1.42	0.96
6	1	0.26	1.30	0.32	5.14	0.49
	1	0.71	0.98	0.60	4.57	0.82
	1	0.97	0.79	0.93	4.34	0.98

are second- or fourth-order Butterworth, ITAE (integral time absolute error), or Bessel filters. They consist of one or several cascaded filters of the form

$$G_f(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

Let ω_B be the desired bandwidth of the filter. The damping ζ and the frequency ω for filters of different orders are given in Table 11.1. The Bessel filter has the interesting property that its phase curve is approximately linear, which implies that the waveform is also approximately invariant.

The prefilter introduces additional dynamics into the system that have to be taken into account in the control design. The Bessel filter can be approximated with a time delay. Assume that the bandwidth of the filter is chosen to be

$$|G_{aa}(i\omega_N)| = \beta$$

where $G_{aa}(s)$ is the transfer function of the filter and $\omega_N = \pi/h$ is the Nyquist frequency. Parameter β is the attenuation of the filter at the Nyquist frequency. Table 11.2 gives the approximate time delay T_d as a function of β. The table

Table 11.2 The approximate time delay T_d due to the anti-aliasing filter as a function of the desired attenuation β at the Nyquist frequency for a fourth-order Bessel filter. h is the sampling period.

β	ω_N/ω_B	T_d/h
0.05	3.1	2.1
0.1	2.5	1.7
0.2	2.0	1.3
0.5	1.4	0.9
0.7	1.0	0.7

also gives ω_N as a function of filter bandwidth ω_B. The relative delay increases with attenuation. For reasonable values of the attenuation the delay is more than one sampling period. This means that the dynamics of the filter must be taken into account in the control design. We illustrate this by an example.

EXAMPLE 11.1 The effect of the anti-aliasing filter

Consider a process described by

$$G(s) = \frac{1}{s(s+1)}$$

A pole placement controller is designed to give a closed-loop system whose dominant poles are given by $\omega_m = 1$ rad/s and $\zeta_m = 0.7$. The digital controller has a sampling period of $h = 0.5$. To illustrate the effect of aliasing, we assume that the output of the system is disturbed by a sinusoidal signal; that is, the measured signal is

$$y_m(t) = y(t) + a_d \sin(\omega_d t)$$

with $a_d = 0.1$. This signal is filtered through a fourth-order Bessel filter with bandwidth ω_B. Figure 11.2 shows a number of simulations of the system that

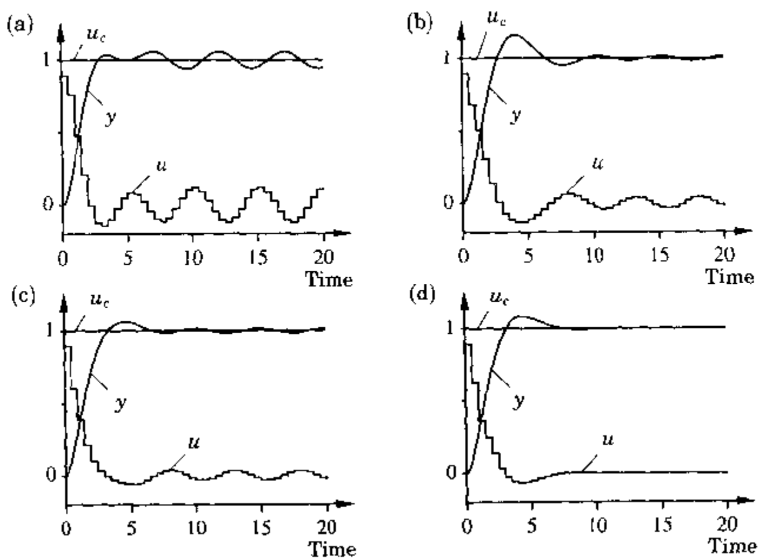

Figure 11.2 Output, reference value, and control signal for the system in Example 11.1. The measurement disturbance has the frequency $\omega_d = 11.3$ rad/s. (a) $\omega_B = 25$ rad/s; (b) $\omega_B = 6.28$ rad/s; (c) $\omega_B = 6.28$ rad/s and the regulator compensated for a delay of $0.7h$; (d) $\omega_B = 2.51$ rad/s and the regulator compensated for a delay of $1.7h$.

illustrate the effects of aliasing and filtering. Figure 11.2(a) shows setpoint, process output, and control signal. Since the bandwidth of the filter is $\omega_B = 25$, the measured signal is not attenuated much by the filter. The disturbance with frequency $\omega_d = 11.3$ is aliased to 1.3 rad/s because of the sampling with Nyquist frequency $\omega_N = 2\pi$. The aliased disturbance is clearly visible in the process input and output. In Fig. 11.2(b) the bandwidth of the prefilter is reduced to $\omega_B = 6.28$ rad/s. This bandwidth is not sufficiently small to give a substantial reduction of the disturbance. Notice also that the overshoot has increased a little because of the delay in the prefilter. In Fig. 11.2(c) the dynamics of the prefilter have been taken into account by adding a time delay of $0.7h$ in the model. The overshoot is then reduced, but the effect of the disturbance is similar to Fig. 11.2(b). In Fig. 11.2(d) the filter bandwidth has been reduced to $\omega_B = 2.51$. The disturbance is now reduced significantly. We have also taken the dynamics of the filter into account as a time delay of $1.7h$ in designing the controller. The aliased disturbance is now barely noticeable in the figure. □

Example 11.1 shows that it is important to use an anti-aliasing filter and that the filter has to be considered in the design. For a Bessel filter, however, it is sufficient to approximate the filter by a time delay. The additional dynamics cause no principle problems for an adaptive controller because all parameters are estimated. However, inclusion of the prefilter dynamics will increase the model order significantly. In the particular case of Example 11.1 the model will increase from second order to sixth order. This means that the number of parameters that we have to estimate increases from 4 to 12. A simple way to reduce the number of parameters is to approximate the prefilter by a delay. It is then sufficient to estimate five parameters of the model

$$\begin{aligned} y(t) + a_1 y(t-h) + a_2 y(t-2h) &= \\ b_0 u(t-dh) + b_1 u(t-dh-h) &+ b_2 u(t-dh-2h) \end{aligned}$$

where the value of d depends on the bandwidth of the filter (see Table 11.2).

It is cumbersome and costly to change the bandwidth of an analog prefilter. This poses problems for systems in which the sampling rate has to be changed. A nice implementation in such a case is to use dual rate sampling. A high fixed sampling rate is used together with a fixed analog prefilter. A digital filter is then used to filter the signal at a slower rate when that is needed. This implies that fewer parameters have to be estimated.

The output of a D-A converter is a piecewise constant signal. This means that the control signal fed to the actuator is a piecewise constant signal that changes stepwise at the sampling instants. This is adequate for many processes. However, for some systems, such as hydraulic servos for flight control and other systems with poorly damped oscillatory modes, the steps may excite these modes. In such a case it is advantageous to use a filter that smooths the signal from the D-A converter. Such a filter is called a *postsampling filter*. The postsampling filter may be a simple continuous-time filter with a response time that is short in comparison with the sampling time. Special D-A converters that give a smooth signal have also been constructed. Another solution to the problem is to use a system with dual rate sampling. The primary control system should then be designed so that the output is piecewise linear between the sampling instants. A fast sampling can then be used to generate an approximation to this signal, possibly followed by an analog postsampling filter.

Controller Windup

Linear theory is adequate to deal with many problems in control system design. There is, however, one nonlinear problem that we must deal with in almost all practical control systems, and that is actuator saturation. The feedback will be broken when the actuator saturates. Large deviation may then occur if the process or the controller is unstable. A simple case in which this occurs is when the controller has integral action. The phenomenon was first observed in connection with PID control. It is therefore often called *integrator windup* because the integral term "winds up" when the actuator is saturated. Since integral action was also called reset, the phenomenon is also called *reset windup*. It is necessary to include a scheme for avoiding windup in systems in which the process and/or the controller is unstable.

There are many different ways to introduce anti-reset windup. One simple way is based on the interpretation of a controller as a combination of a state estimator and state feedback. Such a system is shown in Fig. 11.3. The controller is composed of two components, a state estimator and a state feedback. The state estimator determines an estimate of the state based on the process input and output. The state feedback generates the control signal based on the estimated state. It is intuitively clear from the figure that the state estimator will perform poorly when the actuator saturates because it is uses a wrong

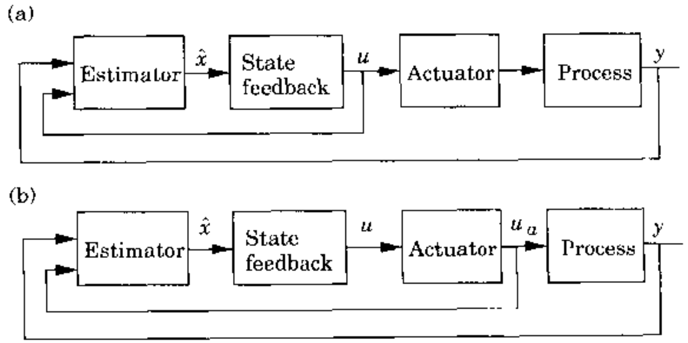

Figure 11.3 Block diagrams of controllers based on state feedback and state estimation with a process having a nonlinear actuator.

value of the control signal applied to the process. This interpretation also suggest that the problem can be avoided by feeding back the actual process input, u_a, or an estimate of it as in Fig. 11.3(b).

Since we are using polynomial representations of the controller in this book, we also give a polynomial interpretation of the scheme. Consider the controller

$$R(q)u(t) = T(q)u_c(t) - S(q)y(t)$$

where the polynomial $R(q)$ is assumed to be monic. The controller can be written in observer form as

$$A_o(q)u(t) = T(q)u_c(t) - S(q)y(t) + (A_o(q) - R(q))u(t)$$

where $A_o(q)$ is the observer polynomial. Let the saturating actuator be described by the nonlinear function $f(u)$. A controller that avoids windup is then given by

$$\begin{aligned} A_o(q)v(t) &= T(q)u_c(t) - S(q)y(t) + (A_o(q) - R(q))u(t) \\ u(t) &= f(v(t)) \end{aligned} \tag{11.1}$$

A similar scheme can be used when the saturation is dynamic. Notice that the controller responds with the observer dynamics when the feedback is broken. A particularly simple case is when $A_o^* = 1$, which corresponds to a deadbeat

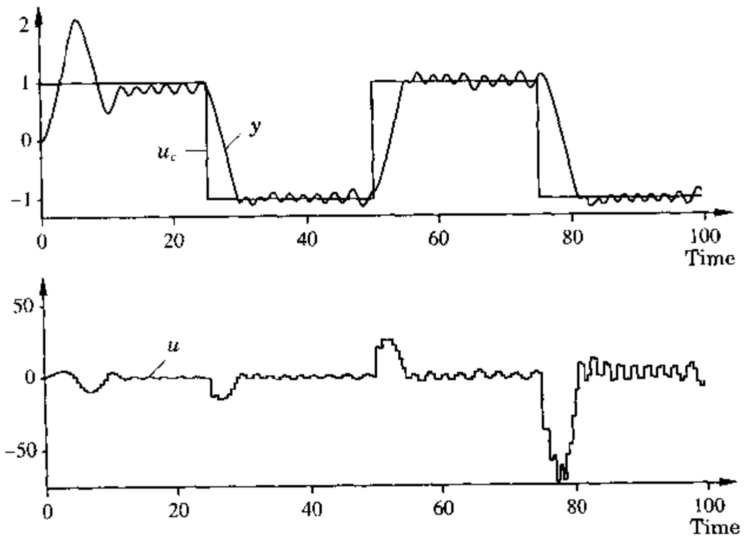

Figure 11.4 Simulation of adaptive control of the unstable process with a saturating actuator in Example 11.2.

observer. The controller is then

$$u(t) = f\Big(T^*(q^{-1})u_c(t) - S^*(q^{-1})y(t) + (1 - R^*(q^{-1}))u(t)\Big)$$

We illustrate windup by an example.

EXAMPLE 11.2 Windup and how to avoid it

Consider the simple example of adaptive control in Example 3.5. The process has the transfer function

$$G(s) = \frac{1}{s(s+1)}$$

Assume that there is an actuator that saturates when the magnitude of the control signal is 0.5. Figure 11.4 shows the behavior of the system if no precautions are taken in the controller. The figure clearly shows the detrimental effects of actuator saturation. The process runs open loop when the actuator saturates, and the output is drifting because the process has an integrator. This will also happen with a controller with fixed parameters. With an adaptive controller the saturation also causes the gain parameters (b_0 and b_1) to be underestimated. The controller gain is then too high, and the system becomes unstable. Windup is thus much more serious in adaptive control than in a controller with constant gain.

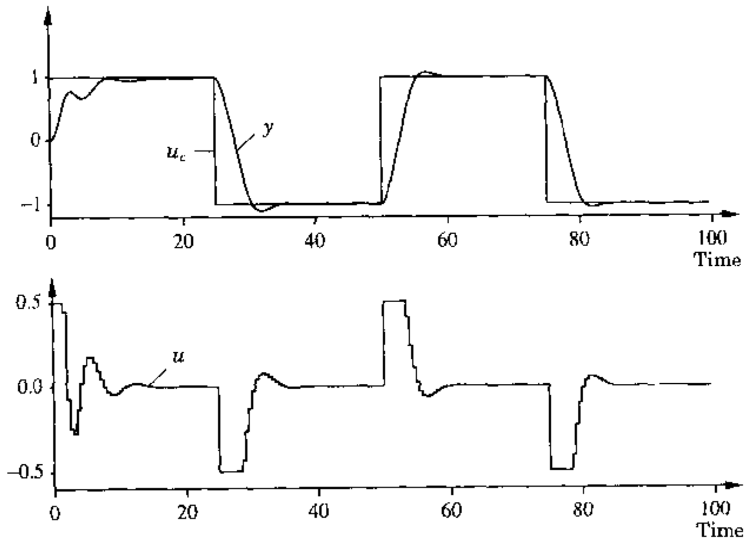

Figure 11.5 Simulation of an adaptive controller for a process with a saturating actuator with a controller having windup protection. Compare with Fig. 11.4, which shows the same simulation for a system without windup protection.

Figure 11.5 shows a simulation corresponding to Fig. 11.4 when the modification (11.1) is introduced to avoid windup. In this case there are clearly no difficulties. The control signal remains within the bounds $-0.5 < u < 0.5$ all the time. If the control signal saturates over a longer period of time, the adaptation should be switched off. □

Windup will always cause difficulties. In cases like Example 11.2 the phenomenon is serious because the process is unstable. If the controller is unstable, the control law given by Eqs. (11.1) will automatically reset the controller state with a speed corresponding to the observer dynamics.

11.3 CONTROLLER DESIGN

Automatic control design is another important part of an adaptive controller. Fairly simplistic design methods were used in developing the adaptive controllers in Chapters 3, 4, and 5. Simple methods were used to keep the overall complexity of the system reasonable. To achieve this, it was sometimes necessary to assume that the processes were minimum-phase systems. This was the case, for example, for model-reference adaptive systems. Since control design is done automatically in closed loop, it is also necessary to introduce safeguards to make sure that all conditions required for the design method are fulfilled. For instance, it may be necessary to test whether the estimated process model is minimum-phase or whether there are common factors in the estimated polynomials. Direct adaptive controllers have the advantage that the design step is eliminated, since the parameters of the regulator are estimated directly. Notice, however, that several assumptions are made implicitly in using the direct algorithms. Direct methods are also restricted to special classes of systems.

Design Procedures

Many different design procedures can be used for adaptive control. Feedback by itself can make a closed-loop system insensitive to variations in process dynamics. There are also special so-called robust design methods that take process uncertainty into account explicitly. In deriving an adaptive controller it seems appealing to base it on a robust design method. It is also of interest to try to combine robust and adaptive control. The estimator should then provide estimates of the model and its uncertainty. The design method should take the uncertainty into account. Unfortunately, control and estimation theory has not yet progressed to the state in which such estimation and control procedures are available. Many of the robust design methods do also require manual interaction. Such procedures cannot be used in an adaptive controller. The pole placement design procedure is quite useful in practice, in spite of its simplicity. However, it can be improved significantly by some simple modifications that give more robust closed-loop systems. Some ways to do this are discussed in this section.

Specifications

To obtain a robust controller, it is very important that specifications be chosen in a sensible way. With a pole placement design, this means that the desired closed-loop poles have to be chosen with care. Poles that are too fast will give controllers that are very sensitive. This can be understood from the following expression, which gives a sufficient condition for stability of a pole placement design:

$$|H(z) - H_0(z)| < \left| \frac{H(z)T(z)}{H_m(z)S(z)} \right|$$

In this expression, H is the pulse transfer function of model used to design the controller, H_0 is the pulse transfer function of the true plant, $H_m = B_m/A_m$ is the desired response, and S and T are the controller polynomials. The inequality should hold on the unit circle. The condition implies that high model precision is required for those frequencies at which the desired closed-loop system has significantly higher gain than the model.

A reasonable way to determine the closed-loop poles is to make the observation that it is difficult to obtain a crossover frequency that is significantly higher than the frequency at which the plant has a phase lag of $180^\circ - 270^\circ$. Notice that these frequencies can conveniently be determined by a relay feedback experiment.

Youla Parameterization

The Diophantine equation is a key element of pole placement design. This equation has many solutions. If the polynomials R^0 and S^0 are solutions of the equation

$$AR^0 + BS^0 = A_c^0$$

it follows that the polynomials R and S given by

$$\begin{aligned} R &= XR^0 + YB \\ S &= XS^0 - YA \end{aligned} \tag{11.2}$$

satisfy the equation

$$AR + BS = XA_c^0$$

If a controller characterized by the polynomials R^0 and S^0 gives a closed-loop system with the characteristic polynomial A_c^0, then the controller

$$(XR^0 + YB)u = -(XS^0 - YA)y \tag{11.3}$$

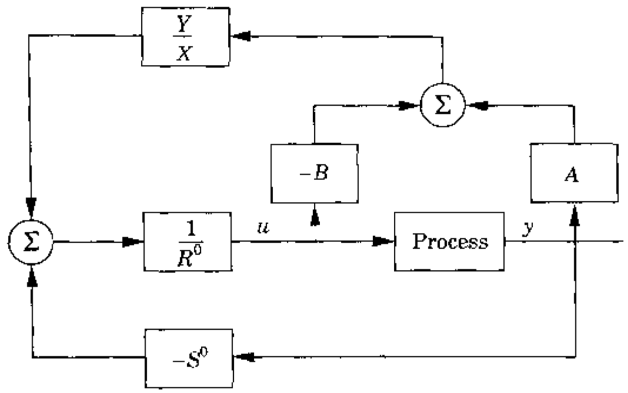

Figure 11.6 Block diagram of the closed-loop system with the controller (11.4).

gives a closed-loop system with the characteristic polynomial XA_c^0. The system is stable if the polynomial X is a stable polynomial but the polynomial Y can be chosen arbitrarily. It thus follows that if the controller $R^0u = -S^0y$ stabilizes the system $Ay = Bu$, then all controllers that stabilize the system are given by Eq. (11.3). The equation is called the *Youla parameterization* of all controllers that stabilize the system. Equation (11.3) can also be written as

$$u = -\frac{S^0}{R^0}y + \frac{Y}{XR^0}(Ay - Bu) \tag{11.4}$$

This control law is illustrated by the block diagram in Fig. 11.6.

Robust Pole Placement

The Youla parameterization can be used to impose extra conditions on the controller. This idea was used in Section 3.6 to obtain controllers that have integral action. We now use it to improve the robustness of the controller. One way to do this is to require that the controller have small gain for those frequencies at which the process is very uncertain. For example, it is useful to require that the controller should have zero gain at the Nyquist frequency. This is accomplished by the condition $S(-1) = 0$. We can also require that the controller gain be zero at frequency ω_0. This is equivalent to requiring that the polynomial

$$q^2 - 2\cos(\omega_0 h)q + 1$$

be a factor of $S(q)$. To satisfy such requirements, the order of the closed-loop system must thus be increased. The additional poles introduced are specified by the polynomial X. We illustrate this by an example.

EXAMPLE 11.3 Robust pole placement

Assume that we have obtained a controller R^0, S^0 that gives a closed-loop system with the characteristic polynomial A_c^0 and that we want to improve its robustness by requiring that $S(-1) = 0$. To do this, we introduce one more closed-loop pole. We choose

$$X(q) = q - x_0$$

with $|x_0| < 1$. The polynomial Y can also be of first order. Equation (11.2) gives

$$S(q) = (q - x_0)S^0(q) - (q - y_0)A(q)$$

Requiring that $S(-1) = 0$ gives

$$y_0 = -1 + \frac{(1 + x_0)S^0(-1)}{A(-1)}$$

The robust controller is then characterized by

$$R(q) = (q - x_0)R^0(q) + (q - y_0)B(q)$$
$$S(q) = (q - x_0)S^0(q) - (q - y_0)A(q)$$

Notice that it is possible to proceed recursively to make the controller more and more complex. □

Decoupled Command Signal and Disturbance Responses

Consider the process

$$Ay = Bu + v$$

and the controller

$$Ru = Tu_c - Sy$$

The closed-loop system is characterized by

$$y = \frac{BT}{A_c} u_c + \frac{R}{A_c} v$$
$$u = \frac{AT}{A_c} u_c - \frac{S}{A_c} v \tag{11.5}$$

where $A_c = AR + BS$ is the closed-loop characteristic polynomial. Assume that no process zeros are canceled, and factor the characteristic polynomial as $A_c = A_o A_m$. If we choose $T = T' A_o$, Eqs. (11.5) become

$$y = \frac{BT'}{A_m} u_c + \frac{R}{A_o A_m} v$$
$$u = \frac{AT'}{A_m} u_c - \frac{S}{A_o A_m} v$$

The command signal response is governed by the dynamics of A_m, but the disturbance response is governed by the dynamics of $A_o A_m$. In this sense it is thus coupling between the command signal response and the disturbance response. In some cases it is desirable that the dynamics of the command signal responses and the disturbance responses are completely decoupled. This can be achieved by requiring that $T = T' A_o$ and $R = R' A_m$. The closed-loop system is then characterized by

$$y = \frac{BT'}{A_m} u_c + \frac{R'}{A_o} v$$
$$u = \frac{AT'}{A_m} u_c - \frac{S}{A_o A_m} v$$

The Diophantine equation then becomes

$$AA_m R' + BS = A_o A_m$$

To have a causal controller, we must require that $\deg A_m \geq \deg A$. The minimum-degree causal solution to this equation is such that $\deg S = \deg A + \deg A_m - 1$. Furthermore, $\deg R = \deg A_m + \deg A_o - \deg A$. To have a causal controller, we must thus require that $\deg S \leq \deg R$. This implies that

$$\deg A + \deg A_m - 1 \leq \deg A_m + \deg A_o - \deg A$$

Hence $\deg A_o \geq 2 \deg A - 1$. We thus find that the minimum-degree solution that decouples the response to command signals and disturbances is such that

$$\deg A_m = n \qquad \deg A_o = 2n - 1 \qquad \deg R = \deg S = 2n - 1$$

where $n = \deg A$.

A design of this type can be very useful when there are very noisy measurements and a fast setpoint response is desired.

11.4 SOLVING THE DIOPHANTINE EQUATION

Several of the design methods discussed earlier involve the solution of a Diophantine equation

$$AR + BS = A_c \tag{11.6}$$

Efficient methods for solving this equation are needed. The equation is linear in the polynomials R and S. A solution always exists if A and B are relatively prime. However, the equation has many solutions. This is easily seen: If R^0 and S^0 are solutions, then

$$R = R^0 + BQ$$
$$S = S^0 - AQ$$

are also solutions, where Q is an arbitrary polynomial. A particular solution can be specified in several different ways. Since a controller must be causal, the condition $\deg S \le \deg R$ must hold. This condition will restrict the number of solutions significantly. An efficient way to solve the equation is to use a classical algorithm of Euclid.

Euclid's Algorithm

This algorithm finds the greatest common divisor G of two polynomials A and B. If one of the polynomials, say B, is zero, then G is equal to A. If this is not the case, the algorithm is as follows. Put $A_0 = A$ and $B_0 = B$ and iterate the equations

$$\begin{aligned} A_{n+1} &= B_n \\ B_{n+1} &= A_n \bmod B_n \end{aligned} \tag{11.7}$$

until $B_{n+1} = 0$. The greatest common divisor is then $G = B_n$. When A and B are polynomials, $A \bmod B$ means the remainder when A is divided by B. This is in full agreement with the case when A and B are numbers. Backtracking, we find that G can be expressed as

$$AX + BY = G \tag{11.8}$$

where the polynomials X and Y can be found by keeping track of $A_n \operatorname{div} B_n$ in Euclid's algorithm. This establishes the link between Euclid's algorithm and the Diophantine equation. The extended Euclidean algorithm gives a convenient way to determine X and Y as well as the minimum-degree solutions U and V to

$$AU + BV = 0 \tag{11.9}$$

Equations (11.8) and (11.9) can be written as

$$F \begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} X & Y \\ U & V \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} G \\ 0 \end{pmatrix} \tag{11.10}$$

The matrix F can thus be viewed as the matrix, which performs row operations on $\begin{pmatrix} A & B \end{pmatrix}^T$ to give $\begin{pmatrix} G & 0 \end{pmatrix}^T$. A convenient way to find F is to observe that

$$\begin{pmatrix} X & Y \\ U & V \end{pmatrix} \begin{pmatrix} A & 1 & 0 \\ B & 0 & 1 \end{pmatrix} = \begin{pmatrix} G & X & Y \\ 0 & U & V \end{pmatrix}$$

The extended Euclidean algorithm can be expressed as follows: Start with the matrix

$$M = \begin{pmatrix} A & 1 & 0 \\ B & 0 & 1 \end{pmatrix}$$

If we assume that $\deg A \ge \deg B$, then calculate $Q = A \operatorname{div} B$, multiply the second row of M by Q, and subtract from the first row. Then apply the same

procedure to the second row and repeat until the following matrix is obtained:

$$\begin{pmatrix} G & X & Y \\ 0 & U & V \end{pmatrix}$$

A nice feature of this algorithm is that possible common factors in A and B are determined automatically. The essential difficulty in implementing the algorithm is to find a good way to test for a polynomial being zero.

Solving the Diophantine Equation

By using the extended Euclidean algorithm it is now straightforward to solve the Diophantine equation

$$AR + BS = A_c \tag{11.11}$$

This is done as follows: Determine the greatest common divisor G and the associated polynomials X, Y, U, and V using the extended Euclidean algorithm. To have a solution to Eq. (11.11), G must divide A_c. A particular solution is given by

$$\begin{aligned} R^0 &= XA_c \operatorname{div} G \\ S^0 &= YA_c \operatorname{div} G \end{aligned} \tag{11.12}$$

and the general solution is

$$\begin{aligned} R &= R^0 + QU \\ S &= S^0 + QV \end{aligned} \tag{11.13}$$

where Q is an arbitrary polynomial. The minimum-degree solution is obtained by choosing $Q = -S^0 \operatorname{div} V$. This implies that $S = S^0 \bmod V$.

Relations to Ordinary Linear Equations

By equating coefficients of equal order, the Diophantine equation given by Eq. (11.11) can be written as a set of linear equations:

$$\begin{pmatrix} 1 & 0 & \dots & 0 & b_0 & 0 & \dots & 0 \\ a_1 & 1 & \ddots & \vdots & b_1 & b_0 & \ddots & \vdots \\ a_2 & a_1 & \ddots & 0 & b_2 & b_1 & \ddots & 0 \\ \vdots & \vdots & \ddots & 1 & \vdots & \vdots & \ddots & b_0 \\ a_n & \vdots & & a_1 & b_n & \vdots & & b_1 \\ 0 & a_n & & \vdots & 0 & b_n & & \vdots \\ \vdots & \ddots & \ddots & & \vdots & \ddots & \ddots & \\ 0 & \dots & 0 & a_n & 0 & \dots & 0 & b_n \end{pmatrix} \begin{pmatrix} r_1 \\ \vdots \\ r_k \\ s_0 \\ \vdots \\ s_l \end{pmatrix} = \begin{pmatrix} a_{c\,1} - a_1 \\ \vdots \\ a_{c\,n} - a_n \\ a_{c\,n+1} \\ \vdots \\ a_{c\,k+l+1} \end{pmatrix} \tag{11.14}$$

The matrix on the left-hand side is called the *Sylvester matrix*; it occurs frequently in applied mathematics. It has the property that it is nonsingular if and only if the polynomials A and B do not have any common factors. If there are no common factors, a unique solution to Eq. (11.14) exists. Notice, however, the nonuniqueness with respect to the orders of R and S. Different choices of k and l will give different R and S, as discussed above. The solution to Eq. (11.14) can be obtained by Gaussian elimination. This method does not use the special structure of the Sylvester matrix.

11.5 ESTIMATOR IMPLEMENTATION

There are many issues that have to be considered in the implementation of an estimator. This section can be summarized by the following motto:

> "Use only good relevant data, treat it carefully, and don't throw away useful information."

The key issue is that we want to obtain a model that is relevant for the task of control system design and that we want to track changes in the model. The tasks are influenced by many factors. In this section we discuss selection of model structure, filtering and excitation, parameter tracking, estimator windup, and robustness modifications.

Model Structure

The real physical processes that we try to control may have complicated dynamics. They may be nonlinear or infinite dimensional. One reason for the success of automatic control is that good control can often be based on relatively simple dynamical models. Such models can work very well under specific operating conditions, but the parameters of the model will depend on the operating conditions. In adaptive control it is attempted to fit a simple linear model on line and to adjust the parameters. For this purpose it is of paramount importance to understand what happens in fitting complicated dynamics with simple models. One fundamental fact is that the result obtained is crucially dependent on the nature of the input signal. This is illustrated by the following example.

EXAMPLE 11.4 Fitting low-order models to high-order systems

Consider a process with transfer function $G(s)$. Assume that one attempts to model the system by a first-order system with transfer function

$$\hat{G}(s) = \frac{b}{s+a}$$

If the input signal is sinusoidal with frequency ω_o, it is possible to get a perfect fit with finite values of the parameters if $\mathrm{Im}\{G(i\omega_o)\} \neq 0$. Straightforward calculations show that $\hat{G}(i\omega_o) = G(i\omega_o)$ if the parameters are chosen to be

$$a = -\frac{\omega_o \,\mathrm{Re}\{G(i\omega_o)\}}{\mathrm{Im}\{G(i\omega_o)\}}$$

$$b = -\frac{\omega_o |G(i\omega_o)|^2}{\mathrm{Im}\{G(i\omega_o)\}}$$

The transfer function of the model will then fit the data perfectly, but the parameters obtained depend on ω_o. The parameter values may change significantly with the frequency of the input signal. □

An interesting property of adaptive systems is that the parameters are estimated in closed loop. This implies that the simple model in the adaptive controller is fitted with the actual signals generated by the feedback. It explains intuitively the self-tuning property.

Another important observation is that the difficulty in on-line parameter estimation increases significantly with the number of parameters in the model. With many parameters the requirements on excitation also increase. For this purpose it is useful to try to reduce the number of unknown parameters as much as possible. This can be done by using *a priori* knowledge. This is often expressed in continuous-time models. The results are also strongly application dependent. Examples of this are given among the problems in the end of this chapter. Models consisting of low-order dynamics and time delays have proved very useful in process control. Such models can be represented by pulse transfer functions of the form

$$H_1(z) = \frac{b_0 z + b_1}{z^d(z + a_1)} \tag{11.15}$$

or

$$H_2(z) = \frac{b_0 z^2 + b_1 z + b_2}{z^d(z^2 + a_1 z + a_2)} \tag{11.16}$$

where the time delay τ is between dh and $dh + h$. Equation (11.16) can also represent second-order oscillatory systems. More b parameters can be included if the time delay is uncertain. In many cases it is known *a priori* that there are integrators in the model. This leads to transfer functions that contain the factor $z - 1$ in the denominator.

Data Filters and Excitation

Assume that the process is described by the discrete-time model

$$y(t) = G_0(q)u(t) + v(t) \tag{11.17}$$

Notice that possible anti-aliasing filters appear as part of the process $G_0(q)$. The disturbance $v(t)$ can be the sum of deterministic, piecewise deterministic,

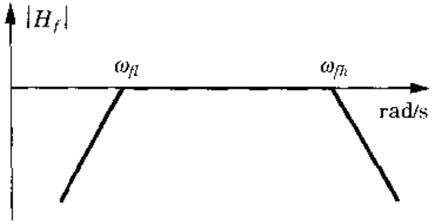

Figure 11.7 Amplitude curve for the data filter $H_f(q)$.

and stochastic disturbances. The signal has low-frequency and high-frequency components. In stochastic control problems it is important to design a controller that is tuned to a particular disturbance spectrum. In that case it is, of course, important to estimate the disturbance characteristics. In a deterministic problem we are concerned primarily with the term $G_0(q)u(t)$ in the above equation, and we are not particularly interested in the detailed character of the disturbance $v(t)$. In the following discussion we consider this case.

The presence of the disturbance $v(t)$ will, of course, create difficulties in the parameter estimation. However, the effect of $v(t)$ can be reduced by filtering. Assume that we introduce a data filter with the transfer function H_f and that we apply this filter to Eq. (11.17). Then

$$y_f(t) = G_0(q)u_f(t) + v_f(t) \tag{11.18}$$

where

$$y_f(t) = H_f(q)y(t) \qquad u_f(t) = H_f(q)u(t) \qquad \text{and} \qquad v_f(t) = H_f(q)v(t)$$

By a proper choice of the data filter we may now make the relative influence of the disturbance term smaller in Eq. (11.18) than in Eq. (11.17). The filtering should also emphasize the frequency ranges that are of primary importance for control design. The disturbance $v(t)$ typically has significant components with low frequencies. Low-frequency components should thus be reduced. Very high frequencies should similarly be attenuated. One reason for this is that if the model

$$A(q)y_f(t) = B(q)u_f(t)$$

is fitted by least squares, it is desirable that $A(q)v_f(t)$ be white noise. Since filtering with A implies that high frequencies are amplified, it means that $v_f(t)$ should not contain high frequencies. The data filter will therefore typically have band-pass character, as shown in Fig. 11.7. The center frequency is typically around the crossover frequency of the system.

In Section 3.5 we suggested using a filter with the transfer function

$$H_f(z) = \frac{1}{A_o(z^{-1})A_m(z^{-1})}$$

This filter is a typical low-pass filter that does not attenuate low frequencies. In Section 11.7 we will present other ways to choose the data filter. A typical data filter is given by

$$H_f(q) = \frac{(1-\alpha)(q-1)}{q-\alpha}$$

Some ways to choose the data filter will be discussed later.

It has been emphasized many times that it is necessary for the input signal to be persistently exciting of sufficiently high order to estimate parameters reliably. Taking into account that we are fitting low-order models to high-order systems, it is also necessary that persistency of excitation be achieved with signals in a frequency band where model accuracy is required.

Parameter Tracking

The key property of an adaptive controller is its ability to track variations in process dynamics. To do so, it is necessary to discount old data, a process that involves compromises. If parameters are constant, it is desirable to base the estimation on many measurements to reduce the effects of disturbances. If parameters are changing, however, it can be very misleading to use a long data record, since the parameters may not be the same. There are many ways to accommodate this problem. The best solutions are obtained if the nature of parameter variations is known. There are two prototype situations. One case is when parameters are slowly drifting; the other is when parameters are constant for long periods and jump from one value to another. Many attempts have been made to deal with the problem of parameter tracking, and there is a substantial literature. Most work is based on the assumption of detailed descriptions of the nature of parameter variations. A typical example is that the parameters are Markov processes with known transition probabilities. Such detailed information about the parameter variations is rarely available, and we therefore give some heuristic ways to deal with parameter tracking.

Exponential Forgetting

Exponential forgetting is a way to discard old data. It is based on the assumption that the least-squares loss function is replaced by a loss function in which old data is discounted exponentially. It follows from Theorem 2.4 that the recursive least-squares estimate with exponential forgetting is given by

$$\begin{aligned}
\hat\theta(t) &= \hat\theta(t-1) + K(t)\left(y(t) - \varphi^T(t)\hat\theta(t-1)\right) \\
K(t) &= P(t-1)\varphi(t)\left(\lambda + \varphi^T(t)P(t-1)\varphi(t)\right)^{-1} \\
P(t) &= \frac{1}{\lambda}\left(I - K(t)\varphi^T(t)\right)P(t-1)
\end{aligned} \tag{11.19}$$

Table 11.3 Relations between the ratio T_f/h and the coefficient λ.

T_f/h	λ
1	0.37
2	0.61
5	0.82
10	0.90
20	0.95
50	0.98
100	0.99

where the sampling period h was chosen as the time unit. The forgetting factor is given by

$$\lambda = e^{-h/T_f}$$

where T_f is the time constant for the exponential forgetting. To make an assessment of reasonable values of the forgetting factor, we give the values of the forgetting factor for different ratios T_f/h in Table 11.3.

It is possible to generalize the method with exponential forgetting and have different forgetting factors for different parameters. However, this requires information about the nature of the changes in different parameters. Another modification is to modify Eqs. (11.19) so that only the diagonal elements are divided by λ.

Tracking of a time-varying parameter is illustrated by an example.

EXAMPLE 11.5 Tracking parameters of a time-varying system

Consider a process described by the differential equation

$$\frac{dy}{dt} = -y(t) + K_p(t)u$$

where the process gain is time varying. The process is controlled by an indirect adaptive controller that estimates parameters of the discrete-time model

$$y(kh + h) + ay(kh) = bu(kh)$$

and designs a controller with integral action using robust pole placement with

$$A_m(q) = q + a_m = q - e^{-h/T_m}$$

and

$$A_o(q) = q + a_o = q - e^{-h/T_o}$$

Straightforward computations give a controller of the form

$$u(kh) = t_0 u_c(kh) + t_1 u_c(kh - h) - s_0 y(kh) - s_1 y(kh - h) + u(kh - h)$$

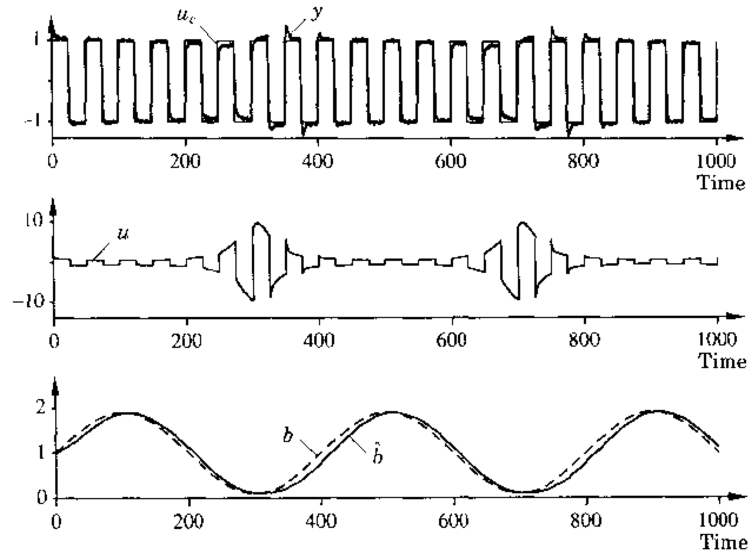

Figure 11.8 Tracking time-varying parameters.

where

$$t_0 = \frac{1 + a_m}{b}$$
$$t_1 = a_o t_0$$
$$s_0 = \frac{1 + a_o + a_m - a}{b}$$
$$s_1 = \frac{a_o a_m + a}{b}$$

Since only the process gain is unknown, we will estimate only the parameter b. In the simulation we will also assume that the gain varies sinusoidally between the values 0.1 and 1.9 with the period 400. In Fig. 11.8 we show a simulation of the system when the command signal is a square wave with period 50, there is measurement noise with the standard deviation 0.02, and the forgetting factor is $\lambda = 0.95$. Notice that the gain variation is clearly noticeable in the shape of the control signal, which changes significantly over one step. Figure 11.8 shows that the estimated gain lags the true gain. The forgetting factor is $\lambda = 0.95$, and the sampling period is $h = 0.5$. The time constant associated with the exponential forgetting is then $T_f = 10$ s, which is a crude estimate of the time lag in the estimator. Notice also that the lag is different for increasing and decreasing gains, a feature that indicates the nonlinear nature of the problem. The forgetting factor can be decreased to reduce the tracking lag. The estimates will then have more variation. To illustrate this, we simulate

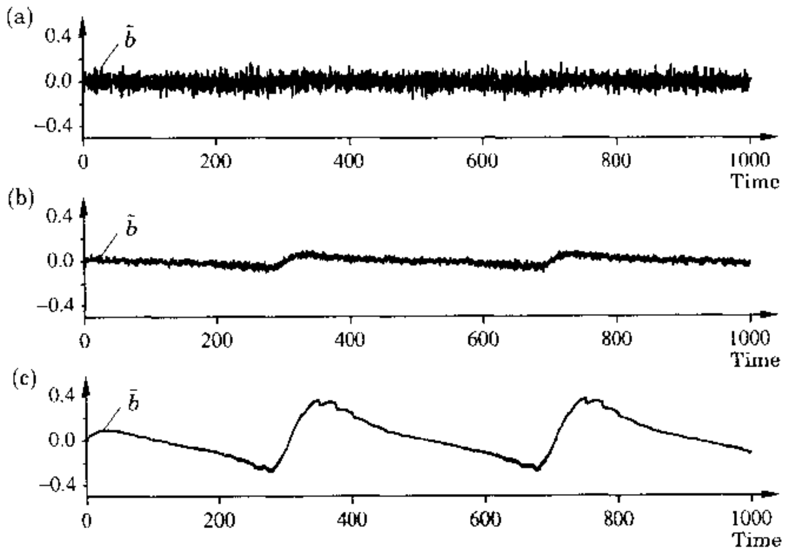

Figure 11.9 Parameter tracking error $\tilde{b} = b - \hat{b}$ for different forgetting factors: (a) $\lambda = 0.1$; (b) $\lambda = 0.7$; (c) $\lambda = 0.95$.

the same system as in Fig. 11.8 with different forgetting factors. The results are shown in Fig. 11.9. The figure shows that the forgetting factor $\lambda = 0.95$ is too large because the systematic tracking error is too large. The forgetting factor $\lambda = 0.1$, on the other hand, is too small, and the systematic error is small, but the random component is large. In this particular case the value $\lambda = 0.7$ is a reasonable compromise. The reason for the low value of λ is that the parameter variations are quite rapid. □

Covariance Resetting

In some situations the parameters are constant over long periods of time and change abruptly occasionally. Exponential forgetting, which is based on the assumption of a behavior that is homogeneous in time, is less suitable in this case. In such a situation it is more appropriate to reset the covariance matrix of the estimator to a large matrix when the changes occur. This is called *covariance resetting*. We illustrate this method by an example.

EXAMPLE 11.6 Covariance resetting

Consider the same system as in Example 11.5, but assume now that the parameter is piecewise constant. In Fig. 11.10 we show the results obtained

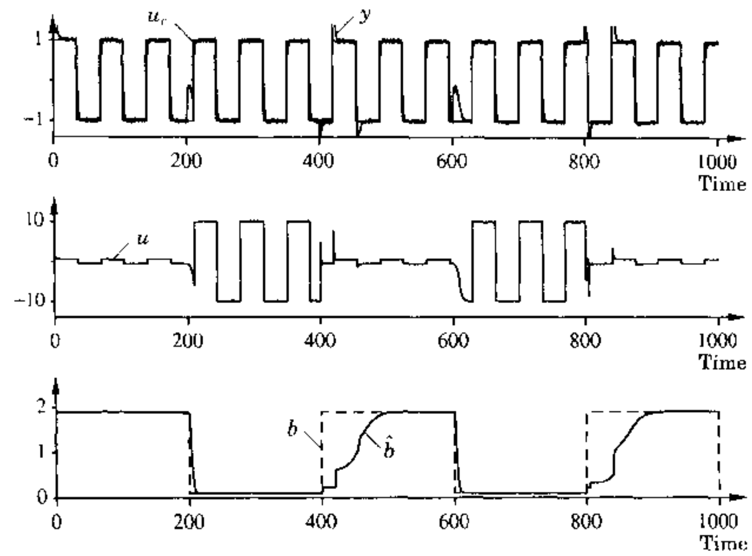

Figure 11.10 Tracking piecewise constant parameters using exponential forgetting when $\lambda = 0.95$.

with exponential forgetting with $\lambda = 0.95$. The figure shows clearly that the estimate of the process gain responds quite slowly when the gain changes. Notice also the strong asymmetry in the response of the estimate when the gain changes. It takes much longer for the estimate of the gain to increase than to decrease. The reason for this is the large difference in excitation. Also notice the stepwise nature of the estimates. Good excitation is obtained only when the command signal changes. In Fig. 11.11 we show the same system as in Fig. 11.10 with $\lambda = 1$ and covariance resetting. The covariance matrix is reset by reducing λ to 0.0001 when the parameter changes. Notice the drastic difference in the tracking rate. □

The example clearly illustrates the advantage of using covariance resetting when the parameters change abruptly. To use this effectively, it is necessary to detect the changes in the parameters. There are many ways to do this by analyzing residuals or parameter changes. It is also possible to reset the covariance periodically.

Parallel Estimators and Other Schemes

There are many other ways to deal with parameter tracking. One possibility is to have several parallel estimators with different forgetting factors and to choose the one in which the estimates have the smallest residuals. It is also

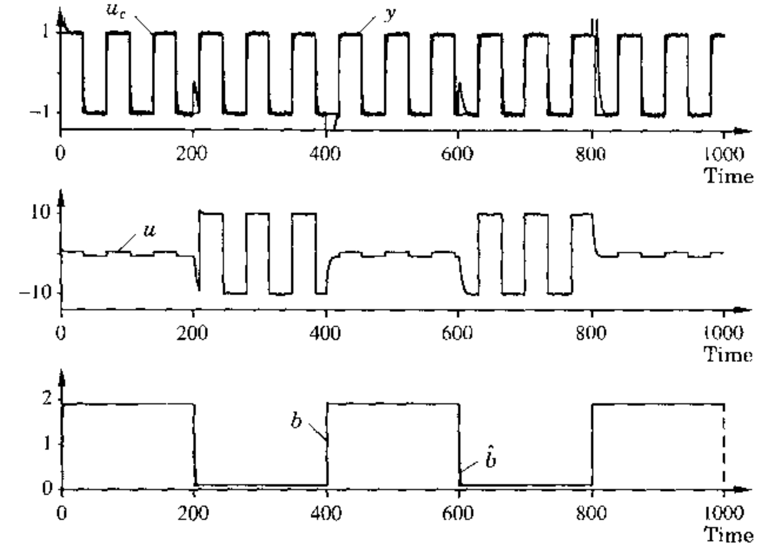

Figure 11.11 Tracking piecewise constant parameters using covariance resetting.

possible to have several parallel estimators that are reset periodically in a staggered way. There are also other schemes in which the forgetting factor is made signal dependent.

Estimator Windup

Exponential forgetting works well only if the process is properly excited all the time. There are problems with exponential forgetting when the excitation is poor. To understand this, we first consider the extreme case in which there is no excitation at all, that is, $\varphi = 0$. The equations for the estimate then become

$$\theta(t+1) = \theta(t)$$
$$P(t+1) = \frac{1}{\lambda} P(t)$$

The equation for the estimate θ is thus unstable with all eigenvalues equal to 1, and the equation for the P-matrix is unstable with all eigenvalues equal to $1/\lambda$. In this case the estimate will thus remain constant, and the P-matrix will grow exponentially if $\lambda < 1$. Since the estimator gain is $P\varphi$, the gain of the estimator will also grow exponentially. This means that the estimates may change very drastically whenever φ becomes different from zero. The phenomenon is called *estimator windup* in analogy with integrator windup.

A similar situation occurs if the regression vector is different from zero but restricted to a subspace. We illustrate this by an example.

EXAMPLE 11.7 A constant regression vector causes windup

Consider a process with the transfer function

$$G(s) = \frac{\beta}{s+\alpha}$$

with an indirect adaptive controller based on estimation of parameters a and b in the discrete-time model

$$y(kh+h) + ay(kh) = bu(kh)$$

The control design is the same as in Example 11.5. The controller has integral action. The parameters have the values $\alpha = 1$ and $\beta = 1$, the sampling period is $h = 0.5$ s, there is measurement noise with a standard deviation of 0.05, the setpoint is piecewise constant, and the forgetting factor is $\lambda = 0.95$. To illustrate the effect of poor excitation, the setpoint will be kept constant for long periods of time.

The parameters are in R^2. To have excitation, the regression vectors should also span R^2 persistently. When the setpoint is constant, the input and the

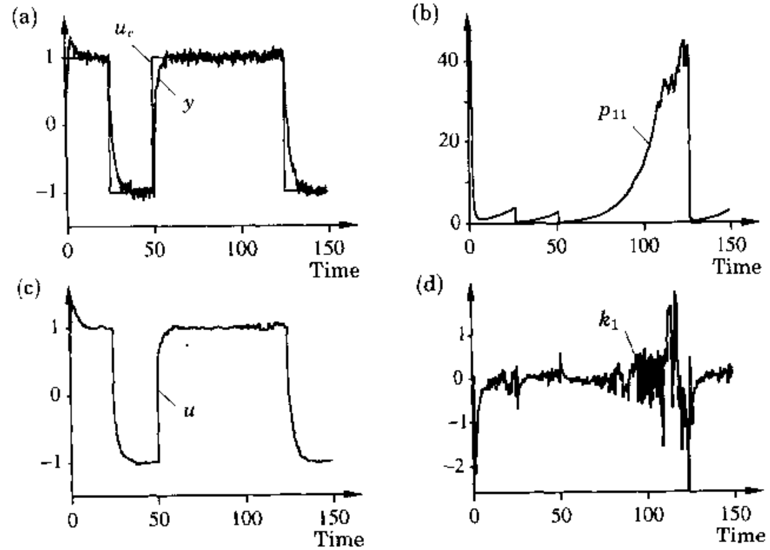

Figure 11.12 Illustration of estimator windup due to poor excitation. (a) Output y and setpoint u_c; (b) covariance p_{11}; (c) control signal u; and (d) estimator gain k_1.

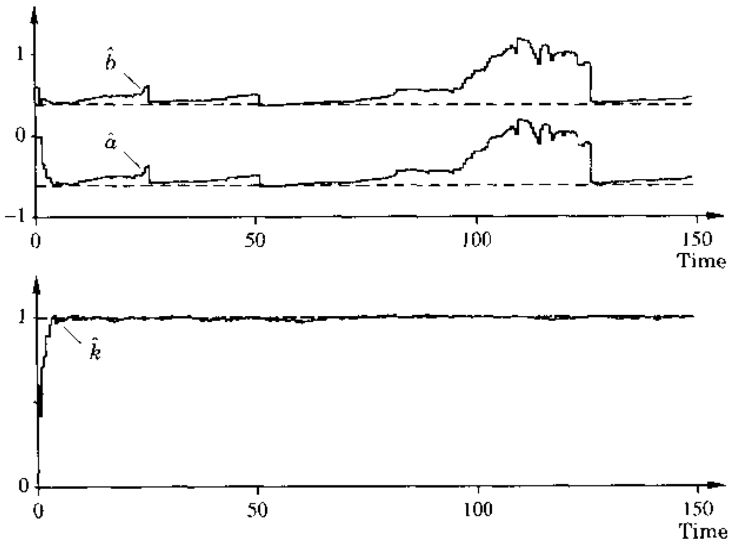

Figure 11.13 Parameter estimates in the case of estimator windup due to poor excitation. The dashed lines show the correct values of the parameters and of the gain of the process.

output settle to constant values after a transient. The regression vector then becomes

$$\varphi(t) = \begin{pmatrix} -u_c & au_c/b \end{pmatrix}$$

This vector lies in a one-dimensional subspace of R^2 and is thus not persistently exciting. The simulation shown in Fig. 11.12 illustrates the behavior of the system. The process output tracks the command signal quite well, and the control signal is also quite reasonable. Figure 11.12 shows that the element p_{11} of the P-matrix grows approximately exponentially during the periods when the command signal is constant. The deviations are due to the measurement noise that gives some excitation. The other elements of the P-matrix behave similarly. The estimator gains also grow significantly. Exponential growth of the P-matrix and the associated increase in the estimator gains are clearly noticeable in Fig. 11.12. The parameter estimates will change significantly, as shown in Fig. 11.13. The estimates are very inaccurate at the end of the periods when the command signal has been constant. In Fig. 11.13 we also show the estimate of process gain calculated from

$$\hat{k} = \frac{\hat{b}}{1+\hat{a}}$$

This estimate is very good for the whole period because this variable is well

excited. This is why the controller behaves reasonably well in spite of the poor estimates of a and b. □

To get further insight into the windup phenomenon, we make a simplified analysis of the behavior shown in Example 11.7. For this purpose we assume that the regression vector is constant, that is, $\varphi(t) = \varphi_0$. The inverse of the P-matrix is given by

$$P^{-1}(t+1) = \lambda^t P_0^{-1} + \sum_{k=1}^{t} \lambda^{t-k}\varphi_0\varphi_0^T = \lambda^t P_0^{-1} + \frac{1-\lambda^t}{1-\lambda}\varphi_0\varphi_0^T$$

Using the matrix inversion lemma (Lemma 2.1), we find after some calculations that the covariance matrix can be written as

$$P(t+1) = \frac{1}{\lambda^t}\left(P_0 - \frac{P_0\varphi_0\varphi_0^T P_0}{\lambda^t\alpha(t) + \varphi_0^T P_0\varphi_0}\right)$$

where

$$\alpha(t) = \frac{1-\lambda}{1-\lambda^t}$$

Furthermore, we find that

$$P(t+1)\varphi_0 = a(t)\frac{P_0\varphi_0}{\lambda^t a(t) + \varphi_0^T P_0\varphi_0} \tag{11.20}$$

The P-matrix can be decomposed as

$$P(t+1) = \tilde{P}(t) + \beta(t)\varphi_0\varphi_0^T$$

where

$$\tilde{P}(t) = \frac{1}{\lambda^t}\left(P_0 - \frac{P_0\varphi_0\varphi_0^T P_0}{\lambda^t\alpha(t) + \varphi_0^T P_0\varphi_0}\right) - \beta(t)\varphi_0\varphi_0^T$$

and

$$\beta(t) = \alpha(t)\frac{\varphi_0^T P_0\varphi_0}{(\varphi_0^T\varphi_0)^2(\lambda^t\alpha(t) + \varphi_0^T P_0\varphi_0)}$$

The matrix $\tilde{P}(t)$ is of rank $n-1$ with $\tilde{P}(t)\varphi_0 = 0$. Since $|\lambda| < 1$ we have as $t \to \infty$

$$a(t) \to 1-\lambda$$
$$b(t) \to \frac{1-\lambda}{(\varphi_0^T\varphi_0)^2}$$

In the decomposition of $P(t+1)$ we thus find that the matrix $\tilde{P}$ goes to infinity as λ^{-t} and that $\beta(t)\varphi_0\varphi_0^T$ goes to a constant $(1-\lambda)\varphi_0\varphi_0^T/(\varphi_0^T\varphi)^2$.

Intuitively, the result of the calculation can be interpreted as follows: When the regression vector is constant, we obtain information only about the component of the parameter that is parallel to the regression vector. This component can be estimated reliably with exponential forgetting. The "projection" of the

P-matrix in this direction converges to $1 - \lambda$, and the "orthogonal" part of the P-matrix goes to infinity as λ^{-t}. Estimator windup is thus obtained by exponential forgetting combined with poor excitation. There are several ways to avoid estimator windup. We now discuss some of these techniques.

Conditional Updating

One possibility to avoid windup in the estimator is to update the estimate and the covariance only when there is excitation. The algorithms obtained are called algorithms with conditional updating or dead zones. A correct detection of excitation should be based on calculation of covariances or spectra as discussed in Section 2.4. Simpler conditions are often used in practice. Common tests are based on the magnitudes of the variations in process inputs and outputs or other signals such as ε and $\varphi^T P \varphi$. Notice that the quantity $\varphi^T P \varphi$ is dimension free.

If the regression vector is constant, it follows from Eq. (11.20) that

$$\varphi_0^T P(t)\varphi_0 = a(t)\,\frac{\varphi_0^T P_0 \varphi_0}{\lambda^t a(t) + \varphi_0^T P_0 \varphi_0}$$

As $t \to \infty$, it follows that $a(t) \to 1 - \lambda$. If $\varphi^T P \varphi$ is used as a test quantity, it is thus natural to normalize it by $1 - \lambda$. The effect of conditional updating is illustrated by an example.

EXAMPLE 11.8 Conditional updating

Consider the system in Example 11.7, but modify the estimator to provide conditional updating. In this particular case the estimate is updated if the test quantity

$$\varphi(t)^T P(t)\varphi(t) > 2(1 - \lambda)$$

Figure 11.14 shows a simulation that is comparable to Fig. 11.12. Notice that the exponential growth is now avoided. The elements of the P-matrix remain bounded, and the estimator gains are well behaved. □

The selection of the condition for updating is critical. If the criterion is too stringent, the estimates will be poor because updating is done too infrequently. If the criterion is too liberal, we get covariance windup.

Constant-Trace Algorithms

Another way to keep the P-matrix bounded is to scale the matrix at each iteration. A popular scheme is to scale it in such a way that the trace of the matrix is constant. An additional refinement is to also add a small unit matrix.

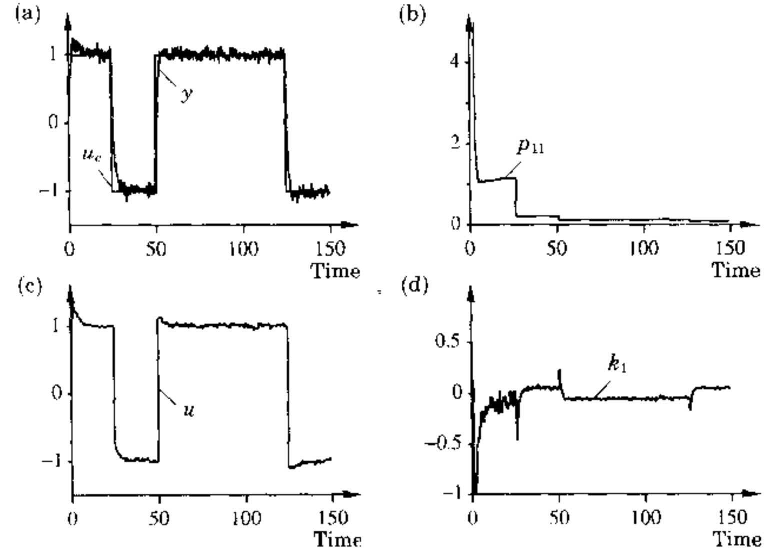

Figure 11.14 Illustration of how estimator windup can be avoided with conditional updating. Compare with Fig. 11.12.

This gives the so-called *regularized constant-trace algorithm*:

$$
\begin{aligned}
\hat\theta(t) &= \hat\theta(t-1) + K(t)\Big(y(t) - \varphi^T(t)\hat\theta(t-1)\Big)\\
K(t) &= P(t-1)\varphi(t)\left(\lambda + \varphi(t)^T P(t-1)\varphi(t)\right)^{-1}\\
\bar P(t) &= \frac{1}{\lambda}\left(\bar P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{1 + \varphi(t)^T P(t-1)\varphi(t)}\right)\\
P(t) &= c_1\,\frac{\bar P(t)}{tr\left(\bar P(t)\right)} + c_2 I
\end{aligned}
\tag{11.21}
$$

where $c_1 > 0$ and $c_2 \geq 0$. Typical values for the parameters can be

$$c_1/c_2 \approx 10^4$$
$$\varphi^T\varphi \cdot c_1 \gg 1$$

The constant-trace algorithm may also be combined with conditional updating.

Directional Forgetting

Another way to forget old data is based on the fact that one observation gives a projection of the parameter on the regression vector. Exponential forgetting can then be done only in the "direction" of the regression vector. This approach is called *directional forgetting*. To derive the equations, we observe that the inverse of the P-matrix with exponential forgetting is given by

$$P^{-1}(t+1) = \lambda P^{-1}(t) + \varphi(t)\varphi^T(t)$$

In directional forgetting we start with the formula

$$P^{-1}(t+1) = P^{-1}(t) + \varphi(t)\varphi^T(t)$$

The matrix $P^{-1}(t)$ is decomposed as

$$P^{-1}(t) = \tilde{P}^{-1}(t) + \gamma(t)\varphi(t)\varphi^T(t) \tag{11.22}$$

where $\tilde{P}^{-1}(t)\varphi(t) = 0$. This gives

$$\gamma(t) = \frac{\varphi^T(t)P^{-1}(t)\varphi(t)}{(\varphi^T(t)\varphi(t))^2}$$

Exponential forgetting is then applied only to the second term of Eq. (11.22), which corresponds to the direction where new information is obtained. This gives

$$P^{-1}(t+1) = \tilde{P}^{-1}(t) + \lambda\gamma(t)\varphi(t)\varphi^T(t) + \varphi(t)\varphi^T(t)$$

which can be written as

$$P^{-1}(t+1) = P^{-1}(t) + \left(1 + (\lambda - 1)\frac{\varphi^T(t)P^{-1}(t)\varphi(t)}{(\varphi^T(t)\varphi(t))^2}\right)\varphi(t)\varphi^T(t)$$

There are several variations of the algorithms. The forgetting factor is sometimes made a function of the data. One method has the property that the P-matrix is driven toward a matrix proportional to the identity matrix when there is poor excitation.

Leakage

Another way to avoid estimator windup, called *leakage*, was discussed in Section 6.9. In continuous time the estimator was modified as shown in Eq. (6.84) by adding the term $\alpha(\theta^0 - \theta)$. This means that the parameters will converge to θ^0 when no useful information is obtained, that is, when $e = 0$. A similar modification can also be made in discrete-time estimators. When a least-squares type of algorithm is used, it is also common to add a similar term to the P equation to drive it toward a specified matrix.

Robust Estimation

The least-squares estimate is optimal if the disturbances are Gaussian and such that the equation error is white noise. In practice the least-squares

estimate has some drawbacks because the assumptions are violated. It is a direct consequence of the least-squares formulation that a single large error will have a drastic influence on the result because the errors are squared in the criterion. This is a consequence of the Gaussian assumption that implies that the probability of large errors is very small. Estimators with very different properties are obtained if it is assumed that the probability for large errors is not negligible. Without going into technicalities, we remark that the estimators will be replaced by equations such as

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi(t-1)f(\varepsilon(t))$$

$$\frac{d\hat{\theta}}{dt} = P\varphi f(\varepsilon)$$

where the function $f(\varepsilon)$ is linear for small ε but increases more slowly than linear for large ε. A typical example is

$$f(\varepsilon) = \frac{\varepsilon}{1 + a|\varepsilon|}$$

The net effect is to decrease the consequences of large errors. The estimators are then called *robust*.

11.6 SQUARE ROOT ALGORITHMS

It is well known in numerical analysis that considerable accuracy may be lost when a least-squares problem is solved by forming and solving the normal equations. The reason is that the measured values are squared unnecessarily. The following procedure for solving the least-squares problem is much better conditioned numerically. Start with Eq. (2.4):

$$\mathcal{E} = Y - \Phi\theta$$

An orthogonal transformation Q, that is, $Q^TQ = QQ^T = I$, does not change the Euclidean norm of the error

$$\tilde{\mathcal{E}} = Q\mathcal{E} = QY - Q\Phi\theta$$

Choose the transformation Q so that $Q\Phi$ is upper triangular. The above equation then becomes

$$\begin{pmatrix} \tilde{e}^1 \\ \tilde{e}^2 \end{pmatrix} = \begin{pmatrix} \tilde{y}^1 \\ \tilde{y}^2 \end{pmatrix} - \begin{pmatrix} \tilde{\Phi}_1 \\ 0 \end{pmatrix}\theta$$

where $\tilde{\Phi}_1$ is upper triangular. It then follows that the least-squares estimate is given by

$$\tilde{\Phi}_1\theta = \tilde{y}^1$$

and the error is $(\tilde{e}^2)^T\tilde{e}^2$. This way of computing the estimate is much more accurate than solving the normal equation, particularly if $\|\mathcal{E}\| \ll \|Y\|$. The method based on orthogonal transformation is called a *square root method* because it works with Φ or the square root of $\Phi^T\Phi$. There are several numerical methods that can be used to find an orthogonal transformation Q, for example, Householder transformations or the QR method. We will not discuss these methods further, because we are primarily interested in recursive methods.

Representation of Conditional Mean Values

Recursive square root methods can naturally be explained by using probabilistic arguments. Some preliminary results on conditional mean values for Gaussian random variables will first be developed. We can now show the following result.

THEOREM 11.1 Conditional mean values and covariances

Let the vectors x and y be jointly Gaussian random variables with mean values

$$E\begin{pmatrix} y \\ x \end{pmatrix} = \begin{pmatrix} m_y \\ m_x \end{pmatrix} \tag{11.23}$$

and covariance

$$\operatorname{cov}\begin{pmatrix} y \\ x \end{pmatrix} = \begin{pmatrix} R_y & R_{yx} \\ R_{xy} & R_x \end{pmatrix} = R \tag{11.24}$$

where $R_{xy} = R_{yx}^T$. Further assume that $\dim x = n$ and $\dim y = p$. The conditional mean value of x, given y, is Gaussian with mean

$$E(x|y) = m_x + R_{xy}R_y^{-1}(y - m_y) \tag{11.25}$$

and covariance

$$\operatorname{cov}(x|y) = R_{x|y} = R_x - R_{xy}R_y^{-1}R_{yx} \tag{11.26}$$

A nonnegative matrix R can be decomposed as

$$R = \rho\begin{pmatrix} I & 0 \\ K & L_x \end{pmatrix}\begin{pmatrix} D_y & 0 \\ 0 & D_x \end{pmatrix}\begin{pmatrix} I & 0 \\ K & L_x \end{pmatrix}^T \tag{11.27}$$

where D_x and D_y are diagonal matrices and L_x is lower triangular. Then

$$R_{xy}R_y^{-1} = K \tag{11.28}$$

and

$$R_{x|y} = \rho L_x D_x L_x^T \tag{11.29}$$

Proof: We first show that the vector z defined by

$$z = x - m_x - R_{xy}R_y^{-1}(y - m_y) \tag{11.30}$$

has zero mean, is independent of y, and has the covariance

$$R_z = R_x - R_{xy}R_y^{-1}R_{yx} \tag{11.31}$$

The mean value is zero. Furthermore,

$$\begin{aligned} Ez(y - m_y)^T &= E\left\{(x - m_x)(y - m_y)^T - R_{xy}R_y^{-1}(y - m_y)(y - m_y)^T\right\} \\ &= R_{xy} - R_{xy}R_y^{-1}R_y = 0 \end{aligned}$$

The variables z and y are thus uncorrelated. Since they are Gaussian, they are also independent. It now follows that

$$\begin{pmatrix} y - m_y \\ x - m_x \end{pmatrix} = \begin{pmatrix} I & 0 \\ R_{xy}R_y^{-1} & I \end{pmatrix}\begin{pmatrix} y - m_y \\ z \end{pmatrix}$$

The joint density function of x and y is

$$\begin{aligned} f(x,y) = {} & (2\pi)^{-(n+p)/2}(\det R)^{-1/2} \\ & \exp\left\{-\frac{1}{2}\left(z^T R_z^{-1} z + (y - m_y)^T R_y^{-1}(y - m_y)\right)\right\} \end{aligned}$$

The density function of y is

$$f(y) = (2\pi)^{-p/2}(\det R_y)^{-1/2}\exp\left\{-\frac{1}{2}(y - m_y)^T R_y^{-1}(y - m_y)\right\}$$

where p is the dimension of y. The conditional density is then

$$f(x|y) = \frac{f(x,y)}{f(y)} = (2\pi)^{-n/2}(\det R_y)^{1/2}(\det R)^{-1/2}\exp\left\{-\frac{1}{2}z^T R_z^{-1} z\right\}$$

where n is the dimension of x. But

$$\begin{aligned} \det R &= \det\begin{pmatrix} R_y & R_{yx} \\ R_{xy} & R_x \end{pmatrix} = \det\begin{pmatrix} R_y & R_{yx} \\ 0 & R_x - R_{xy}R_y^{-1}R_{yx} \end{pmatrix} \\ &= \det R_y \cdot \det\left(R_x - R_{xy}R_y^{-1}R_{yx}\right) = \det R_y \cdot \det R_z \end{aligned}$$

Hence

$$f(x|y) = (2\pi)^{-n/2}(\det R_z)^{-1/2}e^{-(1/2)z^T R_z^{-1} z}$$

where z is given by Eq. (11.30) and R_z by Eq. (11.31).

The first part of the theorem is thus proved. To show the second part, notice that Eq. (11.27) is

$$R = \rho\begin{pmatrix} D_y & D_y K^T \\ K D_y & L_x D_x L_x^T + K D_y K^T \end{pmatrix}$$

Identification of the different terms gives

$$\begin{aligned} R_y &= \rho D_y \\ R_{xy} &= \rho K D_y \\ R_x &= \rho\left(L_x D_x L_x^T + K D_y K^T\right) \end{aligned}$$

Hence

$$R_{xy}R_y^{-1} = K$$

and

$$R_{x|y} = R_x - R_{xy}R_y^{-1}R_{yx} = \rho L_x D_x L_x^T$$

Remark. It follows from the theorem that the calculation of the conditional mean of a Gaussian random variable is equivalent to transforming the joint covariance matrix of the variables to the form of Eq. (11.27). Notice that this form may be viewed as a square root representation of R. □

Application to Recursive Estimation

The basic step in recursive estimation can be described as follows: Let θ be Gaussian $N(\theta^0, P)$. Assume that a linear observation

$$y = \varphi^T\theta + e$$

is made, where e is normal $N(0, \sigma^2)$. The new estimate is then given as the conditional mean $E(\theta|y)$. The joint covariance matrix of y and θ is

$$R = \begin{pmatrix} \varphi^T P\varphi & \varphi^T P \\ P\varphi & P \end{pmatrix} + \begin{pmatrix} \sigma^2 & 0 \\ 0 & 0 \end{pmatrix}$$

The symmetric nonnegative matrix P has a decomposition $P = LDL^T$, where L is a lower triangular matrix with unit diagonal and D is a nonnegative diagonal matrix. The matrix R can then be written as

$$\begin{aligned} R &= \begin{pmatrix} \varphi^T LDL^T\varphi + \sigma^2 & \varphi^T LDL^T \\ LDL^T\varphi & LDL^T \end{pmatrix} \\ &= \begin{pmatrix} 1 & \varphi^T L \\ 0 & L \end{pmatrix} \begin{pmatrix} \sigma^2 & 0 \\ 0 & D \end{pmatrix} \begin{pmatrix} 1 & 0 \\ L^T\varphi & L^T \end{pmatrix} \end{aligned} \tag{11.32}$$

If this matrix can be transformed to

$$R = \begin{pmatrix} 1 & 0 \\ K & \tilde{L} \end{pmatrix} \begin{pmatrix} \tilde{\sigma}^2 & 0 \\ 0 & \tilde{D} \end{pmatrix} \begin{pmatrix} 1 & K^T \\ 0 & \tilde{L}^T \end{pmatrix} \tag{11.33}$$

Theorem 11.1 can be used to obtain the recursive estimate as

$$\hat{\theta} = \theta^0 + K\left(y - \varphi^T\theta\right)$$

with covariance

$$P = \tilde{L}\tilde{D}\tilde{L}^T$$

The algorithm can thus be described as follows.

ALGORITHM 11.1 Square root RLS

Step 1: Start with L and D as a representation of P.

Step 2: Form the matrix of Eq. (11.32), where φ is the regression vector.

Step 3: Reduce this to the lower triangular form of Eq. (11.33).

Step 4: The updating gain is K, and the new P is represented by $\tilde{L}$ and $\tilde{D}$. □

It now remains to find the appropriate transformation matrices. A convenient method is dyadic decomposition.

Dyadic Decomposition

Given vectors

$$a = \begin{pmatrix} 1 & a_2 & \ldots & a_n \end{pmatrix}^T$$
$$b = \begin{pmatrix} b_1 & b_2 & \ldots & b_n \end{pmatrix}^T$$

and scalars α and β, find new vectors

$$\tilde{a} = \begin{pmatrix} 1 & \tilde{a}_2 & \ldots & \tilde{a}_n \end{pmatrix}^T$$
$$\tilde{b} = \begin{pmatrix} 0 & \tilde{b}_2 & \ldots & \tilde{b}_n \end{pmatrix}^T$$

such that

$$\alpha aa^T + \beta bb^T = \tilde{\alpha}\tilde{a}\tilde{a}^T + \tilde{\beta}\tilde{b}\tilde{b}^T \tag{11.34}$$

If this problem can be solved, we can perform the composition of Eq. (11.33) by repeated application of the method.

Equation (11.34) can be written as

$$\begin{aligned} &\alpha \begin{pmatrix} 1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \begin{pmatrix} 1 & a_2 & \ldots & a_n \end{pmatrix} + \beta \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \begin{pmatrix} b_1 & b_2 & \ldots & b_n \end{pmatrix} \\ &= \tilde{\alpha} \begin{pmatrix} 1 \\ \tilde{a}_2 \\ \vdots \\ \tilde{a}_n \end{pmatrix} \begin{pmatrix} 1 & \tilde{a}_2 & \ldots & \tilde{a}_n \end{pmatrix} + \tilde{\beta} \begin{pmatrix} 0 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_n \end{pmatrix} \begin{pmatrix} 0 & \tilde{b}_2 & \ldots & \tilde{b}_n \end{pmatrix} \end{aligned} \tag{11.35}$$

Equating the (1, 1) elements gives

$$\alpha + \beta b_1^2 = \tilde{\alpha} \tag{11.36}$$

Equating the $(1, k)$ elements for $k > 1$ gives

$$\alpha a_k + \beta b_1 b_k = \tilde{\alpha}\tilde{a}_k \tag{11.37}$$

Adding and subtracting $\beta b_1^2 a_k$ give

$$(\alpha + \beta b_1^2)a_k + \beta b_1 b_k - \beta b_1^2 a_k = \tilde{\alpha}\tilde{a}_k$$

Hence

$$\tilde{a}_k = a_k + \frac{\beta b_1}{\tilde{\alpha}}(b_k - b_1 a_k) \tag{11.38}$$

The numbers $\tilde{\alpha}$ and $\tilde{a}_k$ can thus be determined. It now remains to compute $\tilde{\beta}$ and $\tilde{b}_k$. Equating the (k, l) elements of Eq. (11.35) for $k, l > 1$ gives

$$\begin{aligned}\alpha a_k a_l + \beta b_k b_l &= \tilde{\alpha}\tilde{a}_k\tilde{a}_l + \tilde{\beta}\tilde{b}_k\tilde{b}_l\\ &= \frac{(\alpha a_k + \beta b_1 b_k)(\alpha a_l + \beta b_1 b_l)}{\tilde{\alpha}} + \tilde{\beta}\tilde{b}_k\tilde{b}_l\end{aligned}$$

where Eq. (11.37) has been used to eliminate $\tilde{a}_k\tilde{a}_l$. Inserting the expression in Eq. (11.36) for $\tilde{\alpha}$ gives, after some calculations,

$$(b_k - b_1 a_k)(b_l - b_1 a_l) = \frac{\tilde{\alpha}\tilde{\beta}}{\alpha\beta}\tilde{b}_k\tilde{b}_l$$

```
PROCEDURE
DyadicReduction(VAR a,b:col; VAR alpha,beta:REAL;
  i0,i1,i2 :CARDINAL);
CONST
  mzero = 1.0E-10;
VAR
  i             : CARDINAL;
  w1,w2,b1,gam : REAL;
BEGIN
  IF beta<mzero THEN beta:=0.0; END;
  b1 := b[i0];
  w1 := alpha;
  w2 := beta*b1;
  alpha := alpha + w2*b1;
  IF alpha > mzero THEN
    beta := w1*beta/alpha;
    gam := w2/alpha;
    FOR i:=i1 TO i2 DO
      b[i] := b[i] - b1*a[i];
      a[i] := a[i] + gam*b[i];
    END;
  END;
END DyadicReduction;
```

Figure 11.15 Dyadic decomposition.

These equations have several solutions. A simple one is

$$\tilde{b}_k = b_k - b_1 a_k$$

$$\tilde{\beta} = \frac{\alpha\beta}{\tilde{\alpha}}$$

A solution to the dyadic decomposition problem of Eq. (11.34) is given by the equations

$$\begin{aligned}\tilde{\alpha} &= \alpha + \beta b_1^2\\ \tilde{\beta} &= \frac{\alpha\beta}{\tilde{\alpha}}\\ \gamma &= \frac{\beta b_1}{\tilde{\alpha}}\\ \tilde{b}_k &= b_k - b_1 a_k \qquad k = 2, \ldots, n\\ \tilde{a}_k &= a_k + \gamma\tilde{b}_k \qquad k = 2, \ldots, n\end{aligned}$$

The algorithm in Fig. 11.15 is an implementation of the dyadic decomposition.

```
PROCEDURE
LDFilter(VAR theta,d:col; VAR l:matr; phi:col;
  lambda:REAL; n:CARDINAL);
VAR
  i,j  : CARDINAL;
  e,w  : REAL;
BEGIN
  d[0] := lambda;
  e    := phi[0];
  FOR i:=1 TO n DO
    e:=e-theta[i]*phi[i];
    w:=phi[i];
    FOR j:=i+1 TO n DO w:=w+phi[j]*l[i,j]; END;
    l[0,i]:=0.0;
    l[i,0]:=w;
  END;
  FOR i:=n TO 1 BY -1 DO (* Notice backward loop *)
    DyadicReduction(l[0],l[i],d[0],d[i],0,i,n);
  END;

  FOR i:=1 TO n DO
    theta[i]:=theta[i]+l[0,i]*e;
    d[i]:=d[i]/lambda;
  END;
END LDFilter;
```

Figure 11.16 LD decomposition.

In this code, the type

```
col = ARRAY[0..maxindex] OF REAL;
```

has been introduced. By using the procedure DyadicReduction it is now straightforward to write a procedure that implements Algorithm 11.1. Such a procedure is given in Fig. 11.16. The algorithm performs one step of a recursive least-squares estimation. Starting from the current estimate θ, the covariance represented by its LD decomposition, and the regression vector, the procedure generates updated values of the estimate and its covariance. The data type

```
matr = ARRAY[0..maxindex] OF col;
```

is used in the program. The starting values can be chosen to be $L = I$ and $d = [\beta_0, \beta_0, \ldots, \beta_0]$. This gives $LD = \beta_0 I$.

11.7 INTERACTION OF ESTIMATION AND CONTROL

Parameter estimation and control design were treated as two separate subjects in the previous sections of this chapter. In an adaptive controller there are, of course, strong interactions between estimation and control. Some consequences of this interaction are discussed in this section.

Computational Delay

The updating of the estimated parameters and the design are done at each sampling instant. The timing of computations of the controller was discussed in Section 11.2. We pointed out that it is important to have as short a computational delay as possible. The dual time scale of the adaptive control problem implies that the process parameters are assumed to vary slowly. This means that the parameter estimates from the previous sampling instant can be used for calculating the control signal. There will thus be no extra time delay due to the adaptation, provided that the parameter update and the controller design are done after the control signal is sent out to the process.

Integral Action

Practically all controllers need integral action to ensure that calibration errors and load disturbances do not give steady-state errors. In Section 3.6 we showed how the design procedure could easily be modified to give controllers with integral action. In that section it was also shown that a particular adaptive controllers automatically gave zero steady-state error. This situation occurs quite frequently. It is also easy to check whether a particular self-tuner has this ability by investigating possible stationary solutions. A typical example is the following.

EXAMPLE 11.9 Obtaining integral action automatically

Consider the simple direct moving-average self-tuning controller described in Chapter 4, which is based on least-squares estimation and minimum-variance control. The estimation is based on the model

$$y(t+d) = R^*(q^{-1})u(t) + S^*(q^{-1})y(t)$$

and the regulator is

$$u(t) = -\frac{S^*}{R^*}\, y(t)$$

The conditions for a stationary solution are that

$$\hat{r}_y(\tau) = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} y(k+\tau)y(k) = 0 \qquad \tau = d, \ldots, d+l$$

$$\hat{r}_{yu}(\tau) = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} y(k+\tau)u(k) = 0 \qquad \tau = d, \ldots, d+k$$

where k and l are the degrees of the R^* and S^* polynomials, respectively. These conditions are not satisfied unless the mean value of y is zero. When there is an offset, the parameter estimates will get values such that $R^*(1) = 0$, that is, there is an integrator in the controller. However, the convergence to the integrator may be slow. □

A second way to explicitly eliminate steady-state errors is to base an adaptive controller on estimation of parameters in the model

$$A(q)y(t) = B(q)u(t) + v$$

where v is a constant that is estimated. The control design should also be modified by introducing a feedforward from the estimated disturbance. This approach has the drawback that an extra parameter has to be estimated. Furthermore, it is necessary to have different forgetting factors on the bias estimate and the other estimates; otherwise, the convergence to a new level will be very slow. Finally, if the bias is estimated in this way, it is not possible to use the self-tuner as a tuner, since there will be no reset when the estimation is switched off. This is a simple example that shows the drawbacks of mixing the functions of the feedback loop and the adaptation loop. A much better way is to design a controller with integral action, for example, by using the methods discussed in Section 3.6. A data filter of band-pass character should also be used so that the disturbance v does not influence estimation too much. We will also show how a similar approach can be used for a direct self-tuner.

Compatible Criteria for Identification and Control

So far, we have treated identification and control as two different tasks. The criterion for the identification (least squares) was chosen largely on an ad hoc basis. It is clearly desirable to try to find a criterion for identification that matches the final use of the model. This is in general a very complicated problem. We therefore discuss a simplified case. Consider a process described by the model

$$A(q)y(t) = B(q)u(t) \tag{11.39}$$

where u is the control signal and y is the measured variable. Let the controller be

$$R(q)u(t) = T(q)u_c(t) - S(q)y(t) \tag{11.40}$$

where u_c is the setpoint and $R(q)$, $S(q)$, and $T(q)$ are polynomials. The polynomials $R(q)$ and $S(q)$ satisfy the Diophantine equation

$$A(q)R(q) + B(q)S(q) = A_m(q)A_o(q) \tag{11.41}$$

where the desired closed-loop polynomial is $A_o(q)A_m(q)$. This equation has many solutions. It is customary to choose the simplest one that gives a causal controller, but it is also possible to introduce an auxiliary condition. Integral action is obtained by finding a solution such that $R(1) = 0$. High-frequency roll-off is obtained by requiring that $S(-1) = 0$.

The polynomial $T(q)$ is given by

$$T(q) = t_0 A_o(q) \tag{11.42}$$

where $t_0 = A_m(1)/B(1)$. If $R(1) = 0$, it also follows that $T(1) = S(1)$. Combining Eqs. (11.39) and (11.40), we get

$$\begin{aligned} y(t) &= t_0 \frac{B(q)}{A_m(q)} u_c(t) \\ u(t) &= t_0 \frac{A(q)}{A_m(q)} u_c(t) \end{aligned} \tag{11.43}$$

Polynomials $A_o(q)$ and $A_m(q)$ are typically chosen to give good rejection of disturbances and insensitivity to modeling errors and measurement noise.

It is desirable to formulate the adaptive control problem in such a way that the goals for control and identification are compatible. If this is done, it means that a model is fitted in such a way that it matches the ultimate use of of the model.

Consider the situation in which the goal is to control a plant with transfer function P_0. A controller is designed by using pole placement based on the approximate model whose transfer function is $P = B/A$. To compute the control law, the parameters of the polynomials A and B are estimated by using least squares, and the controller is then determined by the pole placement method. Let u_0 and y_0 denote the inputs and outputs that are obtained in controlling the actual plant, and let u and y denote the corresponding signals when the controller controls the design model. The control performance error can then be defined as

$$e_{cp} = y_0 - y$$

We have the following result.

THEOREM 11.2 Compatibility of identification and control

The control performance error e_{cp} is identical to the least-squares estimation error if identification is performed in closed loop and if the transfer function of the data filter is chosen to be

$$H_f = \frac{R}{A_o A_m} \tag{11.44}$$

Proof: The proof is a straightforward calculation. The output of the true system is given by

$$y_0 = \frac{P_0 T}{R + P_0 S} u_c \tag{11.45}$$

and the control signal is

$$u_0 = \frac{T}{R + P_0 S} u_c \tag{11.46}$$

The corresponding signals for the nominal plant are obtained simply by omitting the index 0 on y_0, u_0, and P_0. The control performance error then becomes

$$\begin{aligned} e_{cp} &= \left(\frac{P_0 T}{R + P_0 S} - \frac{PT}{R + PS} \right) u_c = \frac{RT(P_0 - P)}{(R + P_0 S)(R + PS)} u_c \\ &= \frac{R(P_0 - P)}{R + PS} u_0 = \frac{AR(P_0 - P)}{A_o A_m} u_0 \end{aligned} \tag{11.47}$$

where the first equality follows from Eqs. (11.43), (11.45), and (11.46). The second equality is obtained by a simple algebraic manipulation. The third follows from Eq. (11.46), and the last equality follows from Eq. (11.41). The least-squares estimation error is given by

$$e = H_f(Ay_0 - Bu_0)$$

It follows from Eq. (11.47) that e and e_{cp} are identical if estimation is based on closed-loop data and if the data filter is chosen to be Eq. (11.44). □

Remark 1. Notice that the denominator of the filter (11.44) is given by $A_o A_m$, which are given by the specifications.

Remark 2. Notice that for a controller with integral action the filter (11.44) is a bandpass filter.

Remark 3. Notice that only the numerator of the filter has to be adapted. □

This result gives a rational way of choosing the data filter for a servo problem.

11.8 PROTOTYPE ALGORITHMS

In this section we present some prototype algorithms for adaptive control. Guidelines for the coding of the algorithms are given. The algorithms can easily be expanded to a variety of controllers.

Algorithm Skeleton

All adaptive algorithms discussed in this chapter have the following form:

```
1    Analog_Digital_conversion
2    Compute_control_signal
3    Digital_Analog_conversion
4    If estimate then
5      begin{estimate}
6        Covariance_update
7        Parameter_update
8        If tune then
9          begin{tune}
             th_design:=th_estimated
10           Design_calculations
11         end{tune}
12     end{estimate}
13   Organize_data
14   Compute_as_much_as_possible_of_control_signal
```

Row 1 implements the conversion of the measured output signal, the reference signal, and possible feedforward signal. All the converted signals are supposed to be filtered through appropriate anti-aliasing filters, as discussed in Section 11.2. Row 3 sets the control signal to the process. Rows 14 and 2 contain the calculations of the control signal, which are independent of whether the parameters are estimated or not. Notice the division of the calculations of the control signal to avoid overly long computation times. All calculations that are possible to do in advance are done in Row 14. Only calculations that contain the last measurements are done in Row 2.

Rows 4–13 contain calculations that are specific for an adaptive algorithm. There are two logical variables, `estimate` and `tune`, which control whether the parameters are going to be estimated and whether the controller is going to be redesigned, respectively. The estimation is done in Rows 5–7, and the design calculations are done in Row 10. Row 13 organizes the data such that the algorithm is always ready to start estimation when the operator wishes.

The various adaptive algorithms discussed in this section differ only in the design calculations. The estimator part can be the same for all algorithms. One important part of the algorithms that will not be discussed here is the operator interface. This is usually a significant part of an adaptive control system, but it is very hardware-dependent, so it is difficult to discuss in general terms. We now discuss the calculations in Rows 4–13 in more detail.

Parameter_update

We assume that the estimated model has the form

$$y(t) = \varphi^T(t)\theta$$

where the components in the regression vector φ are lagged and filtered inputs and outputs. The ordering, the number of lags, and so on depend on the specific model; these details are easily sorted out for the chosen model structure. Rows 6 and 13 of the algorithm contain the bookkeeping of the φ vector (i.e., the usual shift of some parts of the vector and supplement of the latest measurements and outputs). This part of the algorithm should also include the data filtering discussed in Sections 11.2, 11.3, and 11.7. For simplicity it is assumed that the estimation and the covariance update are done by using ordinary recursive least squares (Eqs. 11.19). The calculations can be organized as in the listing below, where eps is the residual, `th_estimated` is the parameter vector, `P` is the covariance matrix, `phi` is the data vector, and `lambda` is the forgetting factor.

```
"Compute residual
eps = y - phi'*th_estimated
"Update estimate
w = P*phi
den = lambda + phi'*w
gain = w/den
th_estimated = th_estimated + gain*eps
"Update covariance
P = (P - w*w'/den)/lambda
```

The prime is the transpose, and $*$ is matrix multiplication. This skeleton can easily be transferred to any preferred programming language.

Organize_data

This part of the code filters the process input and output by H_f, and it updates the regression vector $\varphi(t)$ and the other states of the system. If $\varphi(t)$ is updated at each sampling period, it is possible to update the estimates irregularly.

Design_calculations

When a direct algorithm such as Algorithm 3.3 is used, the controller parameters are the same as the estimated parameters, and there are no calculations that have to be done in the design block. In the indirect methods a polynomial equation has to be solved. The solution of the Diophantine equation is discussed in Section 11.4. Some care must be taken because of difficulties with possible common factors in the estimated model polynomials.

Compute_control_signal

The computation of the control signal to minimize the computational delay was discussed in Section 11.2, along with the anti-reset windup.

Summary

The program skeleton in this section can now be supplemented with details to become a complete adaptive control algorithm. These details will depend on what algorithm is chosen to be implemented and on which programming language is chosen.

11.9 OPERATIONAL ISSUES

Simple controllers typically have two operating modes, manual and automatic. It is also possible to change the parameters during operation. It is a nontrivial task to deal with the operation of a conventional controller. The current practice has developed over a long period of time. Adaptive controllers can operate in many more ways. It is a difficult problem to find a good solution to the operational problems. The problem will also vary widely with the application area. In this section we discuss a controller for industrial process control, which is designed to operate in many widely different environments.

Operating Modes

An adaptive controller has at least three operating modes: manual, constant-parameter control, and adaptation. The controllers that are used in constant-parameter mode may be of several types: PID, relay, or a general linear controller. In this mode the controller parameters must also be loaded and stored. Parameter estimation may also be initiated in the constant-parameter mode. Estimation can also be enhanced by introducing extra perturbations. The nature of these perturbations must also be specified.

Initialization

There are several ways to initialize a self-tuning algorithm, depending on the available *a priori* information about the process. In one case, nothing is known about the process. The initial values of the parameters in the estimator can then be chosen to be zero or such that the initial controller is a proportional or integral controller with low gain. Auto-tuning, discussed in Chapter 8, is a convenient way to initialize the algorithm, because it generates a suitable input signal and safe initial values of the parameters. This also gives a rational way of choosing the sampling interval.

The inputs and outputs of the process should be scaled so that they are of the same magnitude. This will improve the numerical conditions in the estimation and the control parts of the algorithm. The initial value of the covariance matrix can be 1–100 times a unit matrix if the elements in the φ vector are scaled to approximately unity. These values are usually not crucial,

since the estimator will get reasonable values in a very short period of time. Our experience is that 10–50 samples are sufficient to get a very good controller when the system is excited. During the initial phase it can be advantageous to add a perturbation signal to speed up the convergence of the estimator.

The situation is different if the process has been controlled before with a conventional or an adaptive controller. The initial values should then be such that they correspond to the controller used before. Furthermore, the P-matrix should be sufficiently small.

Sometimes it is important to have disturbances be as small as possible, owing to the startup of the self-tuning algorithm. There are then two precautions that can be taken. First, the estimator can be used for some sampling periods before the self-tuning algorithm is allowed to put out any control actions. During that time a safe, simple controller should be used. It is also possible and desirable to limit the control signal. The allowable magnitude can be very small during the first period of time and can then be increased when better parameter estimates are obtained. The drawback of having small input signals is that the excitation of the process will be poor, and it will take longer to get good parameter estimates.

Supervision

An adaptive controller should also contain facilities for supervision. Basic statistics such as the mean, standard deviation, and maximum and minimum values should be computed for the process input and output. These values should be averaged over the basic period of the loop. Since excitation is so important, it should be monitored. The estimation error also gives useful information about the behavior of the loop. Common factors in the process model should be detected. This will indicate that the model structure should be changed. For special algorithms it is also possible to determine whether the controller behaves as expected. For example, for minimum variance or moving average controllers this can be determined by monitoring the covariance of the process output.

11.10 CONCLUSIONS

Practical aspects on implementation of adaptive controllers have been discussed in this chapter. There are many things to consider, since adaptive controllers are quite complicated devices. The following are some of the important issues:

- Analog anti-aliasing filters must be used. They are typically second- or fourth-order filters that effectively eliminate signal components with frequencies above the Nyquist frequency π/h, where h is the sampling period.

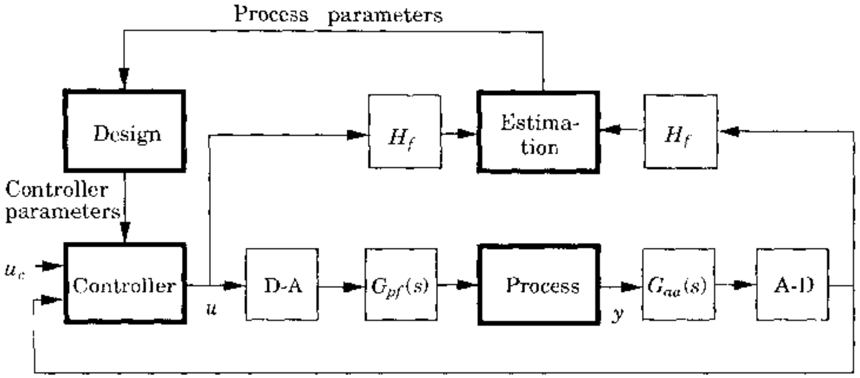

Figure 11.17 Block diagram of an adaptive control system with added filters. G_{aa} is the anti-aliasing filter, G_{pf} is the postsampling filter, and H_f is the data filter for the estimation.

The dynamics of the filters should be taken into account in the control design.

- Inputs and outputs should be filtered by a bandpass filter with H_f before these signals are sent to the parameter estimator. These filters will remove low-frequency disturbances such as levels and ramps. High frequency disturbances are also removed by H_f. The lower limit of the passband should be at least one decade below the desired crossover frequency. Known sinusoidals can also be removed by using notch filters.
- The postsampling filter G_{pf} is used to avoid excitation of high-frequency resonance modes in the process.
- Low-order models are typically used. They are estimated with algorithms having time variable exponential forgetting, regularized constant trace, or directional forgetting. The estimator should also contain a dead zone. Finally, the estimator may contain a "switch," which detects whether the system is sufficiently excited. The "switch" can measure the power in different frequency bands and thus control whether the estimator should be active or not. Square root algorithms are preferable, particularly if there is a high signal to noise ratio.
- The design method for the controller should be robust against unmodeled dynamics. Level and ramp disturbances are eliminated by introducing integrators in the controller. The control signal should be limited, and the controller should include anti-reset windup.

A block diagram of a reasonably realistic adaptive controller is given in Fig. 11.17. Adaptive controllers also contain parameters. Guidelines for choosing these have been given in this chapter.

PROBLEMS

11.1 How should the disturbance annihilation filter $H_f(q)$ be chosen if $v(t)$ in Eq. (11.17) is a sinusoidal?

11.2 Consider the data filter

$$H_f(q) = \frac{(1-\alpha)(1-q)}{q-\alpha}$$

Discuss how the choice of the parameter α influences elimination of constant disturbances.

11.3 Plot the Bode diagram for a fourth-order Bessel filter, and compare it with a pure time delay. Consider the cases in Table 11.2.

11.4 Determine how the behavior of the anti-reset windup controller of Eqs. (11.1) is influenced by the filter A_o.

11.5 Complete the algorithm skeleton for the cases of

(a) a direct self-tuner (Algorithm 3.3).

(b) an indirect self-tuner without zero cancellation (Algorithm 3.2).

11.6 Perform a transformation of a second- and fourth-order Bessel filter, $\omega_B = 1$, into band-pass filters, using the transformation

$$s \to \frac{s^2 + \omega_l\omega_h}{s(\omega_h - \omega_l)}$$

where ω_l and ω_h are the lower and upper cutoff frequencies, respectively. Use $\omega_l = 100$ and $\omega_h = 1000$ rad/s. Compare the band-pass characteristics by using Bode diagrams.

11.7 Use Euclid's algorithm to compute the greatest common divisor of

$$A = q^3 - 2q^2 + 1.45q - 0.35$$
$$B = q^2 - 1.1q + 0.3$$

Also determine the polynomials X and Y in Eq. (11.8).

11.8 Consider the Diophantine equation

$$AR + BS = A_c$$

and let

$$A(q) = (q-1)(q-0.9)$$

Use the method in Section 11.4, and compute the resulting controller when (a) $B(q) = q - 0.6$; (b) $B(q) = q - 0.9$. Assume that the desired closed characteristic polynomial is

$$A_m(q) = q^2 - q + 0.7$$

11.9 One way to solve the Diophantine equation is to multiply it by a persistently exciting signal, such as white noise. Introduce the filtered signals

$$v_a(t) = \frac{\hat{A}}{A_m A_o} v(t) \qquad v_b(t) = \frac{\hat{B}}{A_m A_o} v(t)$$

The Diophantine equation then becomes

$$Rv_a + Sv_b = v$$

The coefficients of the R and S polynomials can now be determined by using the method of least squares, and one iteration can be done at each sampling instance. Discuss the merits and drawbacks of this approach. (*Hint*: What is the convergence rate?)

REFERENCES

Implementation issues for adaptive controllers are discussed in:

Wittenmark, B., and K. J. Åström, 1980. "Simple self-tuning controllers." In *Methods and Applications in Adaptive Control*, ed. H. Unbehauen, pp. 21–30. Berlin: Springer-Verlag.

Åström, K. J., 1983. "Analysis of Rohrs' counter example to adaptive control." *Preprints of the 22nd IEEE Conference on Decision and Control*, pp. 982–987. San Antonio, Tex.

Wittenmark, B., and K. J. Åström, 1984. "Practical issues in the implementation of self-tuning control." *Automatica* **20**: 595–605.

Clarke, D. W., 1985. "Implementation of self-tuning controllers." In *Self-tuning and Adaptive Control: Theory and Applications*, eds. C. J. Harris and S. A. Billings. London: Peter Peregrinus.

Isermann, R., and K.-H. Lachmann, 1985. "Parameter adaptive control with configuration aids and supervision functions." *Automatica* **21**: 623–638.

Middleton, R. H., G. C. Goodwin, D. J. Hill, and D. Q. Mayne, 1988. "Design issues in adaptive control." *IEEE Trans. Automat. Contr.* **AC-33**: 50–58.

Wittenmark, B., 1988. "Adaptive control: Implementation and application issues." In *Adaptive Control Strategies for Industrial Use*, eds. S. L. Shah and G. Dumont, pp. 103–120, Proceedings of a Workshop, Kananaskis, Canada. New York: Springer-Verlag.

Different design methods and their properties are treated in:

Lennartson, B., and T. Söderström, 1986. "An investigation of the intersample variance for linear stochastic control." *Preprints of the 25th IEEE Conference on Decision and Control*, pp. 1770–1775. Athens.

Åström, K. J., and B. Wittenmark, 1990. *Computer Controlled Systems*, 2nd ed. Englewood Cliffs, N.J.: Prentice-Hall.

Aspects on implementation of estimation routines are found in:

Bierman, G. J., 1977. *Factorization Methods for Discrete Sequential Estimation.* New York: Academic Press.

Ljung, L., and T. Söderström, 1983. *Theory and Practice of Recursive Identification.* Cambridge, Mass.: MIT Press.

Goodwin, G. C., and K. S. Sin, 1984. *Adaptive Filtering, Prediction and Control.* Englewood Cliffs, N.J.: Prentice Hall.

Hägglund, T., 1985. "Recursive estimation of slowly time varying parameters." *Preprints of the 7th IFAC Symposium on Identification and System Parameter Estimation*, pp. 1137–1142. York, U.K.

Kulhavý, R., 1987. "Restricted exponential forgetting in real-time identification." *Automatica* **23**: 589–600.

Ljung, L., and S. Gunnarsson, 1990. "Adaptation and tracking in system identification: A survey." *Automatica* **26**: 7–21.

Different ways to introduce integrators in adaptive controllers are treated in:

Wellstead, P. E., and P. Zanker, 1982. "Techniques for self-tuning." *Optimal Control Applications & Methods* **3**: 305–322.

Design of low-pass and band-pass filters can be studied in:

Rabiner, L. R., and B. Gold, 1975. *Theory and Application of Digital Signal Processing.* Englewood Cliffs, N.J.: Prentice-Hall.

The dyadic decomposition method described in Section 11.6 is based on:

Gentleman, W. M., 1973. "Least squares computations by Givens transformations without square roots." *J. Inst. Math. Appl.* **12**: 329–336.

Peterka, V., 1987. "Algorithms for LQG self-tuning control based on input-output delta models." In *Adaptive Systems in Control and Signal Processing 1986*, eds. K. J. Åström and B. Wittenmark, IFAC Proceedings. Oxford, U.K.: Pergamon Press.

The use of the Diophantine equation in control is surveyed in:

Kučera, V., 1993. "Diophantine equations in control: A survey." *Automatica* **29**: 1361–1375.

CHAPTER 12

COMMERCIAL PRODUCTS AND APPLICATIONS

12.1 INTRODUCTION

There have been a large number of applications of adaptive feedback control over the past 30 years. Experiments with adaptive flight-control systems were done in 1960. Industrial experiments with self-tuning regulators were performed in 1972. Full-scale experiments with adaptive autopilots for ship steering were done in 1973. Special adaptive systems have been in continuous use for a long time. Some process control loops have been running continuously since 1974. There are also a number of special products that have been operating for a long time. Commercial systems for ship steering have been in continuous operation since 1980.

Systems implemented by using minicomputers appeared in the early 1970s. However, not until the 1980s did adaptive techniques start to have real impact on industry. The number of applications increased drastically with the advent of the microprocessor, which made the technology cost-effective. Because of this, adaptive controllers are also entering the marketplace even in single-loop controllers. Several commercial products based on adaptive techniques were introduced in the early 1980s, and second- and third-generation versions have been introduced in some cases.

Adaptive techniques are used in a number of products. Gain scheduling is the standard method for design of flight control systems for high-performance aircraft, and it is also used in robotics and process control. The self-oscillating adaptive system is used in several missiles. There are several commercial adaptive systems for ship steering, motor drives, and industrial robots. Adaptive techniques are used both in single-loop controllers and in general-purpose process control systems in the process industry. Most industrial processes are controlled by PID controllers, and a large industrial plant may have thousands of them. Many instrument engineers and plant personnel are used to select, install, and operate such controllers. In spite of this, many controllers are poorly tuned. One reason is that simple, robust methods for automatic tuning have not been available. Adaptive methods are now available for automatic tuning of PID controllers. This is in fact one of the fastest-growing areas of application for adaptive control.

However, adaptive techniques are still not widely used; the technology is not mature. Because of involvement of commercial enterprises in adaptive control, it is not always possible to find out precisely what is being done. Various ideas are hidden in proprietary information that is carefully guarded.

This chapter is organized as follows. An overview of some applications is given in Section 12.2. A number of commercial products that use adaptation are presented in Sections 12.3 and 12.4. Some specific applications are presented in more detail in the sections that follow. Ship steering, automobiles, and ultrafiltration are areas given special attention.

12.2 STATUS OF APPLICATIONS

A large number of experiments with adaptive control have been performed since the mid-1950s. The experiments have had different purposes: to verify ideas, to find out how adaptive systems perform, to compare different approaches, and to find out when they are suitable. The early experiments, which used analog implementations, were plagued by hardware problems. When digital process computers became available, they were natural tools for experimentation. Experiments with adaptive control required substantial programming, since adaptation was not part of the standard software. Applications proliferated with the advent of the microprocessor, which is a convenient tool for implementing adaptive systems. Adaptive techniques now appear both in single-loop controllers and as standard elements of large process control systems. There are tailor-made controllers for special purposes that use adaptive techniques.

Feasibility Studies

A number of feasibility studies have been performed to evaluate the usefulness of adaptive control. They cover a wide range of control problems, such as autopilots for missiles, ships, and aircraft; engine control; motion control; machine tools; industrial robots; power systems; distillation columns; chemical reactors; pH control; furnaces; heating; and ventilation. There are also applications in the biomedical area. The feasibility studies have shown that there are cases in which adaptive control is very useful and others in which the benefits are marginal. Some industrial products also use adaptive techniques. There are both general-purpose controllers and controllers for special applications.

Auto-tuning

Simple controllers with two or three parameters can be tuned manually if there is not too much interaction between the adjustments of different parameters, but manual tuning is not possible for more complex controllers. Traditionally, tuning of complex controllers has taken the route of modeling or identification and controller design. This is often a time-consuming and costly procedure, which can be applied only to important loops or to systems that are to be manufactured in large quantities.

All adaptive techniques can be used to provide automatic tuning. In such applications the adaptation loop is simply switched on. Perturbation signals may be added to improve the parameter estimation. The adaptive controller is run until the performance is satisfactory; then the adaptation loop is disconnected, and the system is left running with fixed controller parameters. The particular methods for automatic tuning of PID controllers that were discussed in Chapter 8 have been found to be particularly attractive because they require little prior information and are closely related to standard industrial practice.

Auto-tuning can be considered a convenient way to incorporate automatic modeling and design in a controller. It simplifies the use of the controller, and it widens the class of problems in which systematic design methods can be used cost-effectively. This is particularly useful for design methods such as feedforward that depend critically on good models.

Automatic tuning can be applied to simple PID controllers as well as to more complicated systems. It is very convenient to introduce tuning into a DDC package because the tuning algorithm can serve many loops. Auto-tuning can also be included in single-loop controllers. For example, it is possible to obtain standard controllers in which the mode switch has three positions: manual, automatic, and tuning. A well-designed auto-tuner is very easy to use, even for unskilled personnel. Experience has shown it to be useful both for commissioning of new systems and for routine maintenance. Auto-tuners can also be used to enhance the skill of the instrument engineers. Automatic tuning will probably also be a useful feature of more complicated controllers.

Automatic Construction of Gain Schedules

Gain scheduling is a very useful technique, but it has the drawback that it may be quite time- and cost-consuming to build a schedule. Auto-tuning can conveniently be used to build gain schedules. A scheduling variable is first determined. The parameters that are obtained when the system is running in one operating condition are then stored in a table together with the scheduling

variable. The gain schedule is obtained when the process has operated at a variety of operating conditions that covers the operating range.

True Adaptive Control

The adaptive techniques may, of course, also be used for genuine adaptive control of systems with time-varying parameters. There are many ways to do this. The operator interface is important, since adaptive controllers also have parameters that must be chosen. Controllers without any externally adjusted parameters can be designed for specific applications, in which the purpose of control can be stated *a priori*. The ship steering autopilot discussed in Section 12.6 is a typical example. In many cases, however, it is not possible to specify the purpose of control *a priori*. It is at least necessary to tell the controller what it is expected to do. This can be done by introducing dials that give the desired properties of the closed-loop system. Such dials are characterized as *performance-related*. New types of controllers can be designed by using this concept. For example, it is possible to have a controller with one dial, labeled with the desired closed-loop bandwidth. Another possibility would be to have a controller with a dial that is labeled with the weighting between state deviation and control action in an LQG problem. Adaptation can also be combined with gain scheduling. A gain schedule can be used to get the parameters quickly into the correct region, and adaptation can then be used for fine-tuning.

Adaptive Feedforward

In many applications it is possible to measure some of the disturbances acting on the process. Feedforward control is very useful when there are measurable disturbances. With feedforward it is possible to decrease the influence of disturbances substantially. However, feedforward control, being an open-loop compensation, requires good models of process dynamics. Identification and adaptation therefore appear to be prerequisites for effective use of feedforward compensation. Until now, very little research and development have been done on adaptive feedforward, even if it was used in the early applications of self-tuning regulators.

Abuses of Adaptive Control

An adaptive controller is more complex than a fixed-gain controller, since it is nonlinear. Before we attempt to use an adaptive controller, it may therefore be useful to investigate whether the problem can be solved with a robust constant-gain controller, as discussed in Chapter 10. As was pointed out in Chapter 1, it is not possible to judge the need for adaptation from the variations in the open-loop dynamics. The open-loop responses may vary much while the closed-loop responses are close and vice versa.

The complexity of the controller has to be balanced against the engineering effort required to make the system operational. Experience has shown that only a modest effort is required to make a standard adaptive system work well.

12.3 INDUSTRIAL ADAPTIVE CONTROLLERS

A number of industrial products incorporate adaptive control techniques. The products can be divided into

- Tuning tools for standard controllers,
- Adaptive standard process controllers,
- General-purpose toolboxes for adaptive control, and
- Special-purpose adaptive controllers.

Because of the large number of different products, it is possible to give only some examples from the different categories.

Tuners for Standard Process Controllers

There are many products for tuning of standard controllers of PID type. Leeds and Northrup announced a PID controller with a self-tuning option in 1981. SattControl in Sweden announced auto-tuning for PID controllers in a small DDC system in 1984 and a single-loop controller with auto-tuning in 1986. Practically all PID controllers that come on the market today have some kind of built-in automatic tuning or adaptation. There are four main solutions for the tuners for standard controllers:

- A parametric model approach,
- A nonparametric model approach,
- External tuning devices, and
- Tuning tools in distributed control systems.

The main idea in the parametric model controllers is to make an experiment, usually in open loop, and estimate a first- or second-order model with time delay. The input signals are usually steps, but pulses or pseudo-random binary sequence (PRBS) signals are also used. The parameters of a PI or PID controller are then determined by using empirical tuning rules or a pole placement technique. Typical products in this category are Protonic from Hartman & Braun and UDC 6000 from Honeywell.

In the nonparametric model approach, a point on the Nyquist curve is generally estimated by using relay feedback. Compare the auto-tuning discussed in Chapter 8. On the basis of this information a modified set of Ziegler-Nichols

tuning rules are used to determine the parameters of the controller. SattControl ECA40 and Fisher-Rosemount DPR900 are typical of this category.

The tuning aids discussed above are built-in features in the standard controllers. The operator initiates tuning by pushing a button or giving a command. The external tuning tools are special types of equipment that are connected to the process for the tuning or commissioning and then removed. The experiments are usually done with the process in open loop. The external tuner then determines suitable controller parameters. The new parameters are often entered manually by the operator. Since the external tuner can be used for different types of standard controllers, it must have detailed knowledge about the parameterization and implementation of algorithms from different manufacturers. Examples of external tuning tools are Supertuner from Toyo Systems in Japan, Protuner from Techmation in Arizona, PIDWIZ from BST Control in Illinois, and SIEPID from Siemens in Germany.

Tuning tools have also been introduced in distributed control systems. Because of the available computing power, it is possible to have very good human-machine interfaces and several options for tuning. Honeywell has a system called Looptune; Fisher-Rosemount Systems has a product called Intelligent Tuner.

Adaptive Standard Process Controllers

The tuners discussed above do not tune the controllers continuously but only on demand from the operator. However, there are also standard controllers with adaptation, which can follow changes in the parameters of the process. The adaptive standard controllers can be divided into

- A parametric model approach,
- A nonparametric model approach, and
- A pattern recognition approach.

The model-based adaptive controller usually estimates a first- or second-order model with time delay using a recursive least-squares algorithm. A pole placement controller with PID structure can then be determined. Examples are the Bailey Controls CLC04 and Yokogawa SLPC-181, -281.

One example of a nonparametric adaptive controller is SattControl ECA 400. (See Fig. 1.23.) It is a development of the relay-based auto-tuner. One point of the Nyquist curve is estimated continuously by using band-pass filtering. The parameters of the controller are then determined by using a modified version of the Ziegler-Nichols tuning rules.

Expert systems or pattern recognition have also been used for adaptive tuning of standard controllers. The first was the Foxboro EXACT, which was announced in October 1984. This controller is described in more detail in the text that follows. In 1987, Yokogawa announced adaptive PID controllers, SLPC-171 and SLPC-271, which have features similar to those of Foxboro's

EXACT. Another controller in this category is Fenwal 570. The Honeywell UDC 6000 controller uses step response analysis for automatic tuning and a rule base for adaptation. These controllers are designed to capture the skill of an experienced control engineer in rules. About 100–200 rules are typically implemented. The controllers are waiting for changes in the reference value or large upsets of the process. On the basis of the response and the tuning rules, the parameters of the controller are modified to increase the performance of the closed-loop system.

Several of the adaptive standard controllers, for example, Fisher DPR 910 and SattControl ECA400, have adaptive feedforward and the possibility to build up gain scheduling tables automatically. These features are very useful and can improve the performance considerably.

Standard controllers with more sophisticated control algorithms are now appearing on the market. One example is U.A.C. (Universal Adaptive Controller) from Process Automation Systems in British Columbia, which is based on predictive control. The controller can also handle multivariable systems.

General-Purpose Toolboxes for Adaptive Control

There is often a need to use more elaborate control algorithms than the standard PID controllers. It is then necessary to estimate higher-order models and to have the possibility to use different design algorithms. To cover these situations, general toolboxes for adaptive control have been developed. The adaptive algorithms are usually modules or blocks in more general packages for direct digital control (DDC). Asea Brown Boveri presented a general-purpose adaptive controller in 1982. First Control Systems in Sweden introduced an adaptive controller in 1986. It is also possible to implement adaptive control in modern distributed control systems.

PLC Implementations

Adaptive controllers can also be implemented in ordinary programmable logic controller (PLC) systems. Such solutions are used by manufacturing companies with competent in-house expertise. For example, 3M has implemented adaptive controllers in this way. The first installation was made in 1987. Currently, there are about 200 adaptive loops in operation. A wide range of processes are controlled. The systems are implemented on a variety of platforms such as General Electric, Modicon, Measurex, Square-D, and Reliance. Programming is done in Basic or C. The applications include standard loops for temperature, pressure, position, and humidity and more specialized loops associated with 3M proprietary processes. The adaptive algorithms that are used are based on estimation of parameters in models having the structure

$$A^*(q^{-1})y(t) = B_1^*(q^{-1})u(t-d) + B_2^*(q^{-1})v(t-d)$$

where v is a measurable disturbance. Polynomial A^* has degree one or two, but polynomials B_1^* and B_2^* may have higher degree to cope with variable time delay. The parameters are estimated by a special gradient technique. The control design is a modified minimum-variance strategy.

Special-Purpose Adaptive Controllers

For many processes, extensive process knowledge is available. To make good control, it is advantageous to use as much *a priori* knowledge as possible. Structures of the model and knowledge of integrators or time constants can be used to design the controller and to facilitate the tuning. For instance, special-purpose adaptive controllers have been developed for ships, pulp digesters, motor drives, ultrafiltration, and cement raw material mixing.

12.4 SOME INDUSTRIAL ADAPTIVE CONTROLLERS

Some representative commercial products and their features are described in this section. Special emphasis is put on properties such as estimation, prior information, and industrial experiences. The section ends with a discussion of some general aspects of industrial use of adaptive controllers.

SattControl ECA40 and Fisher Control DPR 900

This is the original auto-tuner based on relay oscillations, as described in Chapter 8. It was first introduced in a small (about 45 loops) DDC system for industrial process control SDM20. In this application the tuner can be connected to tune any loop in the system. Relay auto-tuning is also available in single-loop PID controllers (SattControl ECA40 and Fisher Control DPR900). In these controllers, tuning is done on demand by pushing a button on the front panel, so-called *one-button tuning*. The controllers are also provided with facilities for gain scheduling. There is a table with three controller settings.

Parameter Estimation. The ultimate period and the ultimate gain are determined by an experiment with relay feedback. The fluctuations in the output signal are measured, and the hysteresis of the relay is set slightly wider than the noise band. The initial relay amplitude is fixed. The amplitude and period are measured for each half-period. A feedback adjusts the relay amplitude so that the limit cycle oscillation has a given amplitude. When two successive half-periods are sufficiently close, PID parameters are computed, and PID control is initiated automatically.

Control Design. When the ultimate gain and the ultimate period are known, the parameters of a PID controller can be determined by a modified Ziegler-

Nichols rule. There is also a limited amount of logic to determine whether derivative action is needed.

Prior Information. A major advantage of the auto-tuner is that no parameters have to be set *a priori*. To use the tuner, the process is simply brought to an equilibrium by setting a constant control signal in manual mode. The tuning is then activated by pushing the tuning button. The controller is automatically switched to automatic mode when the tuning is complete. Different control objectives may be obtained by modifying the parameters in the Ziegler-Nichols rule. One mode is chosen by default, but the user can request a slower or an extra-fast response.

Industrial Experiences. The system has been considered very easy to use, even by inexperienced personnel. Both the auto-tuning and gain-scheduling features have been found to be very useful. In many applications the auto-tuner has contributed significantly to improved tuning. It has also been demonstrated that commissioning time can be shortened significantly by using automatic tuning and that the standard controller can be applied to processes having a wide range of time scales. Simplicity is the major advantage of the auto-tuner. This has proved particularly useful for plants that do not have qualified instrument engineers and for operation during the night shift, when instrument engineers are not available. It is also easy to explain the auto-tuner to the instrument engineers. The properties of the auto-tuner are illustrated by an example.

EXAMPLE 12.1 Level control

Figure 12.1 shows the behavior of the controller when it is used to control the level of a vessel in a pulp mill. A controller with pure proportional action was

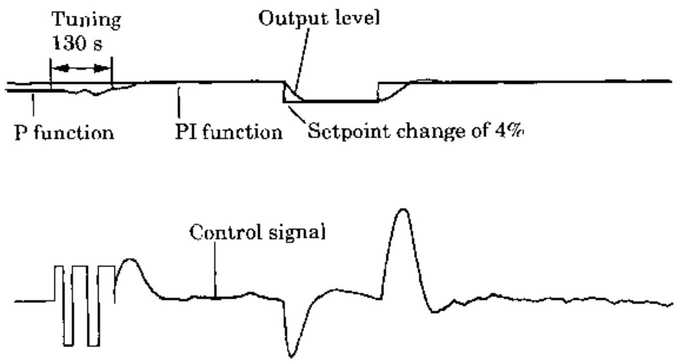

Figure 12.1 Results obtained when using the SattControl ECA40 for level control in a pulp mill.

used originally, resulting in the steady-state error shown in the figure. The tuning took about two minutes and resulted in a PI controller. This example illustrates the usefulness of the logic for selecting control action. Figure 12.1 also shows the control signal. □

EXACT: The Foxboro Adaptive Controller

This controller is based on analysis of the transient response of the closed-loop system to setpoint changes or load disturbances and traditional tuning methods of the Ziegler-Nichols type.

Parameter Estimation. Assuming controller parameters such that the closed-loop system is stable, a typical response of the control error to a step or impulse disturbance is shown in Fig. 12.2. Heuristic logic is used to detect that a proper disturbance has occurred and to detect the peaks e_1, e_2, and e_3 and period T_p. The estimation process is simple, but it is based on the assumption that the disturbances are steps or short pulses. The algorithm can give wrong estimates if the disturbances are two short pulses because T_p will then be estimated to be the distance between the pulses.

Control Design. The control design is based on specifications on damping, overshoot, and the ratios T_i/T_p and T_d/T_p, where T_i is the integration time, T_d is the derivative time, and T_p is the period of oscillation. The damping is defined as

$$d = \frac{e_3 - e_2}{e_1 - e_2}$$

and the overshoot as

$$o = -\frac{e_2}{e_1}$$

In typical cases, both d and o must be less than 0.3. Empirical rules are used to calculate the controller parameters from T_p, d, and o. These rules are based on

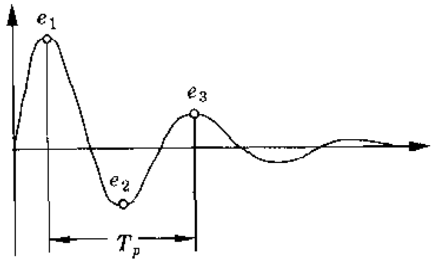

Figure 12.2 Typical response of control error to step or impulse disturbances.

traditional tuning rules of the Ziegler-Nichols type, augmented by experiences from controller tuning.

Prior Information. The tuning procedure requires prior information about the controller parameters K_c, T_i, and T_d. It also requires information on the time scale of the process. This is used to determine the maximum time the heuristic logic waits for the second peak. Some measure of the process noise is also needed to set the tolerances in the heuristic logic. Some parameters may also be set optionally: damping d, overshoot o, maximum derivative gain, and bounds on the controller parameters.

Pre-tuning. The tuning procedure requires reasonable controller parameters to be known so that a closed-loop system with a well-damped response is obtained. There is a pre-tune mode that can be used if the prior information that is needed is not available. A step test is done in which the user specifies the step size. Initial estimates of the controller parameters are determined from the step, and the time scale and the noise level are also determined. The pre-tune mode can be invoked only when the process is in steady state.

Industrial Experiences. Thousands of units of EXACT controllers are in use today. The system is also available in Foxboro's system for distributed process control. Users from a large number of installations have reported favorably, citing the ease with which controllers can be well tuned and the ability to shorten commissioning time. It is also mentioned that derivative action can often yield significant benefits.

Eurotherm Temperature Controller

Temperature control is traditionally done with simple PID controllers, which are cheaper than conventional industrial controllers. Auto-tuning is now also used in such simple systems. One example is controllers produced by Eurotherm in the United Kingdom. A modified relay tuning is used in those controllers. Full control power is used until an artificial setpoint is reached. Two half-periods of a relay tuning are then used, and the controller parameters are calculated from the transient. The controller also has facilities for automatic on-line tuning based on transient response analysis.

In temperature control loops, there are usually different dynamics depending on whether the temperature is increasing or decreasing. This nonlinearity can be handled by using gain scheduling.

Asea Brown Boveri (ABB) Adaptive Controller

The Asea Brown Boveri (ABB) adaptive controller was first marketed under the name Novatune. It is an adaptive controller that is incorporated as a part of ABB Master, a distributed system for process control. The system is

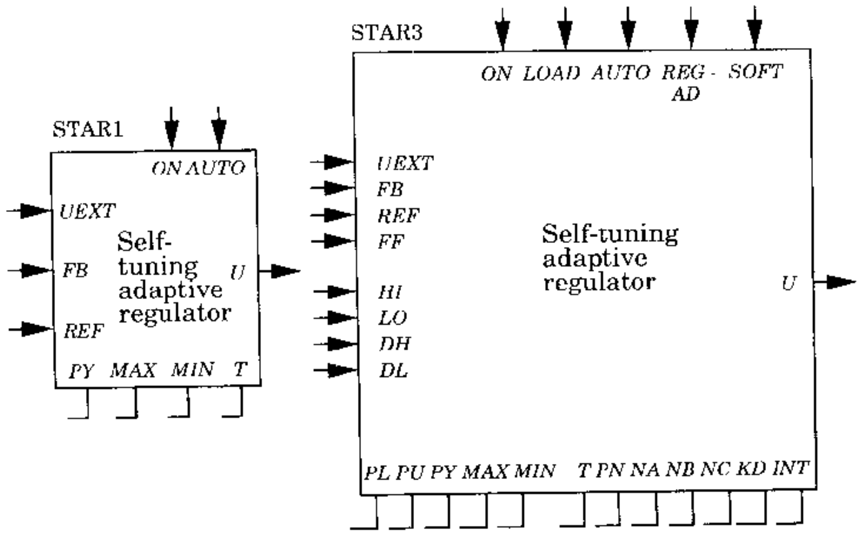

Figure 12.3 Block diagrams of the adaptive modules STAR1 and STAR3, available in the ABB adaptive controller.

block-oriented, which means that the process engineer creates a system by selecting and interconnecting blocks of different types. The system has blocks for conventional PID control, logic, and computation. Three different blocks, called STAR1, STAR2, and STAR3, are adaptive controllers. The adaptive controllers are self-tuning regulators based on least-squares estimation and minimum-variance control. The controllers all use the same algorithm; they differ in the controller complexity and the prior information that must be supplied in using them.

The ABB adaptive controller differs from the controllers that were discussed previously in that it is not based on the PID structure. Instead, its algorithm is based on a general pulse transfer function. It also admits dead-time compensation and feedforward control. The ABB adaptive controller system may be viewed as a toolbox for solving control problems.

Principle. The ABB adaptive controller is a direct self-tuning regulator similar to Algorithm 4.1 in Section 4.3. The parameters of a discrete-time model are estimated by using recursive least squares. The control design is a minimum-variance controller, which is extended to admit positioning of one pole and a penalty on the control signal. The block diagrams in Fig. 12.3 show two of the adaptive modules. The ABB adaptive controller system has three adaptive modules: STAR1, STAR2, and STAR3. STAR3 is the most complicated. The simpler ones have fewer inputs and have default values on some of the parameters in STAR3. In the block diagram the input signals are shown on the left and top sides of the box, the output signals on the right, and the parameters on the bottom. The parameters can be changed at configuration time. The parameters *PL*, *T*, and *PN* can also be changed on-line.

The simplest module, STAR1, has three input signals: the manual input *UEXT*, the measured value *FB*, and the setpoint *REF*. It has three parameters. The variable *PY* is the smallest relevant change in the feedback signal; the adaptation is inhibited for changes less than *PY*. The parameters *MAX* and *MIN* denote the bounds on the control variable, and *T* is the sampling period.

The module STAR2 has more input signals. It admits a feedforward signal *FF*. There are also four signals, *HI*, *LO*, *DH*, and *DL*, that admit dynamic changes on the bounds of the control variable and its rate of change. There are also additional parameters: *PN*, for a penalty on the control variable, and *KD*, which specifies the prediction horizon. The module also has two additional mode switches: *REGAD*, which turns off adaptation when false, and *SOFT*, which allows a soft start.

The module STAR3 has an additional function *LOAD*, which admits parameters stored in an EEPROM to be loaded. It also has several additional parameters, which admit positioning of one pole *PL* and specification of controller structure *NA*, *NB*, *NC*, and *INT*.

Parameter Estimation. The parameter estimation is based on the model

$$\begin{aligned}(1 - PLq^{-1})y(t + KD) &- (1 - PL)y(t)\\ &= A^*(q^{-1})\Delta y(t) + B^*(q^{-1})\Delta u(t) + C^*(q^{-1})\Delta v(t)\end{aligned}$$

where A^*, B^*, and C^* are polynomials in the delay operator q^{-1}, y is the measured variable, u is the control signal, v is a feedforward signal, and Δ is the difference operator $1-q^{-1}$. (Compare with Algorithm 3.6.) The integers *NA*, *NB*, and *NC* give the number of coefficients in the polynomials A^*, B^*, and C^*, respectively. The number *PL* is the desired pole location for the optional pole. When parameter *INT* is zero, a similar model without differences is used. The parameters are estimated by using recursive least squares with a forgetting factor $\lambda = 0.98$. Parameter estimation is suspended automatically when the changes in the control signal and the process output are less than *PU* and *PY*. The parameter updating may also be suspended on demand through the switch *REGAD*. In combination with other modules in the ABB adaptive controller system, this constitutes a convenient way to obtain robust estimation.

Control Design. The control law is given by

$$(\rho + B(q^{-1}))\,\Delta u(t) = (1 - PL)(u_c(t) - y(t)) - A^*(q^{-1})\Delta y(t) - C^*(q^{-1})\Delta v(t)$$

where ρ is a penalty factor related to *PN*. Since the algorithm is a direct self-tuner, the controller parameters are obtained directly from the estimated parameters.

Industrial Experiences. The ABB adaptive controller has been applied to a wide range of process control problems in the steel, pulp, paper, and petrochemical industries, wastewater treatment, and climate control. Some applications have given spectacular improvement of performance compared to PID control. This is particularly the case for processes with time delay, and in applications in which adaptive feedforward can be used. It has also been used to make special-purpose systems for special application areas such as paper winding and climate control. Some ABB adaptive controller applications are described in more detail in Section 12.5. The essential drawback of the ABB adaptive controller is that it is based on a direct self-tuner. This means that the sampling period and the parameter *KD* have to be chosen with care. It may, for example, be difficult to use very short sampling periods.

Firstloop: The First Control Adaptive Controller

The adaptive system Firstloop was developed by First Control Systems, a small company founded by members of the Novatune team. Firstloop is a small controller module with up to eight self-tuning regulators. The system is a toolbox with modules for adaptive control, logic, filtering square root functions, and operator communication. An interesting feature is that the adaptive controller is the *only* controller available in the system. However, by choosing the number of parameters of the estimated model, it is possible to get different controller structures—for instance, a PID controller. The adaptive controller can tune ten parameters with a sampling period of 20–50 ms. The software admits easy configuration of a control system. The First Control

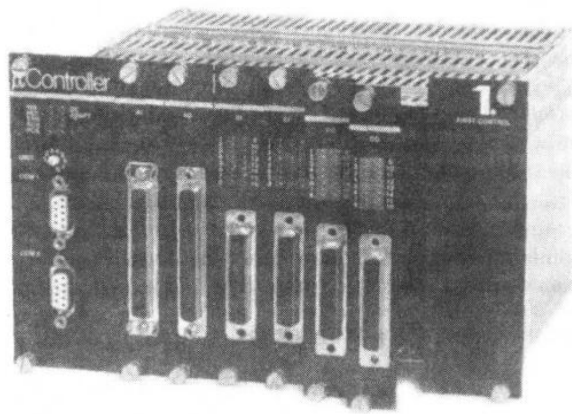

Figure 12.4 The MicroController from First Control. (With courtesy of First Control Systems AB.)

controller is shown in Fig. 12.4. Firstline is a distributed process control system with a block-oriented language for control design. The adaptive controller is incorporated as a standard function module. We will describe the adaptive control module in detail.

Principle. The adaptive control unit used in Firstloop and Firstline is based on recursive estimation of a transfer function model and a control law based on indirect pole placement. The controller also admits feedforward. The main advantage of using an indirect pole placement algorithm is that the system can be applied to nonminimum-phase systems and systems with time-varying time delays. This also implies that short sampling periods can be used. (Compare the discussion in Section 6.9.) The adaptive module comes in two versions, a standard module and an expert module. The standard module is intended for use by ordinary instrument engineers who are not specialists in adaptive control. The expert module shown in Fig. 12.5 is intended for specialists in adaptive control. Many parameters are given default values in the standard module. The variables that must be specified are shown in Fig. 12.5. The signal connections are measured value *MV*, setpoint *SP*, external control signal *UE*, feedforward *FF1*, *FF2*, and controller output *U*. The mode switches *ON*, *AUTO*, and *ADAPT* are for on/off, auto/normal, and adaptation on/off, respectively.

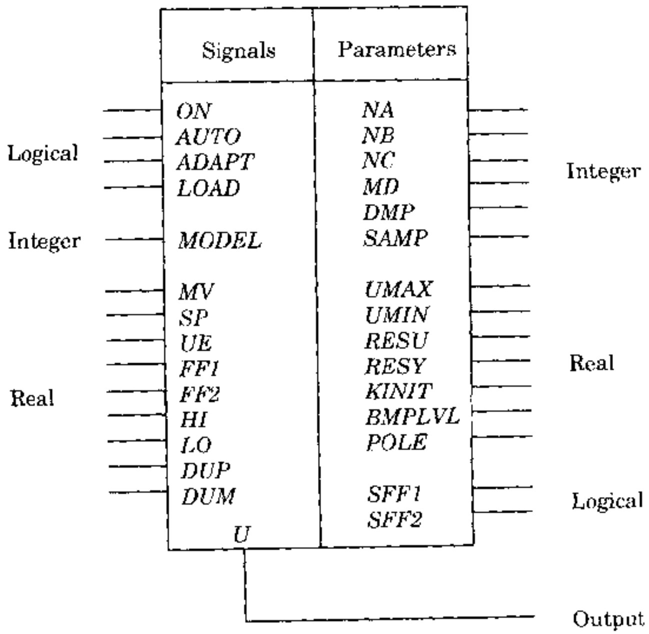

Figure 12.5 The expert module STREGX in Firstloop.

Parameters *UMAX* and *UMIN* define the actuator range. Variables *HI*, *LO*, *DUP*, and *DUM* specify the limits on the control signal that are used internally in the controller. The performance-related parameters are *POLE*, which gives the desired closed-loop pole, and *BMPLVL*, which gives the admissible initial change of the control variable at mode switches.

The desired closed-loop pole is the major variable to be selected. The choice of this variable clearly requires knowledge of the time scales of the process. The recommended rule of thumb is to start with a large value and gradually decrease it.

Parameter Estimation. The parameters of a transfer function model are estimated. Systems with variable time delay can be captured, provided that a large number of b parameters are used. Up to 15 parameters can be estimated in the model. The number of parameters in the model is specified by *NA*, *NB*, and *NC*. Common factors in the pulse transfer function are canceled automatically.

Control Design. The control design is based on pole placement. The desired response is characterized as a first-order system with delay. The remaining poles are positioned at the origin. The design of the algorithm is based on solving the Diophantine equation by a method that cancels common factors in the estimated polynomials. An LQG-based algorithm is also available. The details of the control design are proprietary.

Safety Network. The algorithm is provided with extensive safety logic. Adaptation is interrupted when variations in measured signals and control signals are too small. The limits are given with the parameters *RESU* and *RESY*. Adaptation is also interrupted when the control error is below a certain limit, and there are safeguards to ensure that the influence of a single measurement error or sudden large disturbance is limited. (Compare Section 6.9.) Measured values that result in large model errors are also given a low weight automatically. The details of the safety logic are not available. Different models can be stored for use in different situations. The controller is initialized by a model number equal to *MODEL* when *LOAD* changes from false to true.

Industrial Experiences. Firstloop and Firstline are used in a number of high-performance process control systems. They include control of pulp mills, paper machines, rolling mills, and pilot plants for chemical process control.

Discussion

The products described give an idea of how adaptive techniques are used in commercial products. Additional insight can be derived by analyzing the existing products and trends. Experience from the applications clearly indicates the need for tuning and adaptation; there are undoubtedly many control loops that are poorly tuned. This results in loss of energy, quality, and effective production time. It is also of interest that many different techniques are used, and there are also promising adaptive algorithms that have not yet reached the marketplace. A few specific issues will be discussed in more detail.

Computing Power. Industrial use of adaptive methods has been possible because of the availability of microprocessors. Most of the commercial systems are based on 8-bit processors, with their inherent limitations in addressing capability. This applies to all the PID auto-tuners and the first version of the ABB adaptive controller that used less than 64 kbyte of memory. With 16-bit processors and larger address spaces it is possible to use more sophisticated algorithms and better human-machine communication. The PID auto-tuners typically run with sampling rates of 10–50 Hz.

Intentional Perturbation Signals. To estimate parameters, it is necessary to have data with variations in the control signal. Such variations can be generated naturally or introduced intentionally. Natural perturbations can occur because of disturbances or poorly tuned controllers. Intentional perturbations can be introduced when natural perturbations are not present, as suggested by dual control theory. This method is used in several of the auto-tuning schemes. If prior information about the system dynamics is available, it is possible to find signals that are optimal for the purpose of estimating parameters. Relay feedback automatically generates an input signal having a lot of energy at the frequency at which the process has a phase lag of 180°. Although intentional perturbation signals are both useful and justified by theory, they are often controversial. It should be remembered, however, that poorly tuned controllers may also be considered perturbations.

Controller Structures. Different controller structures are used in the commercial systems. There are both PID controllers and general transfer function systems that admit feedforward and compensation for dead time. The main advantage of the PID structure is that it is close to current industrial practice. Within the PID family there are cases in which derivative action is of little benefit. Systems like the SattControl ECA40 can determine this and choose PI action automatically. However, there is no system that can choose the controller structure generally, although it seems possible to design such systems.

The benefits of feedforward control from measurable disturbances have been known for a long time. Experience with the ABB adaptive controller and Firstloop clearly shows the benefit of adaptive feedforward control. Since feedforward control critically depends on a good model, adaptation is almost a prerequisite for feedforward control. Adaptive controllers like the ABB adaptive controller and Firstloop use a controller structure that is a general transfer function model like

$$R(q)u(t) = T_1(q)u_c(t) + T_2(q)v(t) - S(q)y(t) \tag{12.1}$$

where u is the control variable, u_c is the command signal, v is a measured disturbance, and y is the controlled output. The polynomials R, S, T_1, and T_2 can be chosen so that the controller corresponds to a PID controller. However,

the controller modeled by Eq. (12.1) can also be much more general than a PID controller. It can incorporate many classical features such as filtering, disturbance models, Smith predictors, and notch filters. For more demanding control problems the general transfer function controller thus has significant advantages over the PID controller. However, more expertise in control engineering is needed to understand and interpret the parameters of a controller like Eq. (12.1). Since the PID controller is so common, we can expect it to coexist with more general controllers for a long time.

Multivariable Control. Multivariable control problems can be handled to a limited extent by using the feedforward feature in the ABB adaptive controller and Firstloop. None of the commercial systems admit truly multivariable adaptive control. Up to now there have not been many applications of adaptive control to true multivariable systems. This situation can be expected to change significantly because of the substantial interest in model predictive control.

Pre-tuning. It is interesting to note that many schemes have been provided with a pre-tuning feature. In some cases it appears that this was added afterwards. The reason is undoubtedly that too much user expertise is required for the standard algorithms. The selection of sampling periods or the equivalent time scales is a typical example. It appears that the relay method for automatic tuning would be an ideal method for pre-tuning.

Tuning Automatically or on Demand. The existing products include systems in which tuning is initialized on demand from the operator or automatically. Users of both schemes have documented their experiences. It appears that there are a number of processes for which controllers should be retuned for different operating conditions. In many cases there are measurable signals that correlate well with the operating conditions. In these cases it seems that the combination of on-demand automatic tuning with gain scheduling is a good solution. This will give systems that change parameters faster than systems with adaptation. Of course, it is convenient to have tuning initiated automatically, but it is difficult to give general guidelines for when tuning should be initiated. The simple schemes that are currently in use are often based on simple level detection. Further research is required to find conditions for retuning; this is discussed further in Section 13.4.

An analysis of the division of labor between human and machine gives another viewpoint on the question of on-demand or automatic tuning. When tuning is done on demand of the operator, the ultimate responsibility for tuning clearly remains with the operator or the instrument engineer. This responsibility is carried even further in some systems, in which the instrument engineer has to acknowledge the tuned values before they are used. A good solution would be a system in which the responsibility and the tuning techniques could be moved from the operator to the computer system. Ideally, the system should also allow the operator to learn more about control in general and the particular process in question. Experimental architectures that allow this are available, but not in commercial systems.

Requirements for the User

The requirements for the user are very different for the various commercial systems. The PID controllers in which tuning is initiated automatically require very little. Controllers with on-demand tuning require somewhat more knowledge on the part of the user. Systems such as the ABB adaptive controller and Firstloop can be regarded as toolboxes for solving control problems that are more demanding. They also allow complex control systems to be configured. This is clearly illustrated by the experiences from ABB adaptive controller installations. The system was designed by a very qualified team that included several first-rate Ph.D.s. The design team was also responsible for many of the initial installations, which were extremely successful.

More recent versions of the toolbox systems are much easier to use. Moderate-sized systems have been successfully implemented by instrument engineers with little knowledge of advanced control. There are several reasons for the increased user-friendliness of the systems. The safety logic has been improved significantly; modules in which many parameters are given default values have been designed; and computer-based configuration tools, with a lot of knowledge built in, have been developed. The toolboxes thus allow a user to get started quickly with a modest knowledge of adaptive control, and they also make it possible for a user to construct more advanced systems when more knowledge is acquired.

12.5 PROCESS CONTROL

There are many applications of adaptive control in the field of industrial process control. Some typical examples are discussed in this section. The applications give insight into how adaptive control can be used in practice.

Temperature Control in a Distillation Column

Although the SattControl auto-tuner has been used mostly for conventional loops for control of flow, pressure, and level, it has also been applied to more difficult problems. One example is temperature control in a distillation column. This is a conventional control loop in which the temperature in a tray of a distillation column is measured and the boil-up is manipulated. This control loop was part of a process system with many loops. There had been severe problems with the temperature control for a long time, and several attempts had been made to tune the loop. Figure 12.6 shows a recording of the temperature. The figure shows that the loop is oscillatory with the controller tuning that was used ($K_c = 8$, $T_i = 2000$, and $T_d = 0$). Also notice the long period of the oscillation. The controller was switched to manual at time 11:30, and the temperature then started to drift. Auto-tuning was initiated at time 14:00.

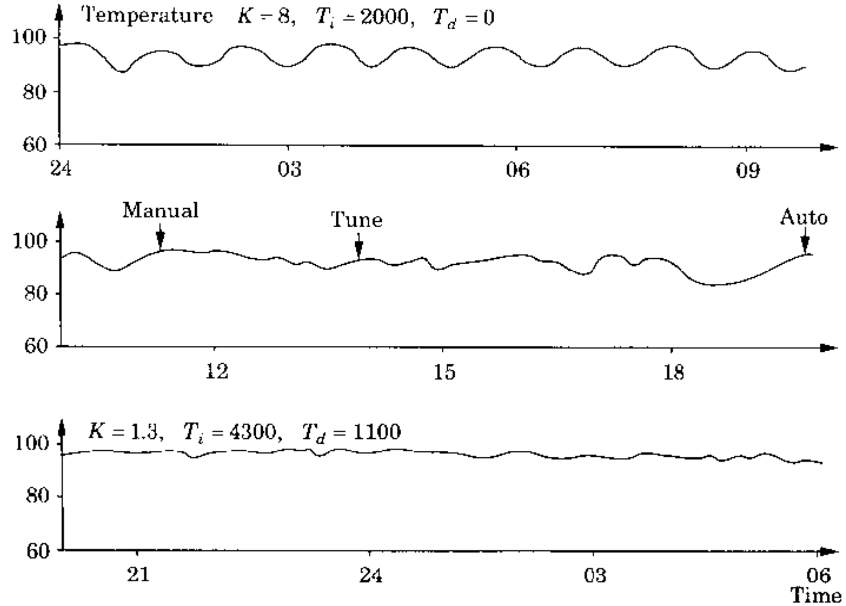

Figure 12.6 Application of the SattControl ECA40 to temperature control in a distillation column.

The tuning phase was completed after six hours at time 20:00, when the controller was automatically set to automatic control. The controller parameters obtained were $K_c = 1.3$, $T_i = 4300$, and $T_d = 1100$. Notice that the whole tuning procedure is fully automatic. The only action taken by the operator was to initiate tuning at time 14:00. The temperature variations during tuning are not larger than those obtained with the conventional controller settings. The example shows that the auto-tuner can cope with a process having drastically different time scales than those normally used.

Chemical Reactor Control

Chemical reactors are typically nonlinear. Characteristics such as catalyst activity change with time, as does the raw material. There are often inherent time delays, which may vary with production level. Poor control can result in lower product quality, damage to the catalyst, or even explosions in exothermic reactors. Chemical reactors are therefore potential candidates for adaptive control. The process in this application consists of two parallel chemical reactors in which ethylene oxide is produced by catalytic oxidation of ethylene. The process is exothermic and time-variable because of changes in catalyst activ-

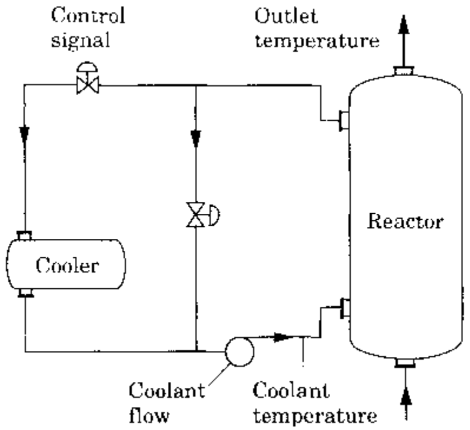

Figure 12.7 Schematic diagram of the reactor.

ity. It is essential to keep the temperature accurately controlled; a reduction of temperature variations improves the yield and prolongs the life of the catalyst. Stable steady-state operation is also a first step toward plant optimization.

The plant was equipped with a conventional control system that used PID controllers to control flow and temperature. The plant personnel were dissatisfied with the system because it was necessary to switch the controllers to manual control in case of many major disturbances, which could happen several times per day.

A schematic diagram of the process is shown in Fig. 12.7. The reactor is cooled by circulating oil to a cooler. The temperature of the coolant at the inlet to the reactor is the primary controlled variable, and the reactor outlet temperature and the coolant flow are also measured. The control signal is the flow to the cooler. The dynamics relating temperatures and flow to valve openings have variable delays and gains.

Disturbances in the process are caused by variations in the incoming gas and load changes. Large disturbances occur with changes in production level or with "shutdowns" caused by failure in surrounding process equipment. During shutdowns it is most important to maintain the process temperature as long as possible so that the production can be restarted easily. With the conventional control system, temperature fluctuations were around ±0.5°C during normal operation and up to ±2°C during larger disturbances. With adaptive control the variations were reduced to ±0.1°C during normal operation and ±0.5°C during large upsets.

The adaptive control system was implemented by using the ABB adaptive controller and the STAR3 module with feedforward from the reactor outlet temperature. By using the other modules in the system, it was also straightforward to handle the dual valves and to reset to manual mode for startup and

shutdown. The system has been in continuous operation since 1982 on a reactor at Berol Kemi AB, which produces 30,000 tons per year. The operational experiences with the system have been very good. With adaptive control, it was possible to reduce the temperature fluctuations significantly. The controllers are now kept in automatic mode most of the time, even during production changes. This has made it possible to revise operational procedures, since operators do not have to spend their time supervising the reactor temperature.

Pulp Dryer Control

Drying processes are common in the process industries. The mechanisms involved in drying are complex and poorly understood, and their dynamics depend on many changing factors. There are often significant benefits in improved regulation, since an even moisture content is an important quality factor. There are also significant potential energy savings. Drying processes are thus good candidates for adaptive control.

In pulp drying, a wet pulp sheet passes a steam-heated drying section and cooling section. A typical system is shown schematically in Fig. 12.8. The moisture content of the sheet entering the dryer is about 55%. At the exit, it is typically 10–20%. It takes about nine minutes to pass the dryer and about half a minute to pass the cooler. The dryer dynamics are complicated. It is influenced by many factors, such as the pH of the sheet. The measurements of the moisture content are obtained by a traversing microwave sensor that moves back and forth across the pulp sheet, describing a diagonal pattern on the sheet. When one traverse movement is complete, the mean value of the diagonal is stored in the computer, the mean value algorithm is reset, the sensor moves back, and the procedure repeats itself. It takes a little less than one minute

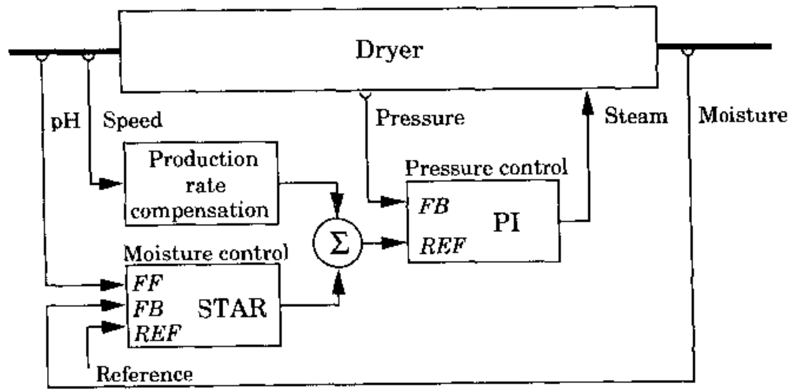

Figure 12.8 Schematic diagram of pulp drying and the control system.

for the sensor to move across the sheet. With manual control, the fluctuations in moisture content often exceed ±1%. The ABB adaptive controller was used in this application. The control system configuration is shown in Fig. 12.8. The moisture control is carried out by an adaptive software controller STAR. The moisture content measured by the traversing system is low-pass filtered and connected to the *FB* input of the STAR. The desired moisture content is chosen by the operator outside the ABB adaptive controller and software connected to the *REF* input of the STAR. The pH value, measured in an earlier process section, is used as the feedforward signal. This signal is connected to the *FF* input of the STAR. The control signal of the STAR defines the desired steam pressure, which is measured and controlled to the desired value by a conventional hardware PI controller. The control signal of this controller acts continuously on the steam flow valve.

The sampling period used in the adaptive controller was 3.5 minutes. A fourth-order Butterworth filter was used as an anti-aliasing filter. This was implemented by using the ABB adaptive controller tools. When the production rate was changed, large upsets were noticed, lasting for about 30 minutes, because it took 5–15 samples for the adaptive controller to settle. It was highly desirable to reduce these upsets, and this was done by introducing a special production rate compensation in the form of a pulse transfer function of the type

$$H(z) = \frac{b(z-1)}{z-a}$$

This gives a rapid change of the steam pressure when pulp speed changes. It was not necessary to make this filter adaptive. The system has been in operation since 1983 at a pulp mill at Mörrum's Bruk that produces 330,000 tons of paper pulp per year. The operational experiences have been very good. Fluctuations in moisture content have been reduced from 1% to 0.2%, which improves quality. It also allows the setpoint to be moved closer to the target value, resulting in significant energy savings.

Control of a Rolling Mill

The process control applications are typical steady-state regulation problems. The rolling mill control problem is much more batch-oriented. It illustrates the use of adaptive techniques in machine control. There are many types of rolling mills, each with its specific control problem. This particular application deals with a skin pass mill located at the end of the production line. The material processed by the mill may vary significantly in dimension and hardness.

The purpose of the mill is to influence quality variables such as hardness and yield limit. A schematic diagram of the process is shown in Fig. 12.9. Let v_1 be the speed of the strip entering the mill, and let v_2 be the speed of the strip at the exit. Because of the thickness reduction, the exit speed is larger

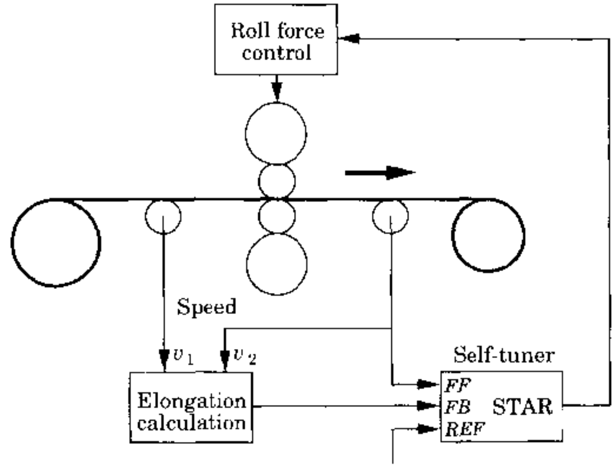

Figure 12.9 Schematic diagram of the rolling mill and the control system.

than the entrance speed. The elongation is defined as

$$\varepsilon = \frac{v_2 - v_1}{v_1}$$

The key control problem is to keep a constant elongation. There is a difficult measurement problem, since the velocity difference is so small. The process operates over a wide range of conditions; the following operating modes can be distinguished:

- Slow rolling at low speed during startup,
- Acceleration to fast rolling,
- Fast rolling at production speed,
- Intermediate decelerations to slow rolling or even to standstill,
- Deceleration to slow rolling at the end of the strip, and
- System at rest waiting for the next strip.

Transition from one mode to another is performed automatically on demand from the operator. It is essential that the control system handle these transitions well. The process dynamics relating elongation to roll force can be described as a high-order dynamical system with an open-loop response time of less than 0.05 s. Changes in production rate from 0 to 2000 m/min in less than 10 s are typical. The dynamics change drastically during the operation; the dynamics of rolling change because of variations in the speed, hardness, and dimension of the strip. There are also significant changes of the inertia of the coilers. All material starts on one coiler and ends up on the other. There are variations in the oil film on the roller bearings due to variations in speed and pressure. The dynamics of the hydraulic system vary with the operating point.

The changes in dynamics due to changing speed are predictable and can (in principle) be taken care of by gain scheduling. Variations in dimension can be handled similarly. The hardness cannot be measured directly on-line, so it must be handled by feedback and adaptation.

The ABB adaptive controller was used in this application. A block diagram of the control system is shown in Fig. 12.9. The speed variations are taken care of in an elegant way. In the ABB adaptive controller, sampling can be triggered by an arbitrary signal. In this case it is triggered by the pulse counters that measure strip speed. This means that sampling is related to the length of the strip, not to time. This is a simple way of making the control system invariant to strip speed (the same idea was used in the ship steering example in Section 9.5). The measurement of the velocity difference is implemented by using pulse generators and counters.

For each strip a saved model is loaded into the controller, and the adaptation is switched on with some delayed action (15 sampling intervals) to avoid adaptation during the first few steps, in which the measurement is irregular. The initial model is taken from a soft strip so that there will be no excessive control action at startup. Soon enough, the controller will adapt to the conditions of the new strip. Figure 12.10 illustrates a typical run of a strip. Notice

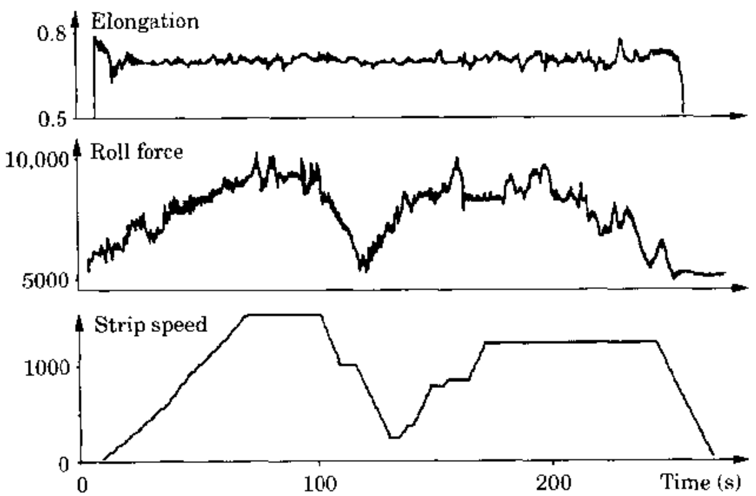

Figure 12.10 Elongation, roll force, and strip speed during a typical run with the system.

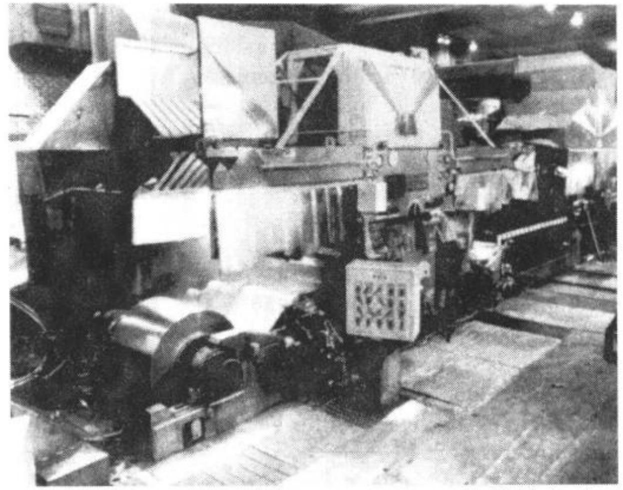

Figure 12.11 The cold rolling mill at Avesta-Sheffield is controlled by First Control's adaptive system. (With courtesy of Avesta-Sheffield, precision strip AB, Kloster.)

in particular how well the system copes with the velocity variations and with the mode changes. The installation of the system took about a week, mostly devoted to function and signal checking and tests. The controller functioned almost immediately when connected to the process. After that, approximately two days were devoted to checking and tuning performance. This involved experiments with different sampling rates.

A significant part of the installation time also involved other parts of the system, particularly the logic. Operational experiences with the adaptive control system have been very favorable. The variation in elongation was better than is found with a conventional system, and the adaptive system also settled faster during mode switches. The system has been in continuous operation since 1983.

Figure 12.11 shows a cold rolling mill at Avesta-Sheffield in Långshyttan, Sweden. The process is controlled by First Control's adaptive control system since 1990. The adaptive regulators keep the deviations in the strip thickness within 2–3 μm, which is considered to be very accurate for this kind of mill.

Pulp Digester

Control of the pulp digester is an important part in manufacturing of chemical pulp. The raw material is wood chips, which are broken down into fibers by processing in a liquor composed of sodium hydroxide and sodium sulfide (white

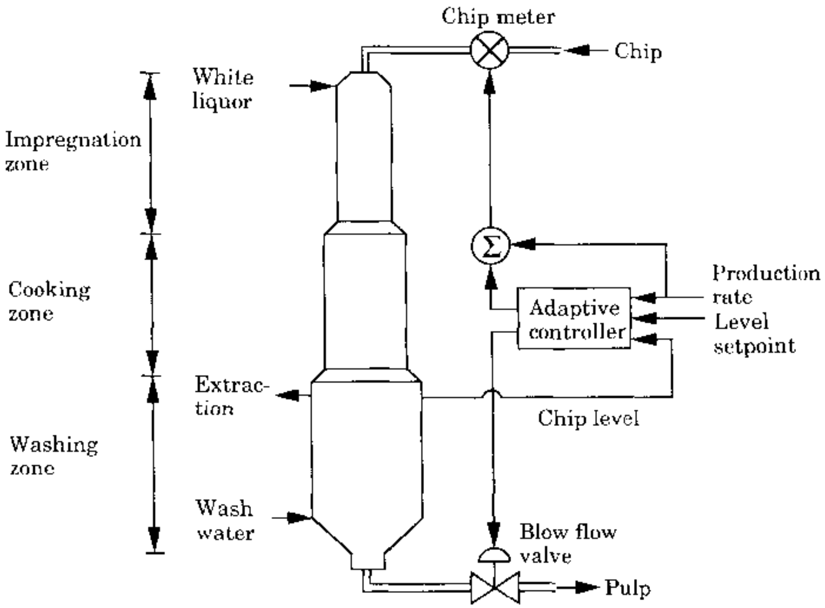

Figure 12.12 Schematic diagram of the chip level controller for a continuous Kamyr digester.

liquor). The process operates either in batches or, more commonly today, as a continuous process.

The Kamyr digester (see Fig. 12.12) is the standard continuous process. The production rate is determined by the chip meter, which feeds chips into the top of the digester. The flow of pulp from the digester is controlled by the blow flow at the bottom. The digester has three zones: impregnation, cooking, and washing. The dynamics that describe the material transport and the chemistry in the digester is very complicated. The total residence time in the digester is about 5 hours. An important control problem is the control of the chip level, which is controlled by the blow flow. The chip level signal is calculated from three strain gauges by using a scheme developed by MoDo Chemetics. The study reported here is a feasibility study made by Pulp and Paper Research Institute of Canada (Paprican) and the pulp company MacMillan Bloedel in Vancouver. The study has resulted in an adaptive controller for digester control developed in cooperation between MoDo Chemetics in Vancouver and Paprican. The commercial adaptive controller manipulates two inputs (blow flow and chip meter) as indicated in Fig. 12.12; in the feasibility study, only the blow flow was manipulated by the adaptive controller.

The industrial digester in the study produced 350 tons per day of kraft pulp. Two grades, R and K, are manufactured. From identification experiments

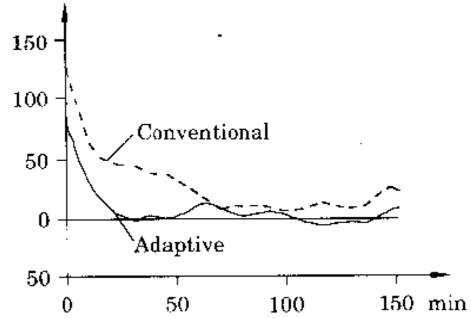

Figure 12.13 Autocovariance of the chip level under conventional and adaptive control. (With courtesy of Paprican.)

it was found that the digester can be described by the model

$$(1 + a_1 q^{-1})\Delta y(t) = (b_0 + b_1 q^{-1} + b_2 q^{-2})\Delta u(t-2) + (1 + c_1 q^{-1} + c_2 q^{-2})e(t)$$

where $\Delta y(t)$ is the difference in level, $\Delta u(t)$ is the change in blow flow, and $e(t)$ is white noise. The sampling period is 5 minutes. The identification experiments also indicated that it would be feasible to use fixed values of all the parameters except the b_i's. The adaptive controller will thus be able to compensate for gain changes and changes in the time delay of the process. A GPC algorithm with $N_u = 1$, $N_1 = 1$, and $N_2 = 15$–20 is used (see Eq. 4.61). Figure 12.13 shows the autocovariance of the level when conventional PID control and adaptive control were used. The chip level signal is essentially uncorrelated after three lags (15 minutes). The standard deviation of the level decreased from 11.3% to 8.6%. This improvement of the chip level leads to direct improvements in pulp quality. Table 12.1 shows permanganate number (P-number), which is a standard laboratory test of the residual lignin in the pulp. The P-number is closely related to the kappa number. The P-numbers were measured on samples from the blow flow collected once every two hours. For both grades (R and K) the average P-numbers are closer to the target values, and the standard deviations are reduced.

In summary, the advantages of the adaptive controller are

- Reduction of chip level and P-number variability,
- Reduced need for operator intervention,
- Elimination of manual retuning, and
- Prediction of potential problems with hang-ups in the chip column.

The pulp digester study is an example of a special-purpose adaptive controller. The process model is tailored to fit the specific application, and the parameters can be related to physical parts of the system.

Table 12.1 P-number variability under conventional and adaptive chip level control. (With courtesy of Paprican.)

Controller	Grade	Setpoint	Mean	Std. dev.	No days
Conventional	R	23.0	22.2	2.06	23
	K	21.0	19.9	1.91	6
Adaptive	R	23.0	22.4	1.76	18
	K	21.0	20.6	1.70	7

Pulp digesters have also been controlled by standard adaptive controllers. One example is the Vallvik mill at Assi Domän in Sweden where Novatune controllers in an ABB Master system are used extensively. Several Novatunes are used to control temperatures, flows and levels. The system was installed and commisioned by the regular mill staff with a core team of two enthusiastic engineers. The critical parameters in the Novatune were the sampling period and the prediction horizon; these values had to be selected individually for each application. Default values were used for the other parameters. The prediction sampling period is typically chosen to be 60% to 90% of the dead time, the predictions horizon is chosen as $KD = 2$ and the controller complexity as $NA = NB = NC = 3$. The standard procedure is to run the controllers in manual mode. The parameter estimation is switched on with restrictions on the control action, which are gradually removed.

The experience with adaptive control has been very good. Control performance is significantly better with adaptive control than with PID control. The systems have not required much attention after installation. The reason for improved performance is that tighter control is obtained with adaptive control. Experiments at the plant indicated that there was a good correlation between variations in chip level and the kappa-number. By introducing adaptive control of the chip level it was also possible to significantly reduce the variation in the kappa-number. The standard deviation was reduced from 0.52 to 0.30. It has also been observed that the adaptive controllers recover much faster from large upsets than the systems used previously.

12.6 AUTOMOBILE CONTROL

Microprocessor-based engine control systems were introduced in the automotive industry in the 1970s to address the demands of increased fuel economy and reduced emissions. Early electronic control systems had modest application. Today, the powertrain computer accomplishes a multitude of control tasks, including vehicle speed or "cruise" control, idle speed regulation, automatic transmission shift actuation, control of various emission-related systems,

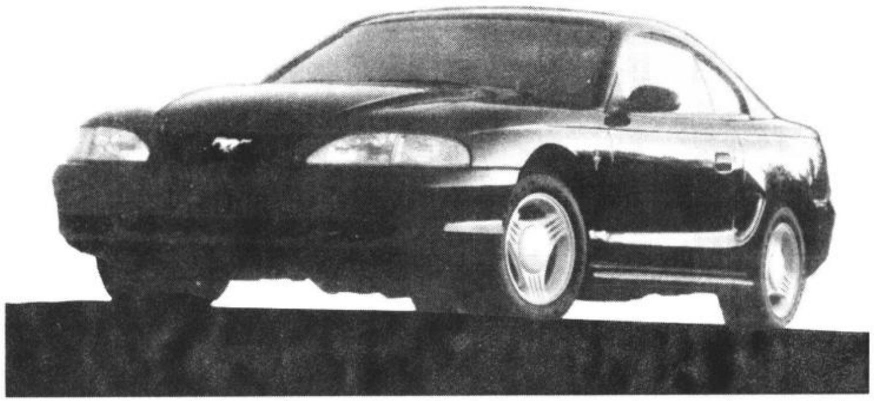

Figure 12.14 This Ford Mustang has state-of-the-art adaptive power train controls. (With courtesy of Ford Motor Company.)

fuel control, and ignition timing, as well as many diagnostic functions. These highly I/0 intensive systems must be cost effective and function acceptably in many thousands of vehicles with attendant manufacturing variability over a wide range of operating conditions.

Many of the control functions in automobiles are open-loop look-up table oriented. Some automatic calibration methods have been developed to optimize table entries with respect to fuel economy, constrained by emissions. Typical closed-loop structures are comprised of individual operational loops, often PI or PID, and may contain several feedforward paths that are designed to reject measurable or predictable disturbances. Applications of adaptive control concepts can be found in many of those powertrain control functions where online self-tuning techniques are used to adjust controller parameters (in most cases, the feedforward parameters) to compensate for component and operating condition variability. One such adaptive control structure is the air-fuel ratio control method introduced by Ford in the mid-1980s to reduce sensitivity to component variability and calibration inaccuracy. Figure 12.14 shows a car with adaptive power train control.

Modern automobiles require precise control of air-fuel ratio to attain high catalytic converter efficiency and minimize tailpipe emissions. Air-fuel ratio control has two principal components: a closed-loop portion in which the fuel injectors are regulated in response to a signal fed back through a digital PI controller from an exhaust gas oxygen sensor located in the engine exhaust stream, and an open-loop or feedforward portion in which fuel flow is controlled in response to an estimate of the air charge entering the engine. (Compare Section 9.5.) This open-loop portion of the control is particularly important during engine transient when the inherent delay of the engine and exhaust system obviate the effectiveness of feedback, and during cold engine operation before the exhaust gas oxygen sensor has reached operational temperature. The purpose of the adaptive algorithm is to adjust the open-loop feedforward gain to reduce deviation from stoichiometric air-fuel ratio operation and improve emission performance under open- and closed-loop operation. This is essentially a gain scheduling process in which an adaptive multiplier is stored in a look-up table as a function of engine speed and load. Initially, all the table entries are unity. As the engine operates throughout its range, the appropriate cell values are increased or decreased to correct for parametric changes or inaccuracies in the initial calibration. In contrast to typical gain scheduling techniques, this adaptation is continuous throughout the vehicle's life.

Another application of adaptive control at Ford is in the area of automotive speed control or "cruise" control. These systems must provide acceptable steady-state error, excellent disturbance rejection, unnoticeable throttle movement, and must be robust to vehicle-to-vehicle variability and operating condition. An adaptive control design based on sensitivity analysis and gradient methods has been used to continuously tune the gains of a PI controller. This was accomplished by constructing a single quadratic cost function and adjusting the proportional and integral control gains to minimize this function. Additional modifications, such as projection and deadband together with slow adaptation, were used to avoid parameter drift and ensure robustness. In this manner, speed control performance is optimized for individual vehicles and operating conditions providing improved performance and reduced calibration effort compared to conventional fixed gain controllers.

12.7 SHIP STEERING

A conventional autopilot for ship steering is based on the PID algorithm. Such a controller has manual adjustment of the parameters of the PID controller and often also a dead zone called *weather adjust*—a simple version of a performance-related knob. Manual adjustments are necessary because the dynamics of a ship vary with speed, trim, and loading. It is also useful to change the autopilot settings when disturbances in terms of wind, waves, currents, and water depth are changed. Adjustment of an autopilot is a burden on the crew. A poor adjustment results in unnecessarily high fuel consumption. It is therefore of interest to have adaptive autopilots. A ship steering autopilot, Steermaster 2000 from Kockum Sonics AB in Sweden, and a roll damping equipment, Roll-Nix from SSPA Maritime Consulting AB in Sweden and Hyde Marine Systems in Ohio, are described in this section.

Ship Steering Dynamics

Simple ship steering dynamics were presented in connection with the discussion of gain scheduling in Section 9.5. That section detailed how the dynamics vary with the velocity of the ship and showed how the variations could be reduced by gain scheduling. It has been shown by hydrodynamic theory that the average increase in drag due to yawing and rudder motions can be approximately described by

$$\frac{\Delta R}{R} = k\left(\bar{\psi}^2 + \lambda \bar{\delta}^2\right) \tag{12.2}$$

where R is the drag and $\bar{\psi}^2$ and $\bar{\delta}^2$ denote the mean square of heading error and rudder angle amplitude, respectively. The parameters k and λ will depend on the ship and its operating conditions. The following numerical values are typical for a tanker:

$$k = 0.014\,\mathrm{deg}^{-2} \qquad \lambda = 1/12$$

It is thus natural to use the criterion

$$V = \frac{1}{T}\int_0^T \left(\left(\psi(t) - \psi_{\mathrm{ref}}\right)^2 + \lambda \delta^2(t)\right) dt \tag{12.3}$$

as a basis for the design and evaluation of autopilots for steady-state course keeping. The disturbances acting on the system are due to wind, waves, and currents. A detailed characterization of the disturbances and their effect on the ship's motion is difficult. In a linearized model, disturbances appear as additive terms. It is common practice to describe them as random signals; the waves have a narrow band spectrum. The center frequency and the amplitude may vary significantly.

Autopilot Design

An autopilot has two main tasks: steady-state course keeping and turning. Minimization of drag induced by the steering is the important factor in course keeping, and steering precision is the important factor in turning. It is therefore natural to have a dual-mode operation. These two modes are described in the text that follows, together with the basic autopilot functions.

The influence of variations in the speed of the ship is handled by gain scheduling. The other disturbances are taken care of by feedback and adaptation. Implementation of the gain scheduling is discussed in Section 9.5. It requires a measurement of the forward velocity of the ship. If disturbances are regarded as stochastic processes, steady-state course keeping can be described as a linear quadratic Gaussian problem. It is then natural to estimate an ARMAX model (Eq. 2.38). The particular process model used is

$$\begin{aligned}\Delta\psi(t) - a\Delta\psi(t-h) &= b_1\delta(t-h) + b_2\delta(t-2h) + b_3\delta(t-3h)\\ &\quad + e(t) + c_1 e(t-h) + c_2 e(t-2h)\end{aligned} \tag{12.4}$$

This model is built on Nomoto's approximation (compare Section 9.5). The additional b term was introduced to allow additional dynamics to be captured as an increased time delay. The difference occurs because there is a pure integration in the model from rate of turn to heading angle. A control law that minimizes the criterion of Eq. (12.3) is then computed by using the certainty equivalence principle. This approach requires the solution of a Riccati equation, which can be done analytically in the particular case. A straightforward minimum-variance control law was used in some early experiments. This was replaced by the LQG control law described previously, because there were significant advantages at short sampling intervals, which could not be used with the minimum-variance control law. The sampling interval in the model is set during commissioning.

Turning Controller

The major concern in turning is to keep tight control of the motion of the ship, even at the expense of rudder motions. For high turning rates the dynamics of many ships are nonlinear. The normal course-keeping controller can handle small changes in heading, but it cannot handle large maneuvers because of the nonlinearities discussed previously. A special turning controller was therefore designed. The controller is a high-gain controller in which the feedback is of PID type. (Compare Fig. 1.3.) Appropriate PID parameters are determined during commissioning. The model used is nonlinear. It is designed so that the command signal is *turning radius*. The turning rate is thus $r = u/R$, where u is the speed of the ship and R is the turning radius.

Human-Machine Interface

The fact that turning radius is used as a command signal instead of turning rate simplifies maneuvering considerably, because it is easy to determine an appropriate turning radius from the chart. It also improves path following, since the speed of the ship may change during a turn. This is then compensated for automatically. The man-machine interface is very simple. There is one joystick to increase and decrease the heading. An optional joystick provides override control; whenever this is moved, it gives direct control of the rudder angle. Control can be transferred to the autopilot by a reset button. In making a turn, the desired turning radius is set by increase-decrease buttons. The turn is initiated when the joystick is moved to the new desired course. The turn is then executed, and the ship turns until the desired course is reached. The fixed-gain controller is used during the turn, and the adaptive course-keeping controller is initiated when the turn is complete.

There are no adjustments on the course-keeping controller; everything is handled adaptively. Some default values are set during commissioning, but the fixed-gain controller can be activated when the operator pushes a switch

labeled *fixed control*. This is typically used when there are heavy waves coming from behind (called a quartering sea). This condition makes steering difficult because the effective rudder forces are small and the disturbing wave forces are large.

Operational Experiences

Early versions of the autopilot were field-tested in 1973, and the product was announced in 1979. The product is used in various kinds of ships. One installation, in a ferry that navigates between Stockholm and Helsinki, has been in continuous operation since 1980. It uses adaptive control all the time. The ability to cope with large variations in speed has been found to be very useful, and the turning radius feature is particularly useful for navigation in archipelagos, where a lot of maneuvering is necessary. Figure 1.24 indicates the improvements in course-keeping that can be obtained through adaptation. The decreased drag with the data shown in the figure corresponds to a reduction in fuel consumption of 2.7%.

Rudder Roll Damping System

On many ships it is desirable to reduce the rolling motion. Conventional roll damping systems on large naval ships use active fins or active as

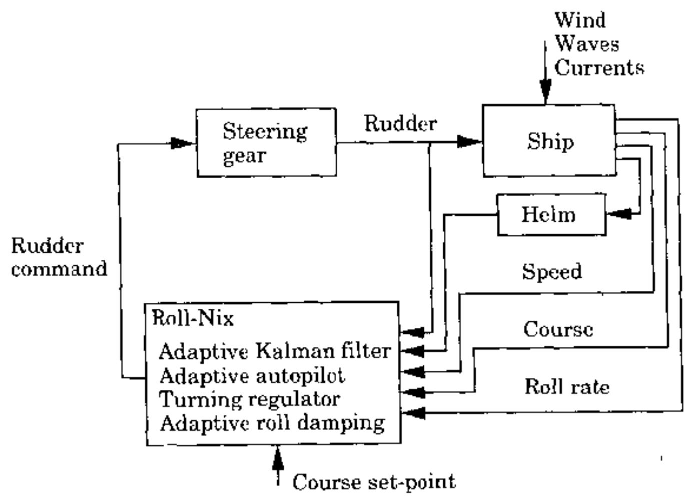

Figure 12.15 Block diagram of the Roll-Nix roll damping system. (With courtesy of SSPA Maritime Consulting AB.)

well as passive tanks. These systems are expensive to install, especially for retrofits. A third approach to roll damping is to use the rudder for roll damping as well as for maneuvering. High-frequency movements of the rudder damp the rolling without influencing the mean value of the heading of the ship. Such a system can be inexpensive, since it can easily be connected to the ordinary steering system. One such system, Roll-Nix, has been developed by SSPA Maritime Consulting in Gothenburg, Sweden. The system is also marketed by Hyde Marine Systems in Cleveland, Ohio. A block diagram of the system is shown in Fig. 12.15. Roll-Nix includes an adaptive Kalman filter, an adaptive course-keeping autopilot (optional), a high-gain turning controller (optional), and an adaptive roll dámping controller. The first three parts are similar to those described for the Steermaster 2000 autopilot.

The system uses a roll rate sensor together with course gyro and speed log to determine rudder commands that are superimposed on the ordinary autopilot commands and fed into the steering engine. The operating principle is that the roll movements created by the rudder are opposite those of the roll movements caused by the waves. These counteractive moves damp the roll motions of the ship. In designing the roll damping system it is important to have quick rudder motions. Slow and large motions will influence the course

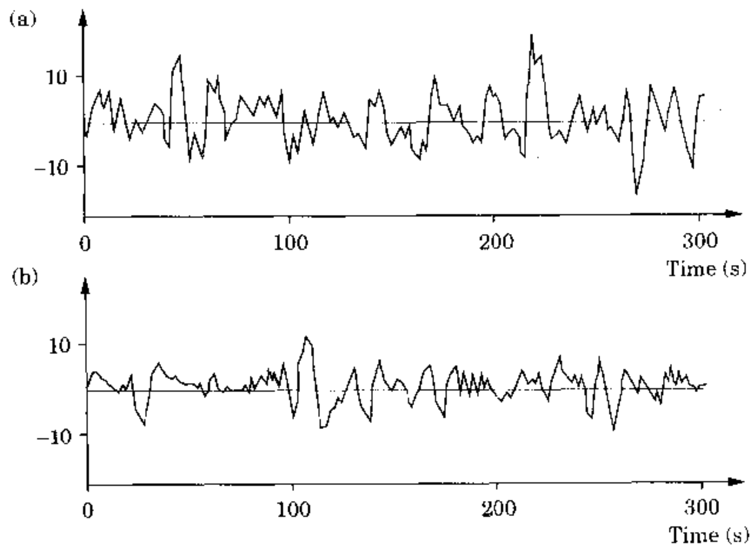

Figure 12.16 Results from sea trials with an attack craft at 27 knots and stern quartering seas (4 Beaufort): (a) without Roll-Nix; (b) with Roll-Nix. The significant roll angle was reduced by 58%, and the maximum roll angle was reduced by 53%. (With courtesy of SSPA Maritime Consulting AB.)

keeping. The Roll-Nix system is provided with an autopilot as an option. The adaptive feature of the roll damping system is necessary to handle different weather conditions and ship speed. The Kalman filter is used to obtain an accurate roll motion signal from the measured roll rate.

Roll-Nix has been tested on several types of ships. For instance, the system has been tested on two Royal Swedish Navy ships: one attack craft and one mine layer. The sea trials show that a significant roll reduction of 45–60% can be obtained for both the standard deviation and the maximum angle of the roll. The result from a sea trial with an attack craft is shown in Fig. 12.16. The roll reduction increases with increasing speed and rudder rates. Tests were also done on the mine layer HMS Carlskrona in September 1987. The following quote from the captain, Commander Hallin, gives an illustration of the performance of the system.

> This particular occasion was when the ship was off the Dutch coast, bound for Helder, with seas coming in from astern on the port quarter. I was resting in my cabin. The time was 04.00 hrs. Suddenly I sensed that the ship had started to roll perceptibly, and I wondered what was going on. At once, I went up on deck and asked the officer of the watch what on earth was happening, and what the reason was for this sudden increase in the ship's rolling motion. I was surprised to receive the reply, "We have just switched off the Roll-Nix. We need to have some data without Roll-Nix working, to see how much damping can be achieved." I think that that is the most illustrative experience I have had of the Roll-Nix system to date.

12.8 ULTRAFILTRATION

Patients with little or no renal function need some form of artificial blood purification to stay alive. In dialysis the blood is cleansed of waste products and excess water, and the electrolytes in the blood are normalized. More than 350,000 patients all over the world undergo this treatment a couple of times a week. In its most common form, hemodialysis, the blood flows past a semipermeable membrane with a suitably composed dialysis fluid on the other side. Because of the large number of different dialyzers that are on the market, the control algorithm in the dialysis machine must be able to handle a wide span in process gain and other process characteristics.

An adaptive pole placement controller has been used in the fluid control monitor (FCM) developed by Gambro AB in Lund, Sweden. The system has been in use for many years and it has performed very well. This is probably one of the most widely used adaptive controllers in the world today. In this section we describe the system.

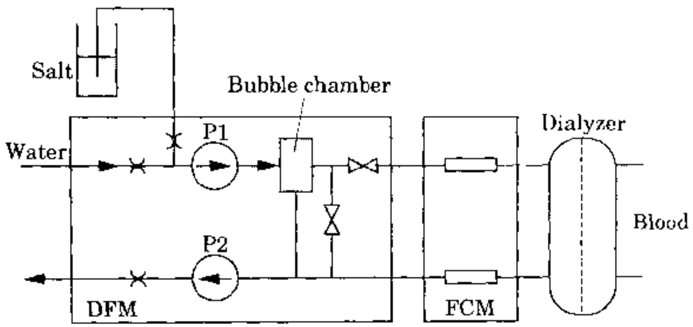

Figure 12.17 Schematic diagram of a dialysis system.

Process Description

A schematic view of the Gambro AK-10 dialysis system is shown in Fig. 12.17. Only the parts that are relevant to flow and pressure control are shown in detail. Clean water is heated to around 37°C, and salt is added to physiological concentration. A pressure drop in the restrictor is created by the first pump to degas the solution. The restrictor and the first pump (P1) determine the flow into the dialyzer. Because of the compressibility of the air in the bubble chamber, flow changes to the dialyzer will be slowed down by a time constant.

After passing a few measuring devices and a valve, the fluid leaves the dialysis fluid monitor (DFM) and passes the first flow-measuring channel of the FCM before entering the dialyzer. Before returning to the DFM, the second measuring channel of the FCM is passed. In the DFM a few more measuring devices and valves are passed before the second pump (P2). A restrictor is placed on the outlet to allow positive pressures in the dialyzer.

To maintain a specified transmembrane pressure, the DFM has a control system that is based on a conventional fixed-gain digital PI controller. (See the block diagram in Fig. 12.18.) This controller has a sampling period of 0.16 s and an integration time of about 30 s. The purpose of the fluid control module is to control weight loss during the treatment. This is done by the external control loop shown in Fig. 12.18, which has the flow difference Q_f as the measured variable and the setpoint to the pressure controller p_c as the control variable.

Process Dynamics

The dialyzer dynamics can be approximately described by the model

$$C\frac{dp}{dt} = Q_f - Bp$$

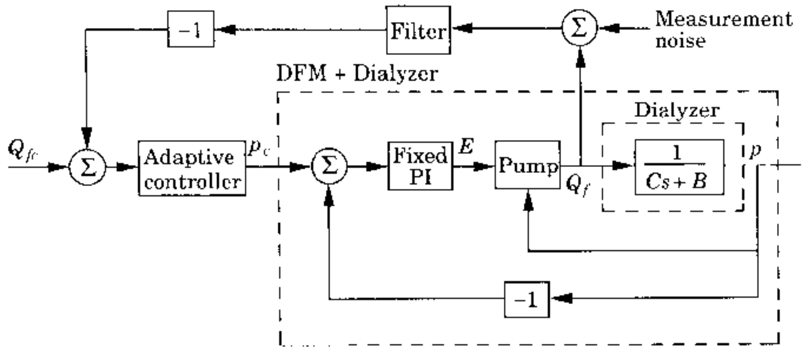

Figure 12.18 Block diagram of the system for controlling transmembrane pressure p and the flow difference Q_f control system.

where p is the transmembrane pressure and Q_f is the net fluid flow from the dialyzer. The constant C is the compliance. Parameter B, which represents the static gain, may, for example, vary from $1.6 \cdot 10^{-12}$ to $120 \cdot 10^{-12}$ ($m^3\ s^{-1}\ Pa^{-1}$), that is, a gain variation by a factor of 75.

The complete dynamics of the pressure loop can be approximately described as a second-order transfer function. It has one pole associated with the dynamics of the ultrafiltration and another associated with the pressure control system. The PI controller is tuned conservatively so that both poles are real. The dominating time constant is 30–50 s. The transfer function from the pressure setpoint to the flow Q_f then also has the same poles, but it also has a zero corresponding to the pole $s = -B/C$ of the ultrafiltration (see Fig. 12.18). This zero can change significantly with the type of dialysis filter used. A consequence is that there is a drastic difference in the dynamics obtained for different filters.

The main function of the system is to control the total water removal V during the treatment. The water removal is given by

$$\frac{dV}{dt} = Q_f \tag{12.5}$$

An Earlier Control System

An earlier system used a PI controller in the outer loop. Because of the large gain variations, it was necessary to use a conservative setting with low gain. This resulted in very sluggish control of the weight loss. Experiments with various simple forms of gain adjustment did not solve the problem and it was decided to test if an adaptive controller was feasible.

Adaptive Control

The adaptive controller was designed as an indirect adaptive pole placement algorithm.

Parameter Estimation. The dynamics can be expected to be of third order, representing the dynamics of the pressure loop and the dynamics of the filter introduced to filter the flow signal. This filter has a time constant of about 30 s. Experiments with system identification indicated, however, that data could be fitted adequately by

$$Q_f(t) = aQ_f(t-h) + b_1 p_c(t-h) + b_2 p_c(t-2h) \qquad (12.6)$$

where Q_f is the filtration flow and p_c is the setpoint of the pressure loop. This model represents first-order dynamics with a time delay. A sampling interval of 5 s was found to be suitable. The parameter estimation was made on differences to avoid problems with a constant level in the signals.

The estimated steady-state gain is an important parameter. With a low estimated gain, the gain in the controller will be large. It is therefore advantageous to have the sum of the b parameters as one of the estimated parameters so that it is easy to set a lower limit to the estimated gain. This has been done in the FCM by using the regression vector

$$\begin{pmatrix} Q_f(t) & p_c(t) & p_c(t)-p_c(t-h) \end{pmatrix}$$

instead of

$$\begin{pmatrix} Q_f(t) & p_c(t) & p_c(t-h) \end{pmatrix}$$

If the estimated gain becomes too small, the estimate is stopped at the limit.

A constant forgetting factor of 0.999 is used to track slowly time-varying parameters. To improve numerics, only the diagonal elements of the covariance matrix P are divided by this factor. It is well known that the equation for $P(t)$ may be sensitive to numerical precision when a forgetting factor is used. This is because the eigenvalues of the P-matrix may be widely separated. Several methods to handle this problem were described in Chapter 11 and in the discussion of the ABB adaptive controller and Firstloop in this chapter. In this case the problem was avoided by careful scaling, and an ordinary recursive least-squares method could be used.

Control Design. A conventional pole placement algorithm and a design that guarantees integral action were used. (See Section 3.6.) Several factors influence the choice of desired closed-loop poles. If a smooth control is desired in the steady state, the speed of setpoint changes should not be set too high. Second, the first step response at startup must not be quicker than the time required to get a reasonable model. A reasonable response time in accumulated flow is one hour. The other closed-loop poles, which correspond to flow changes, were specified by time constants of 25 s, 15 s, and 15 s.

The controller can be reparameterized to correspond to a PID controller with a filtered derivative part. The structure was chosen so that the controller corresponds to a discrete-time PID controller in which the reference signal enters only the P and I parts. This corresponds to $\beta = 1$ in Eq. (8.3) in Chapter 8. A possible common factor in the estimated model was canceled before entering the design calculations.

Special Design Considerations

Control of fluid removal during dialysis has a direct influence on the patient's well-being. This imposes heavy demands on the control system. Several safety features have been included. Smooth performance from the first moment of control is essential. This can be achieved by a careful choice of certain parameters, as we discuss next.

Filtering. The measured flow signal is corrupted by measurement noise. Since a new value is available every second, it is possible to filter the signal. With a sampling period for control of 5 s, it was found to be suitable to use a first-order filter with a time constant of 30 s to filter flow and accumulated flow before using the values in the control algorithm. This smooths the control signal considerably without preventing fairly quick setpoint changes.

Limits on Setpoint Changes. Both the absolute level and the rates of setpoint changes were limited on the basis of physical constraints. The PID controller was provided with conventional anti-windup protection to avoid problems with saturation. Parameter updating is also interrupted when the pressure setpoint is kept constant at a limit. At startup, when the model parameters may be far from their best values, it is also wise to prevent the control algorithm from changing the control signal (i.e., the pressure setpoint) too rapidly. The rate limit on the pressure setpoint prevents this; experience has shown that this limit is hit only rarely.

Startup. A critical moment for an adaptive controller is the start, before the model parameters have been accurately estimated. It was required that its step response be almost perfect from the beginning. For this reason, most of the development time was spent in adjusting the parameters to ensure a smooth start. The following parameters were then found to be important:

- Initial values of the parameter estimates,
- Initial values of the covariance matrix P,
- The desired closed-loop poles,
- The time allowed for signals to settle before estimation and control starts,
- Limits on the estimated parameters (especially the static gain), and
- The limit on control changes (and control).

The initial values of the parameter estimates are important, since they determine the initial controller parameters. They were chosen to model a high-gain dialyzer, with an extra time delay, to give a cautious low-gain controller. This is perfect for a highly permeable (i.e., high-gain) membrane, but for normal membranes the pressure changes will be too small, a situation that is soon detected by the parameter estimator.

It is important to choose the P matrix carefully. This determines the speed of parameter estimation. Values of P that are too large will make the estimates noisy, and there is a risk that the estimates may temporarily give bad controllers. Also, a value of P that is too large can quickly eliminate the carefully chosen initial parameters in the estimator. With values of P that are too small the time needed to find a good model can be very long, a situation that is not at all acceptable.

It was found to be advantageous to introduce a lower bound on the estimated gain in the model. With low-gain dialyzers there would otherwise be a tendency for the estimator to decrease the gain estimate too much, and the controller gain would be too high for a while. A suitable limit for the model gain could be determined from the known data of existing dialyzers. To facilitate the checking of the estimated gain, a special form of the process model was used. The estimated pole was also bounded away from a pure integrator, since this pole enters the expression for the gain limit.

The limit on the setpoint changes also helps to ensure a smooth startup. The desired closed-loop poles are important design parameters. The equivalent time constants should be chosen to be long enough to give the estimator time to find a good model before the setpoint is approached for the first time. They should also be as short as possible to give a rapid response to setpoint changes. The equivalent time constants of the closed-loop systems were chosen to be 720, five, and three sample intervals, which correspond to 1 hour, 25 s, and 15 s, respectively. Without the requirement of a smooth startup it would have been possible to speed up the desired closed-loop dynamics considerably. However, setpoint changes are not very frequent, and smooth startup is much more important than rapid setpoint changes.

If by chance the desired pressure were already set at startup, there would be no pressure change that would help to improve the estimates of model parameters. Therefore there is a period of forced small pressure changes for the first eight minutes after a reset. This is accomplished by periodic changes of the setpoint every 45 s.

With an adaptive controller it is very important to ensure that the estimated model is never destroyed. Therefore the estimator should always be given true values for control and measured signals. If for some reason, such as an alarm situation causing the DFM to bypass the dialysis fluid, the control signal is not allowed to do its job, the estimator must be turned off. The controller will then use the old estimates for a while.

After all such breaks and at startup, a settling period is allowed, during which correct signals are entered into all the vectors but no estimation is done. This settling period is very important, especially at startup, when the estimates are most sensitive to changes in the signals. Errors in the signals

also force the P-matrix to decrease rapidly, so future learning is slowed down considerably.

Alarms. Appropriate alarms are an important part of any useful control system. An alarm indicates if the volume control error is too large and also if something is wrong in the dialysis fluid monitor or with the pipes. If there is a stop in the blood pipe from the dialyzer to the drip chamber, the blood pressure within the dialyzer will rise, causing a large ultrafiltration rate and minimized pressure. The alarm in the FCM will then cause the DFM to enter a patient-safe condition.

Operational Experience

It has been possible to use the algorithm to handle ultrafiltration control for all kinds of dialyzers that are available today. Treatment modes such as single-needle or double-needle treatment or sequential dialysis with periods of isolated ultrafiltration have been tested. Dialyzers with variations in values of B by a factor of 75 have been tested in the laboratory without any problems. After a period of approximately five months of clinical trials at several clinics, full-scale production started in the autumn of 1986. Over 11,000 units had been

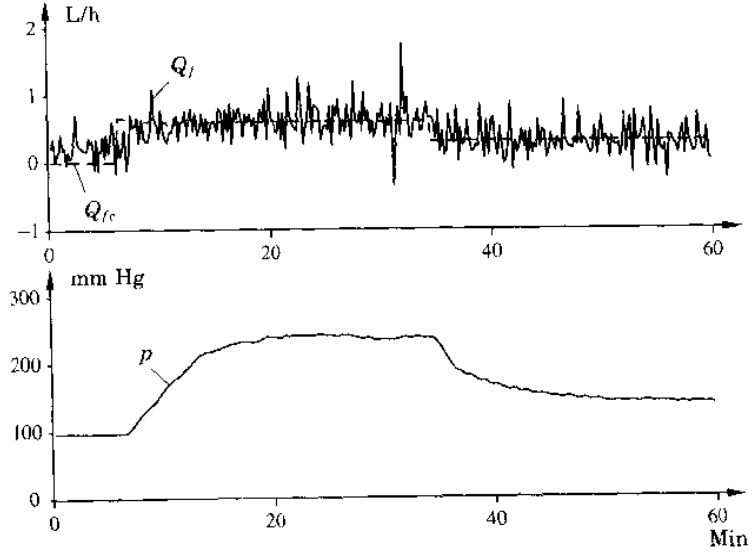

Figure 12.19 Adaptive control of a dialysis system. Responses in differential flow Q_f (solid line) and transmembrane pressure p to step changes in the setpoint Q_{fc} (dashed line) for a plate membrane. The adaptive control starts at $t = 6$. (With courtesy of Gambro AB.)

delivered as of December 1993. Since every machine may be used in several hundred treatments each year, there is now extensive practical experience with this algorithm, which seems to work well under all kinds of conditions.

Figure 12.19 shows responses in differential flow Q_f to step changes in the setpoint Q_{fc} when the system is under adaptive control. The pressure p is also shown. Despite the noisy flow measurement, the response of the closed-loop system is very good.

12.9 CONCLUSIONS

In this chapter we have tried to give an idea of how adaptive techniques are used in real control systems. A few general observations can be made.

Although there are many applications of adaptive control, it is clear that adaptive control is not a mature technology. The techniques were introduced in products in the early 1980s. Those in use today are mostly first-generation products; there are second-generation products in only a few cases.

The description of the products and the real applications show clearly that although the key principles are straightforward, many "fixes" must be done to make the system work well under all possible operating conditions. The need for safety nets, safety jackets, or supervision logic is not specific to adaptive control. Similar precautions must be taken in all real control systems, but since adaptive control systems are complex to start with, the safety nets that are required can be quite elaborate.

The examples clearly show that adaptive systems are not black box solutions that are a panacea. Rather, adaptive methods are useful in combination with other control design methods. Both in the rolling mill example and in the ship steering autopilot, adaptation was combined with gain scheduling. Another example is the use of a feedforward signal in the pulp dryer to improve the adaptation transient.

A third observation is that the human-machine interface is very important. A fourth observation is that some operating conditions are not conveniently handled by adaptive control. One example is the behavior of ship steering autopilots in a quartering sea.

There are unquestionably many different adaptive techniques, but so far, only a few of them have been used in industrial products. In many cases the choices have not been made by comparing several alternatives; one method has been chosen quite arbitrarily. This means that many alternatives have not been tried.

The computing power that is available has a significant influence on the type of control algorithms that can conveniently be implemented. The simple auto-tuners use simple 8-bit microprocessors, whereas some of the more advanced systems use full 32-bit architecture. In most process control applications there are no problems with computing time. The rolling mill applications, on the other hand, are quite demanding. The computing power that is available also has a significant impact on what human-machine interface can be implemented.

The applications also indicate the importance of the safety network. It is of interest to see the facilities provided in the toolbox and the specific solutions used in the dedicated systems. It is clearly much simpler to design a safety network for a dedicated system, in which good parameter bounds can be established.

The applications described in this chapter and elsewhere indicate that there are three cases in which it is very useful to use adaptive control:

- When the system has long time delays,
- When feedforward can be used, and
- When the character of the disturbances is changing.

In all these cases it is necessary to have a model of the process or the disturbances to effectively control the system. It is then beneficial to be able to estimate a model and to adapt to changes in the process.

REFERENCES

A number of applications are described in the books:

Narendra, K. S., and R. V. Monopoli, 1980. *Applications of Adaptive Control.* New York: Academic Press.

Unbehauen, H., ed., 1980. *Methods and Applications in Adaptive Control.* Berlin: Springer-Verlag.

Harris, C. J., and S. A. Billings, eds., 1981. *Self-Tuning and Adaptive Control: Theory and Applications.* London: Peter Peregrinus.

Narendra, K. S., ed., 1986. *Adaptive and Learning Systems: Theory and Applications.* New York: Plenum Press.

and in the survey papers:

Seborg, D. E., T. F. Edgar, and S. L. Shah, 1986. "Adaptive control strategies for process control: A survey." *AIChE Journal* **32**: 881–913.

Åström, K. J., 1987. "Adaptive feedback control." *Proc. IEEE* **75**: 185–217.

Proceedings of the IFAC, CDC, and ACC are also good reference sources. More details about the products are available in manuals, brochures, and application notes from the manufacturers.

The 3M PLC implementations is described in:

Alam, M. A., and K. K. Burhardt, 1979. "Further work on self-tuning regulators." *Proceedings of the 1979 IEEE Conference on Decision and Control,* pp. 616–620.

Alam, M. A., 1984. "A multivariable self-tuning controller for industrial application." *Preprints of the 9th IFAC World Congress*, pp. III:259–262. Budapest.

The Foxboro EXACT is described in:

Bristol, E. H., and T. W. Kraus, 1984. "Life with pattern adaptation." *Proceedings of the 1984 American Control Conference*, pp. 888–892. San Diego, Calif.

Kraus, T. W., and T. J. Myron, 1984. "Self-tuning PID controller uses pattern recognition approach." *Contr. Eng* June: 106–111.

The relay auto-tuning and the adaptive version are described in:

Åström, K. J., and T. Hägglund, 1984. "Automatic tuning of simple regulators with specifications on phase and amplitude margins." *Automatica* **20**: 645–651.

Åström, K. J., and T. Hägglund, 1988. *Automatic Tuning of PID Regulators.* Triangle Research Park, N.C.: Instrument Society of America.

Hägglund, T., and K. J. Åström, 1991. "Industrial adaptive controllers based on frequency response techniques." *Automatica* **27**: 599–609.

A survey of auto-tuners and adaptive PID controllers is given in:

Åström, K. J., T. Hägglund, C. C. Hang, and W. K. Ho, 1993. "Automatic tuning and adaptation for PID controllers: A survey." *Control Eng. Practice* **1**: 699-714.

The ABB adaptive controller/Master Piece is described in:

Bengtsson, G., and B. Egardt, 1984. "Experiences with self-tuning control in the process industry." *Proceedings of the 9th IFAC World Congress*, pp. XI:132–140. Budapest.

The description of the rolling mill example is based on:

Rudolph, W., H. Lefuel, and A. Rippel, 1984. "Regeln des Dressiergrades in einem Kaltbandwalzwerk." *Bänder Bleche Rohre* **25**(2): 36–37.

The description of the digester study is based on:

Allison, B. J., G. A. Dumont, L. H. Novak, and W. J. Cheetham, 1990. "Adaptive-predictive control of Kamyr digester chip level." *AIChE Journal* **36**(7): 1075–1086.

The adaptive systems used at Vallvik are described in:

Brattberg, Ö., 1994. "Adaptive control of a continuous digester." *Preprints of the Control Systems 94*, pp. 298–306. Swedish Pulp and Paper Research Institute, Stockholm.

The ship steering example is based on:

Källström, C. G., K. J. Åström, N. E. Thorell, J. Eriksson, and L. Sten, 1979. "Adaptive autopilots for tankers." *Automatica* **15**: 241–254.

Källström, C. G., P. Wessel, and S. Sjölander, 1988. "Roll reduction by rudder control." *Proceedings of the Spring Meeting, Society of Naval Architects and Marine Engineers*, pp. 67–76. Pittsburgh, Pa., June 8–10.

The particular control algorithm used in the product is described in:

Åström, K. J., 1980. "Design of fixed gain and adaptive ship steering autopilots based on the Nomoto model." *Proceedings of the Symposium on Ship Steering Automatic Control.* Instituto Internazionale delle Comunicazioni, June 25–27, Genoa, Italy.

The description of the control system for ultrafiltration is based on:

Sternby, J., 1995. "Adaptive control of ultrafiltration." Submitted, *Trans. Control Systems Technology* **3**.

CHAPTER 13

PERSPECTIVES ON ADAPTIVE CONTROL

13.1 INTRODUCTION

In this final chapter we attempt to give some perspective on the field of adaptive control. This is important but difficult because the field is in rapid development. The starting point is a short discussion of some closely related areas that are not covered in the book. These include adaptive signal processing in Section 13.2 and extremum control in Section 13.3. Particular attention is given to the field of adaptive signal processing, in which a cross-fertilization with adaptive control appears particularly natural.

Adaptive regulators and auto-tuning have complementary properties. Auto-tuners require very little prior information and give a robust ballpark estimate of gross system properties. Adaptive regulators require more prior knowledge, but they can give systems with much improved performance. It thus seems natural to combine auto-tuning with adaptive control in systems that combine several algorithms. Apart from algorithms for control, estimation, and design, it may also be useful to include supervision. It seems logical to use an expert system to monitor and control the operation of such a system. Systems of this type have been called *expert control systems* and are briefly discussed in Section 13.4. The use of expert systems also provides a natural way to separate algorithms from logic that occurs in all control systems.

Adaptation is related to learning; in Section 13.5 we discuss some early uses of learning in control systems and how it is related to adaptive control as we now understand it. In Section 13.6 we attempt to speculate on future directions in the theory and practice of adaptive control.

13.2 ADAPTIVE SIGNAL PROCESSING

Automatic control and signal processing have strong similarities; similar mathematical models and techniques are used in the two fields. However, there are also some significant differences. The time scales can be different. Signal processing often deals with rapidly varying signals, as in acoustics, in which sampling rates of tens of kilohertz are needed. In control applications it is often (but not always) possible to work with much slower sampling rates.

A more significant difference is that time delays play a minor role in signal processing. It is often permissible to delay a signal without any noticeable difficulty. Because control systems deal with feedback, even small time delays can result in drastic deterioration in performance. A third difference is in the industrial markets for the technologies. In signal processing, there are some standard problems that have a mass market, as in the field of telecommunications. The control market is more diversified and fragmented. Adaptive control is used to design control systems that work well in an unknown or changing environment. The environment is represented by process dynamics and disturbance signals. Adaptive signal processing is used to process signals whose characteristics are unknown or changing. More emphasis is given in signal processing to fast algorithms. Although there have been attempts to bring the fields closer together, much more effort is needed in this direction. To illustrate this, we will describe a few typical adaptive signal processing problems.

Prediction, Filtering, and Smoothing

Prediction, filtering, and smoothing are typical signal processing problems, which can all be described as follows: Given two signals x and y and a filter F, determine the filter such that the signals y and $\hat{y} = Fx$ are as close as possible. The problem can be illustrated by the block diagram in Fig. 13.1. In a typical case we have

$$x(t) = s(t) + v(t) \qquad \text{and} \qquad y(t) = s(t + \tau)$$

where s is the signal of interest and v is some undesirable disturbance. The problem is called *smoothing* if $\tau < 0$, *filtering* if $\tau = 0$, and *prediction* if $\tau > 0$. Solutions to such problems are well known for signals with known spectra and quadratic criteria. The corresponding adaptive problems are obtained when the

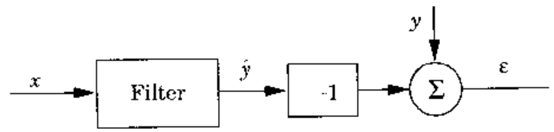

Figure 13.1 Illustration of filtering, prediction, and smoothing.

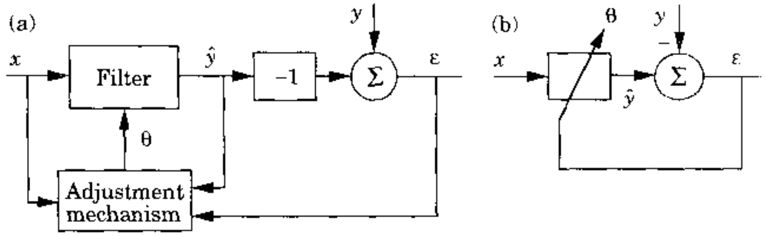

Figure 13.2 (a) An adaptive system for filtering, prediction, or smoothing and (b) its simplified representation.

signal properties are not known. All recursive parameter estimation methods can be applied to the adaptive signal processing problems. This is illustrated in Fig. 13.2, which gives a typical adaptive solution. The adjustment mechanism can be any recursive parameter estimator. The details depend on the structure of the filter and the particular estimation method chosen. An example illustrates the idea.

EXAMPLE 13.1 Output error parameter estimation

Assume that the filter is represented as an ordinary pulse transfer function

$$F(z) = \frac{b_0 z^{n-1} + b_1 z^{n-2} + \cdots + b_{n-1}}{z^n + a_1 z^{n-1} + \cdots + a_n}$$

To obtain a recursive estimator, the parameter vector

$$\theta = \begin{pmatrix} a_1 & \ldots & a_n & b_0 & \ldots & b_{n-1} \end{pmatrix}$$

and the regression vector

$$\varphi(t-1) = \begin{pmatrix} -\hat{y}(t-1) & \ldots & -\hat{y}(t-n) & x(t-1) & \ldots & x(t-n) \end{pmatrix}$$

are introduced. The error is then given by

$$\varepsilon(t) = y(t) - \hat{y}(t) = y(t) - \varphi^T(t-1)\hat{\theta}(t-1)$$

and the equation for updating the estimate is

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi(t-1)\varepsilon(t)$$

□

The special case of Example 13.1, obtained when the filter is an FIR filter and a gradient parameter estimation scheme is used, is particularly simple. This is the *LMS algorithm.*

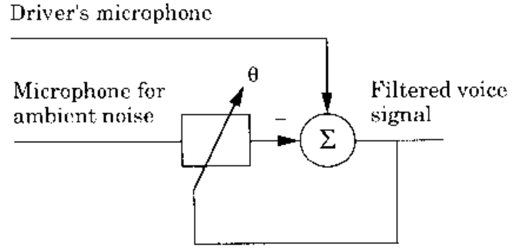

Figure 13.3 Use of an adaptive filter for adaptive noise cancellation.

Block Diagram Representation

The block diagram in Fig. 13.2 represents a solution to a generic signal processing problem. To make it easy to build large systems, it is convenient to consider this module as a building block that can be used for many different purposes. This is simpler if a proper representation is used. For that purpose it is convenient to represent the module as a block that receives signals x and y and delivers estimates $\hat{y}$ and $\hat{\theta}$. Such a representation, shown in Fig. 13.2(b), makes it possible to describe several adaptive signal processing problems.

Adaptive Noise Cancellation

Consider the situation of a mobile telephone in a car where there is a considerable ambient noise. Assume that two microphones are used. One is directional and picks up the driver's voice corrupted by noise; the other is directed away from the driver and picks up mostly the ambient noise. By connecting the microphones to an adaptive filter as shown in Fig. 13.3, it is possible to obtain a signal that is considerably improved. Removal of power frequency hum from measurement signals is another application at adaptive noise cancellation.

Adaptive Differential Pulse Code Modulation (ADPCM)

Digital signal transmission is becoming important because of the rapid development of new hardware. Its use in ordinary telephone communication is increasing. Pulse code modulation (PCM) is the standard method for converting analog signals to digital form. The analog signal is filtered and digitized by using an analog-to-digital (A-D) converter. The digitized signal is then transmitted in serial form. If the A-D converter has B bits and the sampling is f Hz, the transmission rate required is fB bits/s. For standard voice signals, a sampling rate of 8 kHz is typically used. A resolution of 12 bits in the A-D converter is required to get good-quality transmission. The bit rate required is thus 96 kbit/s. By having an A-D converter with a nonlinear characteristic it

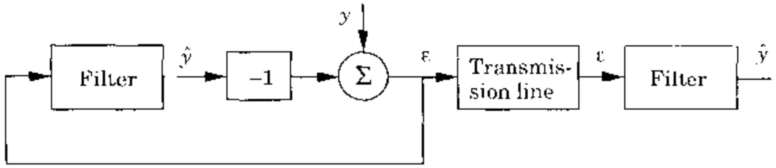

Figure 13.4 Block diagram of a differential pulse code modulation (DPCM) system.

is possible to reduce the bit rate to 64 kbit/s, which is the standard for digital voice transmission.

It is highly desirable to reduce the transmission rate, because more communication channels are then obtained with the same transmission equipment. The bit rate can be reduced significantly by using differential pulse code modulation (DPCM). In this technique the innovations of the signal are computed as $\varepsilon = y - \hat{y}$, where $\hat{y}$ is generated by filtering the innovations through a predictive filter. Only the innovations are transmitted (see Fig. 13.4). The receiver has a prediction filter with the same characteristics as the filter in the sender. The signal $\hat{y}$ can then be reconstructed in the receiver. The bit rate that is required is reduced significantly because fewer bits are required to represent the residual. It has been shown that for voice signals, a resolution of 4 bits is sufficient. This means that the bit rate required for the transmission can be reduced from 64 kbit to 32 kbit.

The prediction filter depends on the character of the transmitted signal. Substantial research into the characterization of speech has shown that it can be well predicted by linear filters. However, the properties of the filter will change with the particular sound that is spoken. To predict speech well, it is thus necessary to make the filters adaptive. The transmission scheme obtained is then called *adaptive differential pulse code modulation (ADPCM)*. Such a scheme, which uses an adaptive filter based on the output error method, is shown in Fig. 13.5. Notice that the adaptive filters at the transmitter and the receiver are driven by the residual only. If the filters in the receiver and the transmitter are identical, the filter parameters will automatically be the same. The adaptive filters have therefore been standardized by CCITT (*Comité Consultatif Internationale de Télégraphique et Téléphonique*). The filter that is used has the transfer function

$$H(z) = \frac{b_0 z^5 + b_1 z^4 + \cdots + b_5}{z^4(z^2 + a_1 z + a_2)}$$

The regression vector associated with the output error estimation is

$$\varphi(t) = \begin{pmatrix} -x(t) & -x(t-1) & e(t) & \ldots & e(t-5) \end{pmatrix}$$

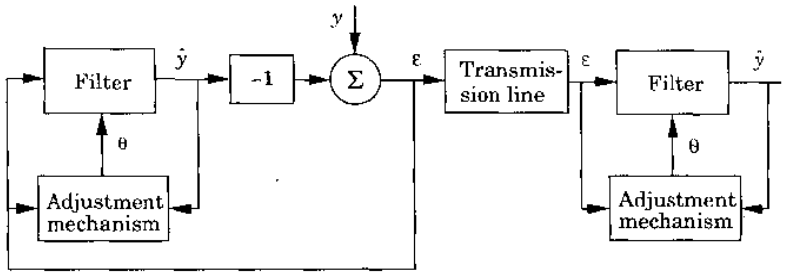

Figure 13.5 Block diagram of an adaptive differential pulse code modulation system.

and the associated parameter vector is

$$\theta = \begin{pmatrix} a_1 & a_2 & b_0 & \ldots & b_5 \end{pmatrix}$$

The standard least-squares estimator is of the form

$$\hat{\theta}(t+1) = \hat{\theta}(t) + P(t+1)\varphi(t)\varepsilon(t+1)$$

Several drastic modifications are made to simplify the calculations. A constant value of the gain is used. The multiplication is avoided by just using the signs of the signals. Leakage is also added to make sure that the estimator is stable. The updating of the parameters b_i is then given by the *sign-sign algorithm*

$$\hat{b}_i(t) = \left(1 - 2^{-8}\right)\hat{b}_i(t) + 2^{-7}\,\mathrm{sign}\,(e(t-i))\,\mathrm{sign}\,(e(t)) \qquad (13.1)$$

Similar approximations are made in the other equations. The computations in Eq. (13.1) are very simple. They can be done by shifts and the addition of a few bits, which can be accomplished with a small VLSI circuit. The CCITT ADCPM standard was achieved after significant experimentation. It is a good example of how drastic simplifications can be made with good engineering.

13.3 EXTREMUM CONTROL

The control strategies that have been discussed in the book have mainly been such that the reference value is assumed to be given. The reference value is often easily determined. It can be the desired altitude of an airplane, the desired concentration of a product, or the thickness at the output of a rolling mill. On other occasions it can be more difficult to find the suitable reference value or the best operating point of a process. For instance, the fuel consumption of a car depends, among other things, on the ignition angle. The mileage of the car

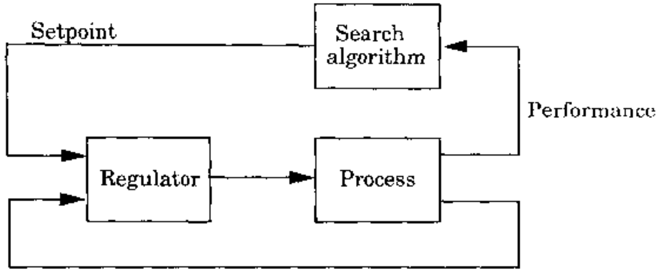

Figure 13.6 A simplified block diagram of an extremum control system.

can be improved by a proper adjustment, but the efficiency will depend on such conditions as the condition of the road and the load of the car. To maintain the optimal efficiency, it is necessary to change the ignition angle.

Tracking a varying maximum or minimum is called *extremum control*. The static response curve relating the inputs and the outputs in an extremum control system is nonlinear. The task of the controller is to find the optimum operating point and to track it if it is varying. Several processes have this kind of behavior. Control of the air-fuel ratio of combustion is one example. The optimum will change, for instance, with temperature and fuel quality. Another example is water turbines of the Kaplan type, in which the blade angle of the turbine is changed to give maximum output power. The same problem is encountered in wind power plants, in which pitch angle is changed depending on wind speed.

Extremum control is related to optimization techniques; many of the ideas have been transferred from numerical optimization. There was great interest in extremum control in the 1950s and 1960s, and some commercial products were put on the market. For instance, the first computer control systems installed in the process industry were motivated by the possibility of optimizing the setpoints of the controllers. The interest then declined, partly because of the difficulty of implementing the optimizing controllers. Furthermore, there is great difficulty in finding appropriate process models. Developments in computers have led to a renewed interest in extremum control and its combination with adaptive control. Improved efficiency of the process can result in large savings in the energy and raw material costs.

Figure 13.6 shows a simplified block diagram of an extremum control system. The process can work in open loop or in closed loop, as in the figure. The most important feature is that the process is assumed to be nonlinear in the sense that at least the performance is a nonlinear function of the reference signal. The goal of the search algorithm is to keep the output as close as possible to the extremum despite changes in the process or the influence of disturbances. The output used in the search algorithm is some measurement of the performance of the system—for instance, efficiency. The conventional

regulator can also use this signal, but it is more common for the regulator to use some other output of the process.

Models

Extremum control systems are, by necessity, nonlinear. How the processes are modeled is therefore all-important. Many investigations of extremum control systems assume that the systems are static. This assumption can be justified if the time between the changes in the reference value is sufficiently long. For static systems it is possible to use many of the methods from numerical optimization. A typical description of the process is

$$y(t) = f\left(u(t), \theta, t\right) \tag{13.2}$$

where f is a nonlinear function and θ is a vector of unknown parameters that may change with time.

If there are dynamics in the process, the performance may not have settled at a new steady-state value before the next measurement is taken. This will give an interaction in the control system that can be difficult to handle. The dynamic influence will increase the complexity considerably.

In many applications it is not easy to find the appropriate models and to determine the exact nature of the nonlinearities involved. It can therefore be appropriate to combine adaptivity and extremum control. One way to simplify the identification of an unknown nonlinear model is to assume that the process can be divided into one nonlinear static part and one linear dynamic part. Models with different properties are obtained if the nonlinearity precedes or follows the linear part. The complexity of the problem will also depend on which of the variables in the process can be measured. One special type of model that has been used in extremum control systems is the *Hammerstein model*. A typical discrete-time model of this type is

$$A(q)y(t) = B(q)f\left(u(t)\right) \tag{13.3}$$

where f is a nonlinear function, typically a polynomial.

The main effect of an input nonlinearity is that it restricts the possible input values for the linear part. The nonlinear control problem can then be treated as a linear control problem with input constraints. The case with the output nonlinearity perhaps leads to more realistic problems but is also more difficult to solve.

Extremum Control of Static Systems

The first extremum control systems were based on analog implementation. One way to perform the optimization is the so-called perturbation method. The basic idea is to add a known time-varying signal to the input of the nonlinearity, then observe the effect on the output and make a correlation between these two signals. Depending on the phase between the two signals, the direction toward the extremum can be determined. The perturbation method has been used for extremum control of chemical reactors, combustion engines, and gas furnaces, for instance.

Extremum control of static systems as in Eq. (13.2) is in essence a problem of numerical optimization. With the analog implementations the possible methods were severely restricted. When a digital computer is available, standard algorithms for function minimization can be used. Usually, it is possible only to measure the function values, not its derivative. The function minimization then has to be done by using numerically computed derivatives. Some methods use only function comparisons. These methods can be used even for minimization of nonsmooth functions.

Performance measurements are typically corrupted by noise. It is then necessary to average out the influence of the noise. This implies that the gain in the optimization algorithm should go to zero. However, if the extremum is changing with time, the gain should not go to zero. This is the same compromise as is discussed in connection with tuning and adaptive control.

Most schemes for extremum control of static systems do not build up any information about the nonlinearity. The "states" of the algorithms are essentially the current estimate of the optimum point and some previous measurements. By using a model and system identification it is possible to utilize the measurements of the system better and to follow time variations in the process.

Extremum Control of Dynamic Systems

If there are dynamics in the process, it is necessary to take this into consideration in doing the optimization. The correlation and interaction between different measurements of the performance will otherwise confuse the optimization routine. One possibility, discussed previously, is to wait until the transients have vanished before the next change is made. Of course, this will increase the convergence time, especially if the process has long time constants. One way around the problem is to base the optimization on nonlinear dynamic models. An example is the Hammerstein model. A model of this type with an input nonlinearity of second order is

$$A(q)y(t) = b_0 + B_1(q)u(t) + B_2(q)u^2(t) + C(q)e(t) \qquad (13.4)$$

The main reason for the popularity of the Hammerstein model is not that it is a good picture of the reality, but rather that it is linear in the parameters. The parameters can be estimated, for example, by using recursive least squares. The static response between the input and the output is given by

$$A(1)y_0 = b_0 + B_1(1)u_0 + B_2(1)u_0^2$$

The methods for static optimization discussed previously can now be used. Also note that the gradients and the Hessian are easily computed, a feature that will speed up the convergence.

Conclusions

The field of extremum control is far from mature. One crucial point is the modeling of the processes and the nonlinearities. It is generally very difficult to analyze nonlinear control problems and to derive optimal controllers, especially if there are stochastic disturbances acting on the system. The extremum control problem also has connections with the dual control problem discussed in Chapter 7. Extremum-seeking methods combined with adaptive control are of great practical interest, since even small improvements in the performance can lead to large savings in raw material and energy consumption. There are commercial extremum controllers.

13.4 EXPERT CONTROL SYSTEMS

All practical control systems contain heuristics. This appears as logic around the basic control algorithm. Adaptive systems have a lot of heuristics in the safety logic. Expert systems offer an interesting possibility of structuring the logic in a control system. If a good way to handle heuristic logic is available, it is also possible to introduce more complex control systems that contain several different algorithms. For example, it is possible to combine auto-tuners and adaptive algorithms that have complementary properties. The auto-tuner requires little prior information; it is very robust and can generate good parameters for a simple control law. Adaptive regulators can be more complex, with potentially better performance. Since they are based on local gradient procedures, they can adjust the regulator parameters to give a closed-loop system with very good performance, provided that reasonably good *a priori* guesses of system order, sampling period, and parameters are given. The algorithms will not work if the prior guesses are too far off. With poor prior data they may even give unstable closed-loop systems. This has led to the development of the safety logic discussed in Chapters 11 and 12.

Expert Systems

One objective of expert systems is to develop computer-based models for problem solving that are different from physical modeling and parameter estimation. An expert system attempts to model the knowledge and procedures used by a human expert in solving problems within a well-defined domain. Knowledge representation is a key issue in expert systems. Many different approaches have been attempted, such as first-order predicate calculus (logic), procedural representations, semantic networks, production systems or rules, and frames. A knowledge-based expert system consists of a knowledge base, an inference engine, and a user interface.

The Knowledge Base. The knowledge base consists of data and rules. The data can be separated into *facts* and *goals*. Examples of facts are statements such as "The system appears to be stable," "PI control is adequate," and "Deviations are normal." Typical examples of goals are "Minimize the variations of the output," "Find out whether gain scheduling is necessary," and "Find a scheduling table." Data is introduced into the database by the user or via the real-time knowledge acquisition system. New facts can also be created by the rules. The rule base contains production rules of the type: "*If* premise *then* conclusion *do* action." The premise represents facts or goals from the database. The conclusion can result in the addition of a new fact to the database or modification of an existing fact. The action can be to activate an algorithm for diagnosis, control, or estimation. These actions are different from those found in conventional expert systems. The rule base is often structured in groups or knowledge sources that contain rules about the same subject. This simplifies the search. In the control application the rules represent knowledge about the control and estimation problem that are built into the system. This includes the appropriate characterization of the algorithms, judgmental knowledge about when to apply them, and supervision and diagnosis of the system. The rules are introduced by the knowledge engineer via the knowledge acquisition system, which assists in writing and testing rules.

Inference Engine. The inference engine processes the rules to arrive at conclusions or to satisfy goals. It scans the rules according to a strategy, which decides from the context (current database of facts and goals) which production rules to select next. This can be done according to different strategies. In *forward chaining* the strategy is to find all conclusions from a given set of premises. This is typical for a data-driven operation. In *backward chaining* the rules are traced backward from a given goal to see whether the goal can be supported by the current premises. This is typical for a diagnosis problem. The search can be organized in many different ways, depth-first or breadth-first. There are also strategies that use the complexity of the rules to decide the order in which they are searched. To devise efficient search procedures, it is convenient to decompose the rule base into pieces that deal with related chunks of knowledge. If the rules are organized in that way, it is also possible for a system to focus its attention on a collection of rules in certain situations. This can make the search more efficient.

User Interface. The user interface can be divided into two parts. The first part is the development support that the system gives. This contains tools such as rule editor and rule browser for development of the system knowledge base. The other part is the run-time user interface. This contains explanation

facilities that make it possible to question how a certain fact was concluded, why a certain estimation algorithm is executing, and so on. It is also possible to trace the execution of the rules. The user interface can also contain facilities to deal with natural language.

Expert Control

The idea of expert control is to have a collection of algorithms for control, supervision, and adaptation that are orchestrated by an expert system. A block diagram of such a system is shown in Fig. 13.7. A comparison with Fig. 1.19 shows that the system is a natural extension of a self-tuning regulator. Instead of having one control algorithm and one estimation algorithm, the system has several algorithms. It also has algorithms for excitation and for diagnosis, as well as tables for storing data. Apart from this, the system also has an expert system, which decides when a particular algorithm should be used. The expert system contains knowledge about particular algorithms and the conditions under which they can be used.

In the special case in which there is only one algorithm of each category, Fig. 13.7 can be viewed as a well-structured way of implementing safety logic for an ordinary adaptive regulator. In that case the approach has the advantage that it separates the safety logic from the control algorithms. Another advantage is that the knowledge is explicit and can be investigated via the user interface.

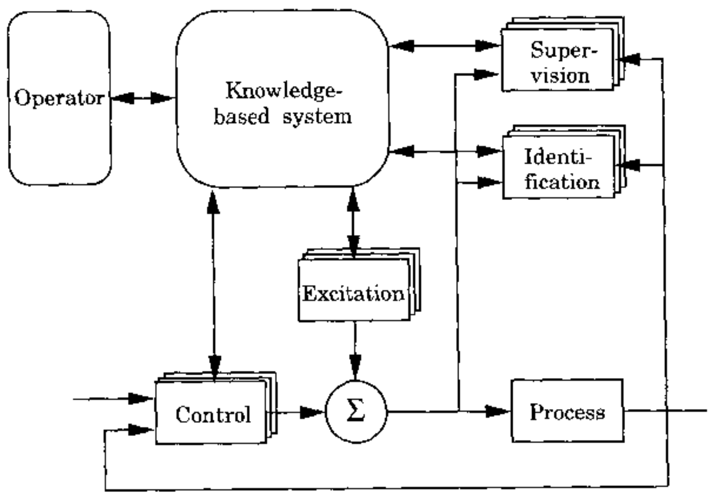

Figure 13.7 A knowledge-based expert control system.

13.5 LEARNING SYSTEMS

The notion of learning systems has been developed in the fields of artificial intelligence, cybernetics, and biology. In its most ambitious form, learning systems attempt to describe or mimic human learning ability. Attainment of this goal is still far away. The learning systems that have actually been implemented are simple systems that have strong relations to adaptive control. The systems have many names: neural nets, connectionist models, parallel distributed processing models, and so on.

Michie's Boxes

This system grew out of early work on artificial intelligence (see Michie and Chambers, 1968) and attempts to balance an inverted pendulum (see Fig. 13.8). The system has four state variables, φ, $\dot{\varphi}$, x, and $\dot{x}$, which are quantized in a crude way, with five levels for the position variables x and φ and three levels for the velocity variables $\dot{x}$ and $\dot{\varphi}$. The state space can thus be described by 225 discrete states. The control variable is quantized into two levels: force left (L) or force right (R). The control law can be represented by a binary table with 225 entries. In the experiment the table was initialized with randomly chosen L's and R's in the table. A simple scoring method was used to update the table entries as a result of experimental runs. Scoring was based on how long the pendulum stayed upright and the number of times the pendulum was in a discrete state. The system was able to balance the pendulum for about 25 minutes after a 60-hour training period. The table that defines the control action can be expressed in logic as:

> *If* cart *is* far left *and* cart *is* hardly moving *and* pendulum *is* hardly leaning *and* pendulum *is* swinging to right *then* apply force right.

For this reason the control law is also called *linguistic control*. When the logic is replaced by fuzzy logic, it is also called *fuzzy control*.

The training algorithm that is used in Michie's Boxes is similar to that used in programs for playing checkers and chess, but the pendulum problem is

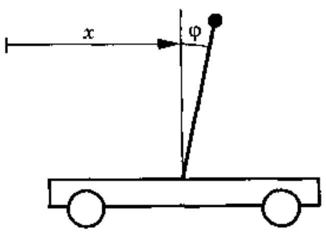

Figure 13.8 An inverted pendulum.

simpler than game playing. Training can be shortened by using a teacher, that is, by applying a scoring algorithm to an experiment in which the pendulum is balanced by an expert. A learning system of this type is obviously closely related to a model-reference adaptive system. The reference model can be viewed as a teacher.

The Perceptron

In a system such as Michie's Boxes the control law is a logic function that gives the control action as a function of sensor patterns. The function is adaptive in the sense that it will adjust itself automatically. The *perceptron* proposed by Rosenblatt (1962) is one way to obtain a learning function. To describe the perceptron, let u_i, $i = 1, 2, \ldots, n$, be inputs, and let y_i, $i = 1, 2, \ldots n$, be outputs. In the perceptron the output is formed as

$$y_i(t) = f\left(\sum_{j=1}^{n} w_{ij}(t)\, u_j(t) - b \right) \qquad i = 1, 2, \ldots, m \tag{13.5}$$

where w_{ij} are weights, b is a bias, and f is a threshold function, for example,

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

To update the weights, the perceptron uses a very simple idea, which is called *Hebb's principle*: Apply a given pattern to the inputs and clamp the outputs to the desired response, then increase the weights between nodes that are simultaneously excited.

This principle was formulated in Hebb (1949), in an attempt to model neuron networks. Mathematically it can be expressed as follows:

$$w_{ij}(t+1) = w_{ij}(t) + \gamma u_i(t) \left(y_j^0(t) - y_j(t) \right) \tag{13.6}$$

where y_j^0 is the desired response and y_j is the response predicted by the model Eq. (13.5). By regarding the weights as parameters, it becomes clear that the updating formula of Eq. (13.6) is identical to a gradient method for parameter estimation.

Widrow and Hoff (1960) developed special-purpose hardware, called the Adaline, to implement perceptronlike devices. The learning algorithm used by Widrow was based on a simple gradient algorithm like Eq. (13.6). In devices like the perceptron and the Adaline, learning is interpreted as adjusting the coefficients in a network. From this point of view it can equally well be claimed that an adaptive system like the MRAS or the STR is learning. The mechanisms for determining the parameters are also similar.

A drawback of the perceptron is that it can recognize only patterns that can be separated linearly. It fell into disfavor because of exaggerated claims that could not be justified. It was heavily criticized in a book by Minsky and Papert (1969). The idea of designing learning networks did, however, persist.

The Boltzmann Machine

The Boltzmann Machine may be viewed as a generalization of a perceptron. It was designed to be a highly simplified model of a neural network. The machine consists of a collection of elements whose outputs are zero or one. The elements are linked by connections having different weights. The output of an element is determined by the outputs of the connecting elements and the weights of the interconnections. The firing is randomized in such a way that the probability of firing increases with the weighted sum of the inputs to an element. Some elements are connected to inputs and others to outputs, and there are also internal nodes. The connections in a Boltzmann Machine are assumed to be symmetric, which is a significant restriction.

In the perceptron there is a direct coupling between the inputs and the output. The Boltzmann Machine is much more complicated, because it can also have internal nodes. This implies that Hebb's principle cannot be applied directly. An extension called *back-propagation* has been suggested in Rumelhart and McClelland (1986).

There are many variations of neural networks. Dynamics can be introduced in the nodes. Hopfield observed that the weights could be chosen so that the network would solve specific optimization problems (Hopfield and Tank, 1986).

Hardware

An interesting feature of the neural networks is that they operate in parallel and that they can be implemented in silicon. Using such circuits may be a new way to implement adaptive control systems. A particularly interesting feature is that it is easy to integrate the networks with sensors.

13.6 FUTURE TRENDS

In this section we speculate on open research issues and the future of adaptive control. One interesting aspect of adaptive control is that it may be viewed as an automation of the modeling and design of control systems. To apply a technique automatically, it is necessary to have a very clear understanding of the conditions under which it can be applied. Ideally, this understanding should be formalized. Research on adaptive control will thus sharpen the understanding of control and parameter estimation.

Industrial Impact

There are adaptive systems that have been in continuous operation since the mid-1970s. Several products were announced in the early 1980s, and the development has accelerated since that time. Adaptation is used both in general-

purpose controllers and in dedicated systems. Practically all PID controllers that are introduced today have some facility for tuning and adaptation. This applies even to simple temperature controllers. It has taken longer for adaptive techniques to appear in distributed systems for process control, but many distributed control systems are now provided with tuning and adaptation. There is a rich variety of special-purpose systems that use adaptation. For example, it has been shown that adaptation can provide improved riding quality in cars.

In summary, many adaptive algorithms are well understood. Our insight into how adaptive methods can be used to engineer better control systems is growing. Insight, understanding, and appropriate computing hardware are available. It seems likely that a large proportion of the control systems made in the future will have automatic tuning or adaptation. When adaptive control becomes more widely used, interesting phenomena that demand theoretical understanding will undoubtedly also be observed. For instance, what happens when many adaptive controllers are connected to one process? Will they interact? How should the system be initialized? We can thus look forward to interesting developments.

Algorithm Development

There are several important issues that relate to algorithm development. Current toolboxes for adaptive control use only a few of the algorithms that have been developed. It seems safe to guess that the toolboxes will be expanded, and it would also seem useful to include auto-tuners in the toolboxes to simplify initialization. Significant improvements can thus be achieved with tools that are already known, but there is also a need for improved techniques. Better methods for control system design are needed. Techniques that can explicitly handle actuator constraints and model uncertainties would be valuable contributions. It would be very useful to have methods for estimating the unstructured uncertainties.

Diagnostic routines that will tell whether a control algorithm is behaving as expected are needed. Such algorithms are well known for minimum-variance control, in which monitoring can be done simply by calculating covariances. It is straightforward to develop similar techniques for other design methods.

There is both theoretical and experimental evidence that probing signals are useful. It is also clear that it is not practical to introduce probing via stochastic control theory because of the excessive computational requirements. A significant challenge is therefore to find other ways to introduce probing. There are many who intuitively object to introducing probing signals intentionally. It must be remembered that a poorly tuned regulator will give larger than necessary deviations in controlled variables.

A systematic approach to design and implementation of safety networks is an issue of great practical relevance. Expert systems may be useful in this context.

Multivariable Adaptive Control

In this book we have focused on single-input, single-output systems, mainly to keep the presentation simple, but there has also been much research on multivariable adaptive control. Many of the results can be extended, but there is one large difficulty. For single-input, single-output systems it is possible to find a good canonical form to represent the systems in which the only parameter is the order of the system. For multivariable systems it is necessary also to know the Kronecker indices to obtain a canonical form. This is difficult both in theory and in practice. For special systems such as those found in robotics, a suitable structure can often be found by using prior knowledge of the system.

Most adaptive control systems used so far are single-loop control. Coupled systems can be obtained by interconnection via the feedforward connection. Interesting phenomena can occur when such regulators are used on multivariable systems; analysis of the behavior of such systems is a fascinating problem.

Theoretical Issues

There are many unresolved theoretical problems in adaptive control. For example, we have no good results on the stability of schemes with gain scheduling. Much work is also needed on analysis of convergence rates. On a very fundamental level, there is a need for better averaging theorems. Many results apply only to periodic signals. This is natural, since the theory was originally developed for nonlinear oscillations. It would be highly desirable to have results for more general signal classes.

Several important problems have arisen in applications. The most important one is the design of proper safety logic; this is currently done in an ad hoc fashion. The development is also hampered by the fact that much of the information is proprietary, for competitive reasons.

13.7 CONCLUSIONS

In this book we have attempted to give our view of the complex field of adaptive control. There are many unresolved research issues and many white spots on the map of adaptive control. The field is developing rapidly, and new ideas are continually popping up.

Our opinion is that adaptive control is a good tool that a control engineer can use on many occasions. We hope that this book will help to spread the use of adaptive control and that it may inspire some of you to do research that will enhance our understanding of adaptive systems.

REFERENCES

There is an extensive literature on adaptive signal processing. A good treatment is given in:

Widrow, B., and S. D. Stearns, 1985. *Adaptive Signal Processing.* Englewood Cliffs, N.J.: Prentice-Hall.

There are strong international efforts by the IFAC and the IEEE to connect the fields of adaptive control and adaptive signal processing. A student would be well advised to pay attention to this effort. Adaptive filters are discussed in:

Treichler, J. R., C. R. Johnson, Jr., and M. G. Larimore, 1987. *Theory and Design of Adaptive Filters.* New York: John Wiley & Sons.

The CCITT standard on adaptive differential pulse code modulation is described in:

Jayant, N. S., and P. Noll, 1984. *Digital Coding of Waveforms: Principles and Applications to Speech and Video.* Englewood Cliffs, N.J.: Prentice-Hall.

Optimalizing control was introduced in:

Draper, C. S., and Y. T. Li, 1966. "Principles of optimalizing control systems and an application to the internal combustion engine." In *Optimal and Self-optimizing Control,* ed. R. Oldenburger. Cambridge, Mass.: MIT Press.

Extremum control problems are discussed in:

Blackman, P. F., 1962. "Extremum-seeking regulators." In *An Exposition of Adaptive Control,* ed. J. H. Westcott. Oxford, U.K.: Pergamon Press.

Sternby, J., 1980. "Extremum control systems: An area for adaptive control?" *Preprints of the Joint American Control Conference, JACC.* San Francisco. Paper WA2-A.

Lachmann, K.-H., 1982. "Parameter adaptive control of a class of nonlinear processes." *Proceedings of the 6th IFAC Symposium on Identification and System Parameter,* pp. 372–378, Washington, D.C.

Wittenmark, B., 1993. "Adaptive control of a stochastic nonlinear system: An example." *Int. J. Adapt. Control and Signal Processing* 7: 327–337.

The notion of expert control was introduced in:

Åström, K. J., J. J. Anton, and K. E. Årzén, 1986. "Expert control." *Automatica* **22**(3): 277–286.

A detailed description of a system based on this idea is given in:

Årzén, K.-E., 1987. "Realization of expert system based feedback control." Ph.D. thesis TFRT-1029, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Good sources for knowledge about expert systems are:

Barr, A., and E. A. Feigenbaum, eds., 1982. *The Handbook of Artificial Intelligence.* Los Altos, Calif.: William Kaufmann.

Hayes-Roth, F., D. Watermann, and D. Lenat, 1983. *Building Expert Systems*. Reading, Mass.: Addison-Wesley.

The program Boxes is described in:

Michie, D., and R. Chambers, 1968. "Boxes: An experiment in adaptive control." In *Proceedings of the 2nd Machine Intelligence Workshop*, eds. Dale, E., and D. Michie, pp. 137–152. Edinburgh, U.K.: Edinburgh University Press.

Fuzzy logic was introduced in:

Zadeh, L. A., 1973. "Outline of a new approach to the analysis of complex systems and decision processes." *IEEE Trans. Systems, Man and Cybernetics* **SMC-3**: 28–44.

Early examples of learning systems are given in:

Fu, K. S., 1968. *Sequential Methods in Pattern Recognition and Machine Learning*. New York: Academic Press.

Saridis, G. N., 1977. *Self-organizing Control of Stochastic Systems*. New York: Marcel Dekker.

The perceptron is described in:

Rosenblatt, F., 1962. *Principles of Neurodynamics*. New York: Spartan Books.

A critique of the perceptron is given in:

Minsky, M., and S. Papert, 1969. *Perceptrons: An Introduction to Computational Geometry*. Cambridge, Mass.: MIT Press.

Hebb's principle for adjusting the weights in a neural network is described in:

Hebb, D. O., 1949. *The Organization of Behavior*. New York: Wiley.

The Adaline is described in:

Widrow, B., and M. Hoff, 1960. "Adaptive switching circuits." *IRE WESCON Convention Record*, pp. 96–104, Pt 4.

This system was applied to many problems, such as how to stabilize an inverted pendulum. Examples of neural networks and some of their uses are found in:

Kohonen, T., 1984. *Self-organization and Associative Memory*. Berlin: Springer-Verlag.

Grossberg, S., 1986. *The Adaptive Brain. I: Cognition, Learning, Reinforcement, and Rhythm*, and *The Adaptive Brain. II: Vision, Speech, Language, and Motor Control*. Amsterdam: Elsevier/North-Holland.

Rumelhart, D. E., and J. L. McClelland, 1986. *Parallel Distributed Processing*, Vols. 1 and 2. Cambridge, Mass.: MIT Press.

These books contain many references and a detailed treatment of the Boltzmann machine. A spectacular application of the Boltzmann machine is given in:

Sejnowski, T., and C. R. Rosenberg, 1986. "NETtalk: A parallel network that learns to read aloud." JHU-EECS-8601, Johns Hopkins University.

Hopfield's network is described in:

Hopfield, J. J., and D. W. Tank, 1986. "Computing with neural circuits: A model." *Science* **233**: 625–633.

Methods for implementing neural networks in silicon and integrating them with sensors are found in:

Hecht-Nielsen, R., 1989. *The Technology of Non-Algorithmic Information Processing*. Reading, Mass.: Addison-Wesley.

Mead, C. A., 1989. *Analog VLSI and Neural Systems*. Reading, Mass.: Addison-Wesley.

Ideas on how to combine adaptation with neural networks and other techniques in control systems are discussed in:

White, D. A., and D. A. Sofge, 1992. *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*. New York: Van Nostrand, Reinhold.

INDEX

A
a priori knowledge, 241
ABB adaptive controller, 509
ABB Master, 509
Ackermann, J., 13, 40
Adaline, 558, 563
adaptive autopilot for ship, 32, 529
adaptive control
  abuses of, 30, 502
  applications of, 27, 499
  definition of, 1
  history of, 2
  implementation of, 448
  robust, 327
  stochastic, 348
  theory, 263
adaptive control problem, 24
adaptive feedforward, 160
adaptive filtering, 57, 546
adaptive industrial controllers, 503
adaptive noise cancellation, 548
adaptive prediction, 546
adaptive predictive control, 168
adaptive signal processing, 546
adaptive smoothing, 546
ADPCM, 548, 549
Alam, M. A., 542, 543
aliasing, 451
Allison, B. J., 543
Alster, J., 373
Amsle, B. E., 446
Anderson, B. D. O., 39, 346
Annaswamy, A. M., 39, 261, 344, 346
anti-aliasing filter, 451
  delay due to, 452
anti-windup, 376, 455
Anton, J. J., 562
Arnold, V. I., 345
Aryabhatta equation, 94
Årzén, K.-E., 562
Asea Brown Boveri, 505, 509
Ash, R. H., 389
associated differential equation, 320
Åström, K. J., 38, 39, 135, 136, 165, 181, 182, 184, 333, 345, 347, 358, 366, 372, 373, 374, 389, 418, 446, 497, 498, 542, 543, 544, 562
Athans, M., 40, 345
Atherton, D. P., 446
augmented error, 233, 238
autoregressive, 58
auto-tuning, 27, 375, 501, 506

averaging, 299
  example, 306
  stochastic, 319

B
backstepping, 249
Barbalat's lemma, 205
Barnett, S., 184
Barr, A., 562
Bar-Shalom, Y., 372, 374
Bélanger, P. R., 373
Bellman, R., 2, 38, 345, 372
Bellman equation, 23, 349, 359
Bengtsson, G., 543
Bertsekas, D., 372
Bessel filter, 452
Bezout identity, 94
BIBO stability, 215, 218
Bierman, G. J., 88, 498
bifurcations, 267
Billings, S. A., 39, 261, 497, 542
Bitmead, R. R., 39, 183, 346
Blackman, P. F., 562
Blankinship, W. A., 135
Bode, H. W., 40, 445
Bodson, M., 39, 88, 261, 345
Bogoliubov, N. N., 345
Bohlin, T., 373
Böhm, J., 184
Boltzmann machine, 559
Borisson, U., 182
bounded-input, bounded-output (BIBO) stability, 215, 218
Boxes, 557
Brattberg, Ö., 543
Braun, L., 38
Bristol, E. H., 543
Brockett, R. W., 417
BST Control, 504
Burhardt, K. K., 542
bursting, 272
Butchart, R. L., 260
Butterworth filter, 452

C
Cadorin, D., 371, 374
Caines, P. E., 344
CARIMA model, 174
cautious control, 356, 357, 362
certainty equivalence controller, 348, 356
certainty equivalence principle, 22, 91, 360
Chambers, R., 557, 563
chaos, 272
chattering, 436
Cheetham, W. J., 543
chemical reactor control, 518
Chen, H.-F., 345
Clark, D., 346
Clarke, D. W., 181, 182, 183, 184, 497
closed-loop estimation, 69, 466
Cohen, G. H., 389
Collins, W. D., 447
combustion control, 411
computational delay, 449
computing power, 515
concentration control, 11, 396
conditional updating, 331, 477
connectionist model, 557
consistency, 48
constant-trace algorithms, 477
continuous-time self-tuners, 109
control design, 92, 137, 458, 506, 508, 511, 514, 537
controller structures, 515
convergence, 48, 263
Coon, G. A., 389
covariance resetting, 52, 286, 471
Cutler, C. R., 183

D
Dale, E., 563
Datta, A., 38
De Keyser, R. M. C., 183
dead zone, 328, 477
  in estimator, 331

Decaulne, P., 446
decrescent, 205
delta operator, 333
describing function, 382
Desoer, C. A., 262
Desphande, P. B., 389
detuned minimum-variance
control, 158
dialysis, 534
differential operator, 25
Diophantine equation, 94
solution methods, 462
direct adaptive control, 25, 91
direct self-tuning regulator, 112, 126
algorithm of, 149
directional forgetting, 478
discounting factor, 53
distillation column, 517
dither signal, 433
Dorato, P., 38
Doyle, J. C., 40, 445
DPCM, 549
Draper, C. S., 562
Draper, N. R., 87
dual control, 22, 24, 349, 354, 360
suboptimal, 362
dual-input describing function,
428, 429
Dugard, L., 182
Dumont, G. A., 374, 497, 543
dynamic matrix control, 168
dynamic programming, 351, 358

E

Edgar, T. F., 38, 542
Edmunds, J. M., 87, 136, 182
EEAS, 435
Egardt, B., 136, 261, 262, 329,
331, 332, 344, 543
Elevitch, C., 369, 374
Elliott, H., 182
ELS, 62
Emelyanov, S. V., 446
equation error, 59
Eriksson, J., 418, 543
error augmentation, 238
error model, 236
estimation, 41
closed-loop, 69
continuous-time model, 55, 60
stochastic model, 61
estimator implementation, 465
estimator windup, 473
Euclid's algorithm, 463
Eurotherm, 509
EXACT, 508
excitation, 44, 64
experimental condition, 63
expert control, 545, 554, 556
expert systems, 554
explicit self-tuning control, 91
exponential discounting, 53
exponential forgetting, 53, 468
extended horizon control, 168
extended least squares, 62
extended space, 216
externally excited adaptive
system, 435
extremum control, 550, 551

F

feedback linearization, 246
feedforward control, 29, 118, 160
Feigenbaum, E. A., 562
Feldbaum, A. A., 373
Feuer, A., 344
Filippov, A. F., 437
filtering, 113, 156, 237, 451
finite-impulse response model, 56
FIR model, 57
First Control Systems, 505, 512, 524
Firstline, 513
Firstloop, 512
Fisher Control, 506
Fisher-Rosemont, 504
flight control, 12, 415
flight dynamics, 12
Floquet theory, 276

Flügge-Lotz, I., 426
forgetting factor, 53
Fortescue, T. R., 88
forward shift operator, 25
Foxboro, 508
Francis, B. A., 40
Fu, K. S., 563
fuel-air control, 412, 528
fuzzy control, 557

G

gain changer, 433
gain limitation, 328
gain scheduling, 19, 28, 390, 501
applications of, 402
design of, 392
Gambro AB, 534
Garcia, C. E., 183
Gauss, K. F., 42, 88
Gawthrop, P. J., 39, 136, 181, 344
Gelb, A., 445
generalized minimum-variance
controller, 158
generalized predictive control
(GPC), 168, 174
Gentleman, W. M., 498
Gevers, M., 183
Gille, J. C., 446
global stability, 293
Gold, B., 498
Goodwin, G. C., 39, 87, 136, 182,
261, 262, 331, 344, 346, 371,
374, 497, 498
Gorozdos, R. E., 446
Gough, N. E., 447
GPC, 174
gradient algorithm, 281
gradient method, 186
Greco, C., 184
Gregory, P. C., 38, 446
Grimble, M. J., 184
Gronwall-Bellman lemma, 298
Grossberg, S., 563
Guckenheimer, J., 343, 345
Gunnarsson, S., 89, 498
Guo, L., 345
Gupta, M. M., 38
Gustafsson, I., 87

H

Hagander, P., 347
Hägglund, T., 88, 389, 446, 498, 543
Hahn, W., 262
Hale, J. K., 345
Halousková, A., 184
Hammerstein model, 552, 553
Hang, C. C., 261, 389, 543
Hanson, R. J., 88
harmonic balance, 382
Harris, C. J., 39, 261, 447, 497, 542
Hartmann & Braun, 503
Hayes-Roth, F., 563
Haykins, S., 89
Hebb, D. O., 558, 563
Hebb's principle, 558
Hecht-Nielsen, R., 564
Helmersson, A., 366, 373
high-frequency gain, 236, 239, 334
high-gain feedback, 4, 419, 427
Hill, D. J., 346, 497
Ho, W. K., 389, 543
Hoff, M., 558, 563
Holmes, P., 343, 345
Holst, J., 322, 347
Honeywell, 503, 504
Hopfield, J. J., 559, 564
Hopfield network, 559
Horowitz, I. M., 40, 419, 420,
445, 446
Householder transformation, 481
Hughes, D. J., 374
human–machine interface, 531
Hunt, L. R., 417
Hyde Marine Systems, 529, 533
hyperbolic fixed point, 272
hyperstate, 23, 349, 355

I
implementation, 448
implicit self-tuning control, 91
indirect adaptive control, 26, 91
indirect deterministic pole placement
  self-tuner, 103
indirect self-tuning regulator, 102
industrial adaptive controllers, 503
industrial experiences, 507, 509, 514
initialization, 493
innovation model, 166
innovations representation, 139
input-output stability, 218
input strictly passive (ISP), 220
instantaneous gain, 236
integrator windup, 455
Intelligent Tuner, 504
internal model principle, 122, 165
Ioannou, P. A., 38, 346
Irving, E., 88
Isermann, R., 39, 497
Isidori, A., 417
Ismail, Z. M., 447
ISP, 220
ITAE filter, 452
Itkis, U., 446

J
Jacobs, O. L. R., 39, 373, 374
Jayant, N. S., 562
Ježek, J., 135
Johansson, R., 87, 182, 327, 347
Johnson, C. R., Jr., 39, 346, 562

K
Kaczmarz, S., 88
Kaczmarz's algorithm, 53, 54
Källström, C. G., 418, 543
Kalman, R. E., 38, 136, 181
Kalman-Yakubovich lemma,
  213, 223, 262
Kanellakopoulos, I., 262
Kanjilal, P. P., 184
Kárný, M., 88, 184
Kershenbaum, L. S., 88
Kezer, A., 260
Khalil, H. K., 262
King, R. E., 447
Kockum Sonics AB, 529
Kohonen, T., 563
Koivo, H., 182
Kokotovic, P. V., 39, 262, 346
Kosut, R. L., 39, 346
Kraus, T. W., 543
Kreisselmeier, G., 346
Krener, A. J., 417
Krstić, M., 262
Krylov, A. N., 345
Kučera, V., 135, 498
Kulhavý, R., 88, 184, 498
Kumar, P. R., 38, 182, 344, 347, 372
Kushner, H., 346
Kwon, W. H., 183

L
Lachmann, K.-H., 39, 497, 562
lambda sond, 413
Landau, Y. D., 260, 262, 347
Larimore, M. G., 562
Larminat, P. de, 344
Lawson, C. L., 88
leakage, 328, 479
learning, 557
least mean-squares-algorithm
  (LMS), 55, 547
least-squares estimation, 42, 43
  geometric interpretation, 46
  normal equation, 44
  recursive, 51
  recursive computations, 49
  statistical interpretation, 47
Leeds and Northrup, 503
Lefschetz, S., 262
Lefuel, H., 543
Lenat, D., 563
Lennartson, B., 497
level control, 507
Li, W., 262, 446

Li, Y. T., 562
limit cycle oscillation, 428
Lin, Y.-H., 261, 344
linear quadratic Gaussian control,
  139, 145
linear quadratic self-tuning
  regulator, 164
linguistic control, 557
Ljung, L., 87, 89, 262, 320, 322,
  346, 347, 498
LMS algorithm, 55, 547
Long, R. S., 182
loop transfer recovery, 420
Looptune, 504
Lozier, J. C., 445
Lyapunov, A. M., 199
Lyapunov function, 201
Lyapunov stability theorem, 201, 204
Lyapunov theory, 199

M
MacMillan Bloedel, 525
Manfredi, C., 184
Mareels, I. M. Y., 39, 346
Marshall, J. E., 447
Mårtensson, B., 338, 347
Matko, D., 39
matrix inversion lemma, 50
Mayne, D. Q., 183, 261, 346, 497
McAvoy, T., 445
McClelland, J. L., 559, 563
McMillan, G. K., 389
Mead, C. A., 564
Menga, G., 184
Metivier, M., 347
Meyer, G., 417
Michalska, H., 183
Michie, D., 557, 563
Middleton, R. H., 182, 346, 497
Milito, R., 371, 374
Millman, R. S., 417
minimum-degree pole placement, 96
minimum-degree solution, 95
minimum-variance control, 137, 140
  detuned, 158
Minorsky, N., 345
Minsky, M., 558, 563
Mishkin, E., 38
MIT rule, 20, 186
model-following, 94, 101, 427
model-following condition, 94
model predictive control, 168
model-reference adaptive systems
  (MRAS), 20, 185
  design using Lyapunov theory, 206
  for a first-order system, 190
  general linear system, 235
MoDo Chemetics, 525
Mohtadi, C., 183, 184
Molusis, J. A., 374
monic, 93
monitoring of excitation, 329
Monopoli, R. V., 38, 39, 88, 234,
  261, 417, 542
Mookerjee, P., 374
Morari, M., 40, 183, 445
Morse, A. S., 261, 335, 344, 347
Mosca, E., 183, 184, 374
moving-average control, 137,
  143, 144
MRAS, 20, 185
multivariable control, 516
Muske, K. R., 183
myopic controller, 353
Myron, T. J., 543

N
Narendra, K. S., 38, 39, 88, 261,
  344, 346, 417, 542
Nedoma, P., 184
neural net, 557
Nichols, N. B., 389
Niemi, A. J., 417, 418
Nijmeijer, H., 417
Noll, P., 562
Nomoto model, 404
nonlinear actuators, 10, 392, 393
nonlinear transformation, 392, 398

nonminimum-phase system, 118
normal equation, 44, 56
normalization, 332
Norton, J. P., 87
Novak, W. H., 543
Novatune, 33, 509
Nussbaum, R. D., 335, 347

O
observer polynomial, 100
ODE approach, 320
Oldenburger, R., 562
one-button-tuning, 375, 506
one-step head controller, 357
open-loop optimal feedback
  control, 350
operational experience, 532, 540
operator view, 216
Orava, P. J., 417
Ortega, R., 38, 346
orthogonal transformation, 481
Osburn, P. V., 260
OSP, 220
output error, 59
output strictly passive (OSP), 220
overfitting, 45
overparameterization, 290, 302
Owens, D. H., 447

P
Padilla, C. S., 371, 374
Padilla, R. A., 371, 374
Papert, S., 558, 563
Papon, J., 183
Paprican, 525, 526
parameter convergence, 193
parameter equilibrium, 302, 307
parameter estimation, 41, 465,
  506, 508, 511, 514, 537
parametric excitation, 279
Parks, P. C., 260, 261
Parseval's theorem, 66
passivity, 220
Patchell, J. W., 373
Payne, R. L., 371, 374
PE, 64
Pearson, A. E., 183
Pelegrin, M. J., 446
perceptron, 558
perfect model-following, 94
persistent excitation (PE), 63,
  64, 67, 285
perturbation signals, 515
Peterka, V., 136, 181, 184, 498
Peterson, B. B., 344
pH control, 406
PID controller, 376
PIDWIZ, 504
Plackett, R. L., 88
pole excess, 93, 246
pole placement, 92
Popov, V. M., 262
positive real, 213
  test, 225
positive (semi) definite function, 200
postsampling filter, 454
Pott, D. C., 417
PR, 213
practical aspects, 448, 538
Prager, D., 136, 182
Praly, L., 39, 344, 346
predictive controller, 137
presampling filter, 451
pre-tuning, 375, 385, 509, 516
prior information, 78, 507, 509
Priouret, P., 347
probing, 368
Process Automation Systems, 505
projection, 328
projection algorithm, 53, 54, 281
Protonic, 503
prototype algorithm, 490
Protuner, 504
pulp dryer control, 520

Q
QR transformation, 481

R
Rabiner, L. R., 498
Ramadge, P. J., 262, 344
Ramaker, B. C., 183
Rao, G. P., 88
Rault, A., 183
Rawlings, J. B., 183
real-time estimation, 41
receding-horizon controller, 168
reciprocal polynomial, 93
recursive estimation, simulation, 71
recursive least-squares estimation,
  51
recursive maximum likelihood, 62
regression model, 43
regression variable, 42
regressor, 42
regularized constant-trace
  algorithm, 478
relatively prime, 93
relay feedback, 380
relay oscillation, 382, 385
relay systems, 426
Ren, W., 182
reset action, 121, 488
reset windup, 455
resetting, 52
residual, 43
Richalet, J. A., 183
Riedle, B. D., 39, 346
Rippel, A., 543
RLS, 51
RML, 62
robot control, 421
robust control, 3, 4, 306
robust design methods, 419
robust estimation, 480
robust high-gain control, 4
robustness, 263, 327
Rocchi, S., 374
Rohrs, C., 345
Roll-Nix, 529, 532
rolling mill, control of, 521
Rosenberg, C. R., 563
Rosenblatt, F., 558, 563
Rudolph, W., 543
Rumelhart, D. E., 559, 563

S
safety jackets, 541
safety nets, 541
safety network, 514
sampling interval, choice of, 451
Sandberg, I. W., 262
Saridis, G. N., 562
Sastry, S., 39, 88, 261, 345
SattControl, 31, 503, 506
Scattolini, R., 183
Schuck, O. H., 446
Schwartz, A., 346
Seborg, D. E., 38, 542
Sejnowski, T., 563
self-organizing control (SOC), 1
self-oscillating adaptive systems
  (SOAS), 419, 426
  design procedure, 432
  main result, 429
  principle of, 427
self-tuning feedforward, 160
self-tuning regulator, 21, 90, 137
  asymptotic properties, 151
  linear quadratic Gaussian, 164
  stochastic, 137, 146
sensitivity derivative, 187
separation, 360
setpoint limits, 538
Shackcloth, B., 260
Shah, S. L., 38, 497, 542
Shaked, U., 445
Shapiro, A., 446
Shinskey, F. G., 389
ship steering, 15, 402, 529
ship steering autopilot, 529
Sidi, M., 445
sign-sign algorithm, 187, 550
Sin, K. S., 39, 87, 136, 182,
  331, 344, 498
Singh, R. P., 261, 344
sinusoidal perturbations, 301

Sjölander, S., 543
sliding mode, 436
Slotine, J.-J. E., 262, 446
slow perturbations, 301
slowly varying parameters, 52
small gain theorem, 219
Smay, J. W., 446
Smith, C. L., 389
Smith, H., 87
SOAS, 419, 426
Söderström, T., 87, 346, 497, 498
Sofge, D. A., 564
SPR, 213
SPR rule, 214, 215
square root method, 481
SSPA Maritime Consulting AB, 529, 532, 533
stability, 199, 215, 263
  asymptotical, 200
  bounded-input, bounded-output, 215
  criterion, 219
  definition, 200
  global, 200
  uniformly asymptotical, 204
  uniform, 204
stability of solution, 200
startup, 293, 538
Stearns, S. D., 88, 562
Steermaster, 529
Stein, G., 39, 40, 345, 417, 445
Sten, L., 418, 543
Sternby, J., 347, 373, 374, 544, 562
stochastic adaptive control, 348
stochastic approximation, 55, 56
stochastic averaging, 319
Stoica, P., 87
STR, 21, 90, 137
strange attractor, 272
strictly positive real, 213
structural instability, 314
structural stability, 272
Sturm sequence, 226
Su, R., 417
Supertuner, 504
supervision, 494
supervision logic, 540
Sussman, H. J., 417
Sylvester matrix, 153, 465
system identification, 56

T

Tang, Y., 346
Tank, D. W., 559, 564
Tannenbaum, A. R., 40
Techmation, 504
temperature control, 517
Testud, J. L., 183
Thompson, M. O., 446
Thorell, N. E., 418, 543
3M, 505
time scaling, 392, 396
torque transformation, 402
Toyo Systems, 504
transversal filter, 57
Treichler, J. R., 562
Trulsson, E., 262
Tse, E., 372, 374
Tsypkin, Y. Z., 2, 38, 262, 389, 445
Tuffs, P. S., 183
tuning, automatic, 516
tuning on demand, 516
turning controller, 530
turn-off phenomenon, 362
Tustin's operator, 333
two-armed bandit problem, 350
two-degree-of-freedom system, 4, 420

U

ultimate gain, 381
ultimate period, 381
ultrafiltration, 534
Unbehauen, H., 38, 88, 497, 542
underlying design problem, 21, 25, 90
universal stabilizers, 335
unmodeled dynamics, 288, 306
Utkin, V. I., 446, 447

V

Valavani, L. S., 261, 344, 345
Van Cauwenberghe, A. R., 183
van der Schaft, A. J., 417
Vander Velde, W. E., 445
Varaiya, P., 347, 372
variable-structure system, 419, 436
Vidyasagar, M., 262
VSS, 436

W

Watermann, D., 563
Wellstead, P. E., 39, 87, 136, 182, 498
Welsh, J. R., 446
Wenk, C. J., 374
Wertz, V., 183
Wessel, P., 543
Westcott, J. H., 562
Whatley, M. J., 417
Whitaker, H. P., 260
White, D. A., 564
Widrow, B., 88, 558, 562, 563
Wieslander, J., 136, 181, 373
Witsenhausen, H. S., 373
Wittenmark, B., 135, 136, 165, 181, 182, 184, 347, 369, 372, 373, 374, 497, 498, 562
Wolovich, W. A., 182

X

X-15, 3
Xianya, X., 182

Y

Yakowitz, S. J., 373
Ydstie, B. E., 88, 183, 346
Yokogawa, 504
Youla parameterization, 459
Young, K.-K. D., 447
Young, P. C., 88

Z

Zadeh, L. A., 563
Zafiriou, E., 40
Zames, G., 40, 262
Zanker, P., 136, 182, 498
Zappa, G., 184, 374
Zarrop, M. B., 39
Zhang, J., 183
Zhao-Ying, Z., 184
Ziegler, J. G., 389
Ziegler-Nichols
  closed-loop method, 381
  step response method, 378
  tuning rules, 378, 381
Zinober, A. S. I., 447