CONTENTS

1 WHAT IS ADAPTIVE CONTROL? 1

1.1 Introduction 1
1.2 Linear Feedback 3
1.3 Effects of Process Variations 9
1.4 Adaptive Schemes 19
1.5 The Adaptive Control Problem 24
1.6 Applications 27
1.7 Conclusions 33
Problems 34
References 38

2 REAL-TIME PARAMETER ESTIMATION 41

2.1 Introduction 41
2.2 Least Squares and Regression Models 42
2.3 Estimating Parameters in Dynamical Systems 56
2.4 Experimental Conditions 63
2.5 Simulation of Recursive Estimation 71
2.6 Prior Information 78
2.7 Conclusions 82
Problems 82
References 87

3 DETERMINISTIC SELF-TUNING REGULATORS 90

3.1 Introduction 90
3.2 Pole Placement Design 92
3.3 Indirect Self-tuning Regulators 102

3.4 Continuous-Time Self-tuners 109
3.5 Direct Self-tuning Regulators 112
3.6 Disturbances with Known Characteristics 121
3.7 Conclusions 128
Problems 129
References 135

4 STOCHASTIC AND PREDICTIVE SELF-TUNING REGULATORS 137

4.1 Introduction 137
4.2 Design of Minimum-Variance and Moving-Average Controllers 137
4.3 Stochastic Self-tuning Regulators 146
4.4 Unification of Direct Self-tuning Regulators 156
4.5 Linear Quadratic STR 164
4.6 Adaptive Predictive Control 168
4.7 Conclusions 178
Problems 179
References 181

5 MODEL-REFERENCE ADAPTIVE SYSTEMS 185

5.1 Introduction 185
5.2 The MIT Rule 186
5.3 Determination of the Adaptation Gain 194
5.4 Lyapunov Theory 199
5.5 Design of MRAS Using Lyapunov Theory 206
5.6 Bounded-Input, Bounded-Output Stability 215
5.7 Applications to Adaptive Control 230
5.8 Output Feedback 235
5.9 Relations between MRAS and STR 243
5.10 Nonlinear Systems 245
5.11 Conclusions 255
Problems 256
References 260

6 PROPERTIES OF ADAPTIVE SYSTEMS 263

6.1 Introduction 263
6.2 Nonlinear Dynamics 265
6.3 Adaptation of a Feedforward Gain 274
6.4 Analysis of Indirect Discrete-Time Self-tuners 280
6.5 Stability of Direct Discrete-Time Algorithms 293
6.6 Averaging 299
6.7 Application of Averaging Techniques 306
6.8 Averaging in Stochastic Systems 319

6.9 Robust Adaptive Controllers 327
6.10 Conclusions 338
Problems 338
References 343

7 STOCHASTIC ADAPTIVE CONTROL 348

7.1 Introduction 348
7.2 Multistep Decision Problems 350
7.3 The Stochastic Adaptive Problem 352
7.4 Dual Control 354
7.5 Suboptimal Strategies 362
7.6 Examples 365
7.7 Conclusions 370
Problems 371
References 372

8 AUTO-TUNING 375

8.1 Introduction 375
8.2 PID Control 376
8.3 Auto-tuning Techniques 377
8.4 Transient Response Methods 378
8.5 Methods Based on Relay Feedback 380
8.6 Relay Oscillations 385
8.7 Conclusions 388
Problems 388
References 389

9 GAIN SCHEDULING 390

9.1 Introduction 390
9.2 The Principle 391
9.3 Design of Gain-Scheduling Controllers 392
9.4 Nonlinear Transformations 398
9.5 Applications of Gain Scheduling 402
9.6 Conclusions 416
Problems 416
References 417

10 ROBUST AND SELF-OSCILLATING SYSTEMS 419

10.1 Why Not Adaptive Control? 419
10.2 Robust High-Gain Feedback Control 419
10.3 Self-oscillating Adaptive Systems 426

10.4 Variable-Structure Systems 436
10.5 Conclusions 442
Problems 442
References 445

11 PRACTICAL ISSUES AND IMPLEMENTATION 448

11.1 Introduction 448
11.2 Controller Implementation 449
11.3 Controller Design 458
11.4 Solving the Diophantine Equation 462
11.5 Estimator Implementation 465
11.6 Square Root Algorithms 480
11.7 Interaction of Estimation and Control 487
11.8 Prototype Algorithms 490
11.9 Operational Issues 493
11.10 Conclusions 494
Problems 496
References 497

12 COMMERCIAL PRODUCTS AND APPLICATIONS 499

12.1 Introduction 499
12.2 Status of Applications 500
12.3 Industrial Adaptive Controllers 503
12.4 Some Industrial Adaptive Controllers 506
12.5 Process Control 517
12.6 Automobile Control 527
12.7 Ship Steering 529
12.8 Ultrafiltration 534
12.9 Conclusions 541
References 542

13 PERSPECTIVES ON ADAPTIVE CONTROL 545

13.1 Introduction 545
13.2 Adaptive Signal Processing 546
13.3 Extremum Control 550
13.4 Expert Control Systems 554
13.5 Learning Systems 557
13.6 Future Trends 559
13.7 Conclusions 561
References 562

INDEX 565

CHAPTER 1

WHAT IS ADAPTIVE CONTROL?

1.1 INTRODUCTION

In everyday language, "to adapt" means to change a behavior to conform to new circumstances. Intuitively, an adaptive controller is thus a controller that can modify its behavior in response to changes in the dynamics of the process and the character of the disturbances. Since ordinary feedback also attempts to reduce the effects of disturbances and plant uncertainty, the question of the difference between feedback control and adaptive control immediately arises. Over the years there have been many attempts to define adaptive control formally. At an early symposium in 1961 a long discussion ended with the following suggestion: "An adaptive system is any physical system that has been designed with an adaptive viewpoint." A renewed attempt was made by an IEEE committee in 1973. It proposed a new vocabulary based on notions like self-organizing control (SOC) system, parameter-adaptive SOC, performance-adaptive SOC, and learning control system. However, these efforts were not widely accepted. A meaningful definition of adaptive control, which would make it possible to look at a controller hardware and software and decide whether or not it is adaptive, is still lacking. However, there appears to be a consensus that a constant-gain feedback system is not an adaptive system.

In this book we take the pragmatic attitude that *an adaptive controller is a controller with adjustable parameters and a mechanism for adjusting the parameters*. The controller becomes nonlinear because of the parameter adjustment mechanism. It has, however, a very special structure. Since general nonlinear systems are difficult to deal with, it makes sense to consider special classes of nonlinear systems. An adaptive control system can be thought of as having two loops. One loop is a normal feedback with the process and the controller. The other loop is the parameter adjustment loop. A block diagram

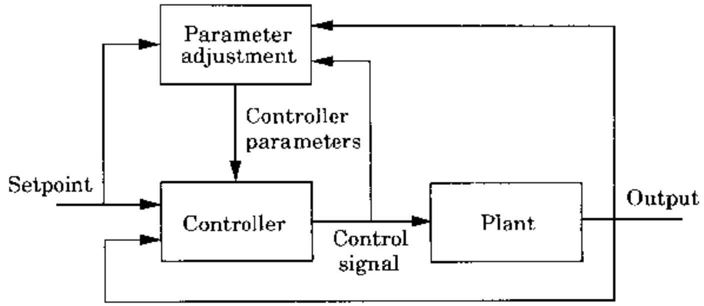

Figure 1.1 Block diagram of an adaptive system.

of an adaptive system is shown in Fig. 1.1. The parameter adjustment loop is often slower than the normal feedback loop.

A control engineer should know about adaptive systems because they have useful properties, which can be profitably used to design control systems with improved performance and functionality.

A Brief History

In the early 1950s there was extensive research on adaptive control in connection with the design of autopilots for high-performance aircraft (see Fig. 1.2). Such aircraft operate over a wide range of speeds and altitudes. It was found that ordinary constant-gain, linear feedback control could work well in one operating condition but not over the whole flight regime. A more sophisticated controller that could work well over a wide range of operating conditions was therefore needed. After a significant development effort it was found that gain scheduling was a suitable technique for flight control systems. The interest in adaptive control diminished partly because the adaptive control problem was too hard to deal with using the techniques that were available at the time.

In the 1960s there were much research in control theory that contributed to the development of adaptive control. State space and stability theory were introduced. There were also important results in stochastic control theory. Dynamic programming, introduced by Bellman, increased the understanding of adaptive processes. Fundamental contributions were also made by Tsypkin, who showed that many schemes for learning and adaptive control could be described in a common framework. There were also major developments in system identification. A renaissance of adaptive control occurred in the 1970s, when different estimation schemes were combined with various design methods. Many applications were reported, but theoretical results were very limited.

In the late 1970s and early 1980s, proofs for stability of adaptive systems appeared, albeit under very restrictive assumptions. The efforts to merge ideas

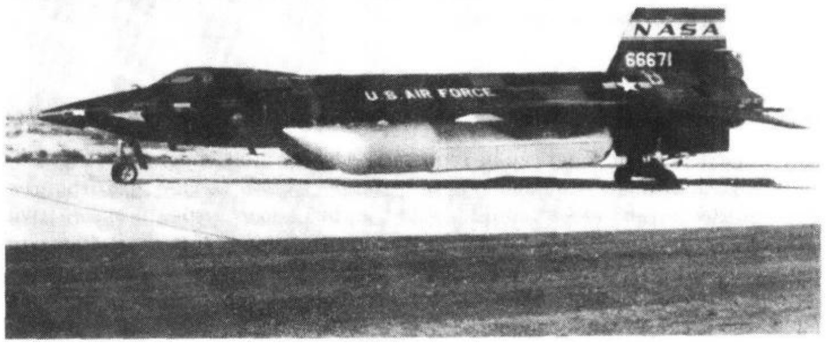

Figure 1.2 Several advanced flight control systems were tested on the X-15 experimental aircraft. (By courtesy of Smithsonian Institution.)

of robust control and system identification are of particular relevance. Investigation of the necessity of those assumptions sparked new and interesting research into the robustness of adaptive control, as well as into controllers that are universally stabilizing. Research in the late 1980s and early 1990s gave new insights into the robustness of adaptive controllers. Investigations of nonlinear systems led to significantly increased understanding of adaptive control. Lately, it has also been established that adaptive control has strong relations to ideas on learning that are emerging in the field of computer science.

There have been many experiments on adaptive control in laboratories and industry. The rapid progress in microelectronics was a strong stimulation. Interaction between theory and experimentation resulted in a vigorous development of the field. As a result, adaptive controllers started to appear commercially in the early 1980s. This development is now accelerating. One result is that virtually all single-loop controllers that are commercially available today allow adaptive techniques of some form. The primary reason for introducing adaptive control was to obtain controllers that could adapt to changes in process dynamics and disturbance characteristics. It has been found that adaptive techniques can also be used to provide automatic tuning of controllers.

1.2 LINEAR FEEDBACK

Feedback by itself has the ability to cope with parameter changes. The search for ways to design a system that are insensitive to process variations was in fact one of the driving forces for inventing feedback. Therefore it is of interest

to know the extent to which process variations can be dealt with by using linear feedback. In this section we discuss how a linear controller can deal with variations in process dynamics.

Robust High-Gain Control

A linear feedback controller can be represented by the block diagram in Fig. 1.3. The feedback transfer function G_{fb} is typically chosen so that disturbances acting on the process are attenuated and the closed-loop system is insensitive to process variations. The feedforward transfer function G_{ff} is then chosen to give the desired response to command signals. The system is called a *two-degree-of-freedom system* because the controller has two transfer functions that can be chosen independently. The fact that linear feedback can cope with significant variations in process dynamics can be seen from the following intuitive argument. Consider the system in Fig. 1.3. The transfer function from y_m to y is

$$T = \frac{G_p G_{fb}}{1 + G_p G_{fb}}$$

Taking derivatives with respect to G_p, we get

$$\frac{dT}{T} = \frac{1}{1 + G_p G_{fb}} \frac{dG_p}{G_p}$$

The closed-loop transfer function T is thus insensitive to variations in the process transfer function for those frequencies at which the loop transfer function

$$L = G_p G_{fb} \tag{1.1}$$

is large. To design a robust controller, it is thus attempted to find G_{fb} such that the loop transfer function is large for those frequencies at which there are large variations in the process transfer function. For those frequencies where $L(i\omega) \approx 1$, however, it is necessary that the variations be moderate for the system to have sufficient robustness properties.

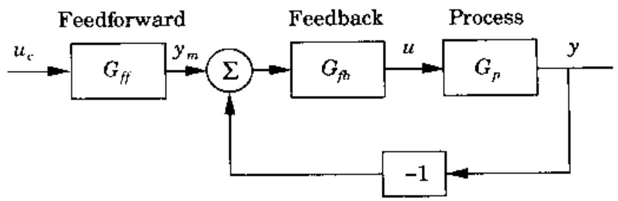

Figure 1.3 Block diagram of a robust high-gain system.

Judging Criticality of Process Variations

We now consider some specific examples to develop some intuition for judging the effects of parameter variations. The following example illustrates that significant variations in open-loop step responses may have little effect on the closed-loop performance.

EXAMPLE 1.1 Different open-loop responses

Consider systems with the open-loop transfer functions

$$G_0(s) = \frac{1}{(s+1)(s+a)}$$

where $a = -0.01, 0$, and 0.01. The dynamics of these processes are quite different, as is illustrated in Fig. 1.4(a). Notice that the responses are significantly different. The system with $a = 0.01$ is stable; the others are unstable. The initial parts of the step responses, however, are very similar for all systems. The closed-loop systems obtained by introducing the proportional feedback with unit gain, that is, $u = u_c - y$, give the step responses shown in Fig. 1.4(b). Notice that the responses of the closed-loop systems are virtually identical. Some insight is obtained from the frequency responses. Bode diagrams for the

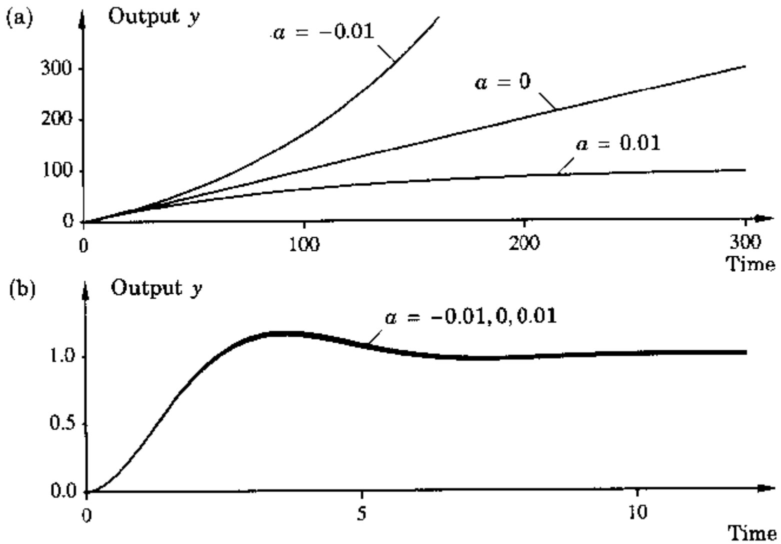

Figure 1.4 (a) Open-loop unit step responses for the process in Example 1.1 with $a = -0.01$, 0, and 0.01. (b) Closed-loop step responses for the same system, with the feedback $u = u_c - y$. Notice the difference in time scales.

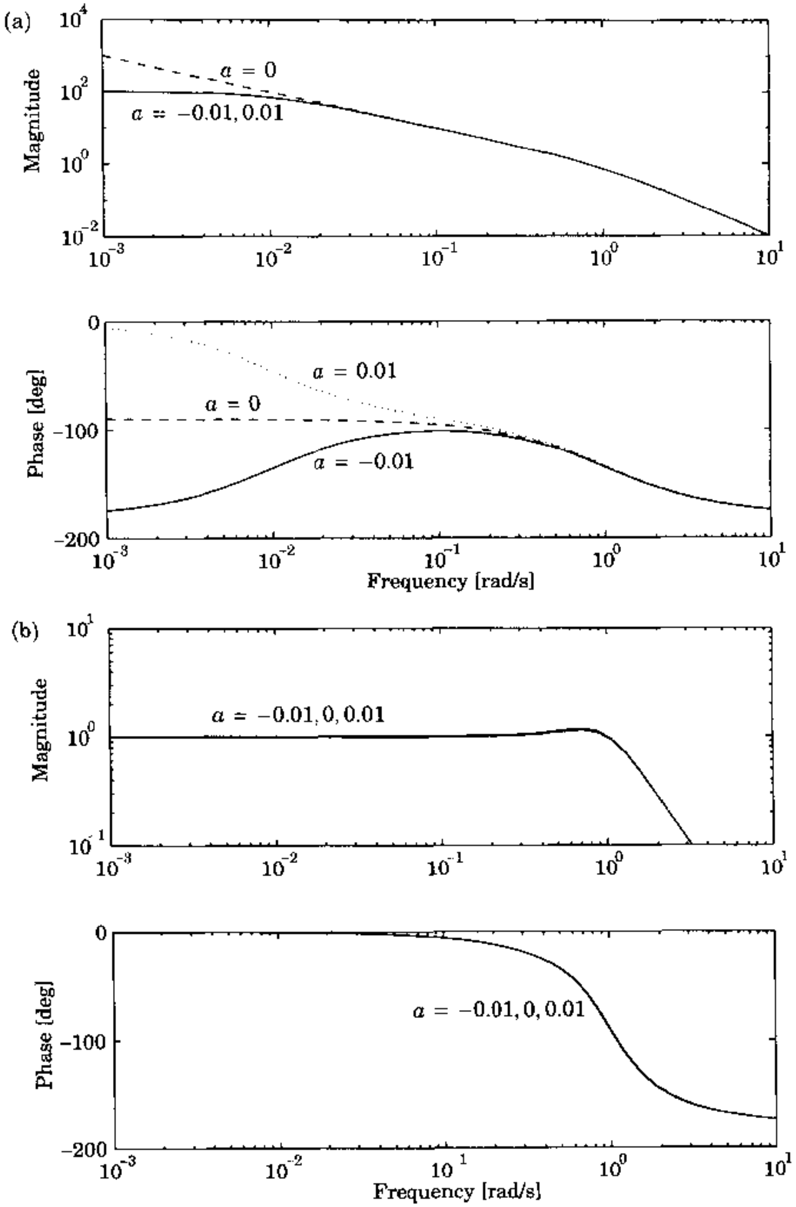

Figure 1.5 (a) Open-loop and (b) closed-loop Bode diagrams for the process in Example 1.1.

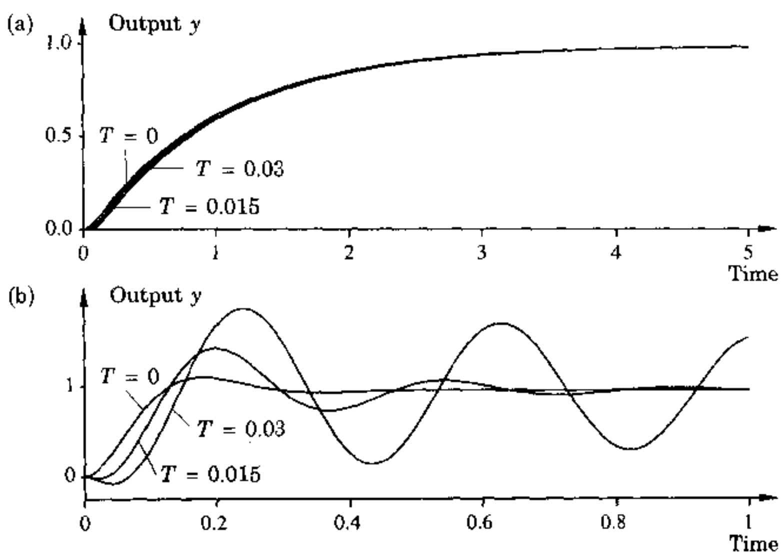

Figure 1.6 (a) Open-loop unit step responses for the process in Example 1.2 with $T = 0$, 0.015, and 0.03. (b) Closed-loop step responses for the same system, with the feedback $u = u_c - y$. Notice the difference in time scales.

open and closed loops are shown in Fig. 1.5. Notice that the Bode diagrams for the open-loop systems differ significantly at low frequencies but are virtually identical for high frequencies. Intuitively, it thus appears that there is no problem in designing a controller that will work well for all systems, provided that the closed-loop bandwidth is chosen to be sufficiently high. This is also verified by the Bode diagrams for the closed-loop systems shown in Fig. 1.5(b), which are practically identical. Also compare the step responses of the closed-loop systems in Fig. 1.4(b). □

The next example illustrates that process variations may be significant even if changes in the open-loop step responses are small.

EXAMPLE 1.2 **Similar open-loop responses**

Consider systems with the open-loop transfer functions

$$G_0(s) = \frac{400(1 - sT)}{(s + 1)(s + 20)(1 + Ts)}$$

with $T = 0$, 0.015, and 0.03. The open-loop step responses are shown in Fig. 1.6(a). Figure 1.6(b) shows the step responses for the closed-loop systems obtained with the feedback $u = u_c - y$. Notice that the open-loop responses

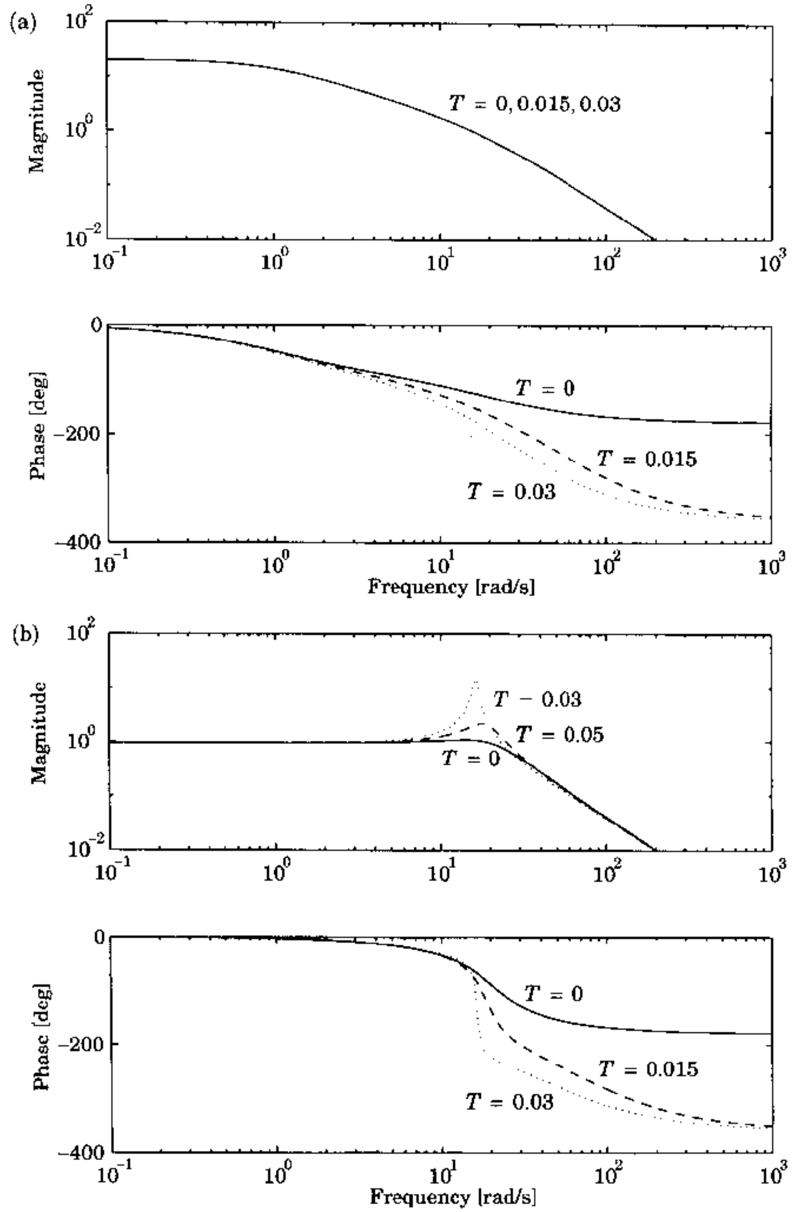

Figure 1.7 Bode diagrams for the process in Example 1.2. (a) The open-loop system; (b) The closed-loop system.

are very similar but that the closed-loop responses differ considerably. The frequency responses give some insight. The Bode diagrams for the open- and closed-loop systems are shown in Fig. 1.7. Notice that the frequency responses of the open-loop systems are very close for low frequencies but differ considerably in the phase at high frequencies. It is thus possible to design a controller that works well for all systems provided that the closed-loop bandwidth is chosen to be sufficiently small. At the crossover frequency chosen in the example there are, however, significant variations that show up in the Bode diagrams of the closed-loop systems in Fig. 1.7(b) and in the step responses of the closed-loop system in Fig. 1.6(b). □

The examples discussed show that to judge the consequences of process variations from open-loop dynamics, it is better to use frequency responses than time responses. It is also necessary to have some information about the desired crossover frequency of the closed-loop system. Intuitively, it may be expected that a process variation that changes dynamics from unstable to stable is very severe. Example 1.1 shows that this is not necessarily the case.

EXAMPLE 1.3 Integrator with unknown sign

Consider a process whose dynamics is described by

$$G_0(s) = \frac{k_p}{s} \tag{1.2}$$

where the gain k_p can assume both positive and negative values. This is a very severe variation because the phase of the system can change by 180°. This process cannot be controlled by a linear controller with a rational transfer function. This can be seen as follows. Let the controller transfer function be $S(s)/R(s)$, where $R(s)$ and $S(s)$ are polynomials. Assume that $\deg R \geq \deg S$. The characteristic polynomial of the closed-loop system is then

$$P(s) = sR(s) + k_p S(s)$$

Without lack of generality it can be assumed that the coefficient of the highest power of s in the polynomial $R(s)$ is 1. The coefficient of the highest power of s of $P(s)$ is thus also 1. The constant coefficient of polynomial $k_p S(s)$ is proportional to k_p and can thus be either positive or negative. A necessary condition for $P(s)$ to have all roots in the left half-plane is that all coefficients are positive. Since k_p can be both positive and negative, the polynomial $P(s)$ will always have a zero in the right half-plane for some value of k_p. □

1.3 EFFECTS OF PROCESS VARIATIONS

The standard approach to control system design is to develop a linear model for the process for some operating condition and to design a controller having

constant parameters. This approach has been remarkably successful. A fundamental property is also that feedback systems are intrinsically insensitive to modeling errors and disturbances. In this section we illustrate some mechanisms that give rise to variations in process dynamics. We also show the effects of process variations on the performance of a control system.

The examples are simplified to the extent that they do not create significant control problems but do illustrate some of the difficulties that might occur in real systems.

Nonlinear Actuators

A very common source of variations is that actuators, like valves, have a nonlinear characteristic. This may create difficulties, which are illustrated by the following example.

EXAMPLE 1.4 Nonlinear valve

A simple feedback loop with a Proportional and Integrating (PI) controller, a nonlinear valve, and a process is shown in Fig. 1.8. Let the static valve characteristic be

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

Linearizing the system around a steady-state operating point shows that the incremental gain of the valve is $f'(u)$, and hence the loop gain is proportional to $f'(u)$. The system can perform well at one operating level and poorly at another. This is illustrated by the step responses in Fig. 1.9. The controller is tuned to give a good response at low values of the operating level. For higer values of the operating level the closed-loop system even becomes unstable. One way to handle this type of problem is to feed the control signal u through an inverse of the nonlinearity of the valve. It is often sufficient to use a fairly crude approximation (see Example 9.1). This can be interpreted as a special case of gain scheduling, which is treated in detail in Chapter 9. □

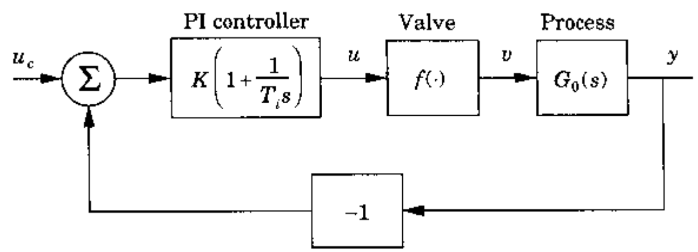

Figure 1.8 Block diagram of a flow control loop with a PI controller and a nonlinear valve.

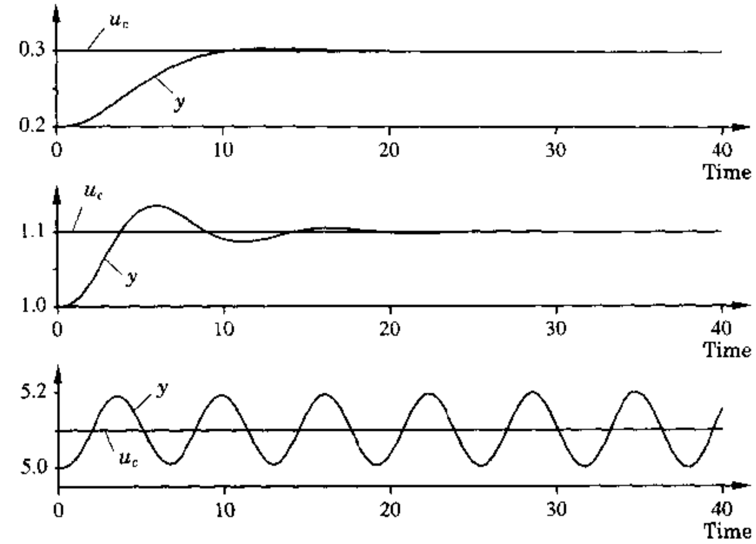

Figure 1.9 Step responses for PI control of the simple flow loop in Example 1.4 at different operating levels. The parameters of the PI controller are $K = 0.15$, $T_i = 1$. The process characteristics are $f(u) = u^4$ and $G_0(s) = 1/(s+1)^3$.

Flow and Speed Variations

Systems with flows through pipes and tanks are common in process control. The flows are often closely related to the production rate. Process dynamics thus change when the production rate changes, and a controller that is well tuned for one production rate will not necessarily work well for other rates. A simple example illustrates what may happen.

EXAMPLE 1.5 Concentration control

Consider concentration control for a fluid that flows through a pipe, with no mixing, and through a tank, with perfect mixing. A schematic diagram of the process is shown in Fig. 1.10. The concentration at the inlet of the pipe is c_{in}. Let the pipe volume be V_d and let the tank volume be V_m. Furthermore, let the flow be q and let the concentration in the tank and at the outlet be c. A mass balance gives

$$V_m \frac{dc(t)}{dt} = q(t)\left(c_{in}(t-\tau) - c(t)\right) \tag{1.3}$$

where

$$\tau = V_d/q(t)$$

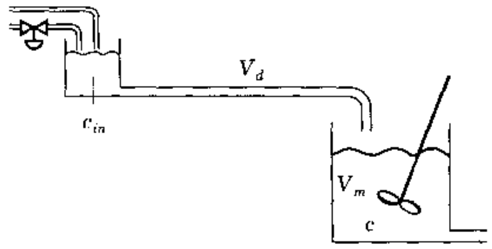

Figure 1.10 Schematic diagram of a concentration control system.

Introduce

$$T = V_m/q(t) \tag{1.4}$$

For a fixed flow, that is, when $q(t)$ is constant, the process has the transfer function

$$G_0(s) = \frac{e^{-s\tau}}{1+sT} \tag{1.5}$$

The dynamics are characterized by a time delay and first-order dynamics. The time constant T and the time delay τ are inversely proportional to the flow q.

The closed-loop system is as in Fig. 1.8 with $f(\cdot) = 1$ and $G_0(s)$ given by Eq. (1.5). A controller will first be designed for the nominal case, which corresponds to $q = 1$, $T = 1$, and $\tau = 1$. A PI controller with gain $K = 0.5$ and integration time $T_i = 1.1$ gives a closed-loop system with good performance in this case. Figure 1.11 shows the step responses of the closed-loop system for different flows and the corresponding control actions. The overshoot will increase with decreasing flows, and the system will become sluggish when the flow increases. For safe operation it is thus good practice to tune the controller at the lowest flow. Figure 1.11 shows that the system can easily cope with a flow change of ±10% but that the performance deteriorates severely when the flow changes by a factor of 2. □

Variations in speed give rise to similar problems. This happens for example in rolling mills and paper machines.

Flight Control

The dynamics of an airplane change significantly with speed, altitude, angle of attack, and so on. Control systems such as autopilots and stability augmentation systems were used early. These systems were based on linear feedback with constant coefficients. This worked well when speeds and altitudes were low, but difficulties were encountered with increasing speed and altitude. The problems became very pronounced at supersonic flight. Flight control was one of the strong driving forces for the early development of adaptive control. The

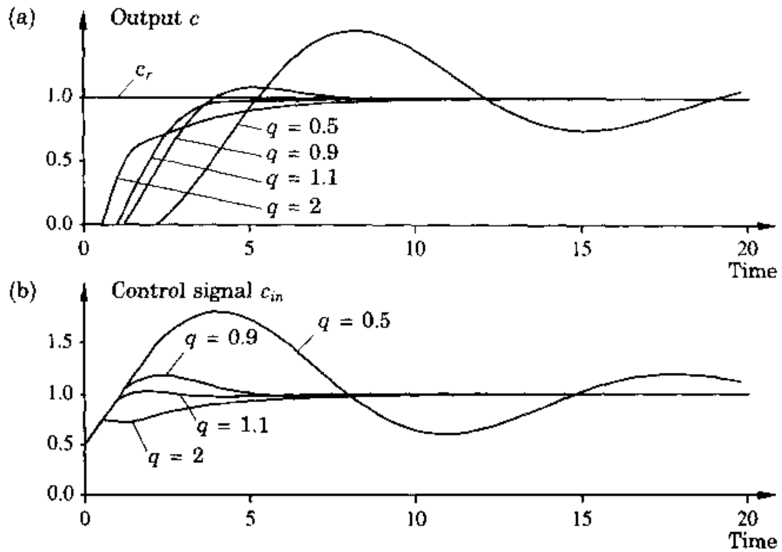

Figure 1.11 Change in reference value for different flows for the system in Example 1.5. (a) Output c and reference c_r concentration, (b) control signal.

following example from Ackermann (1983) illustrates the variations in dynamics that can be encountered. The variations can be even larger for aircraft with larger variations in flight regimes.

EXAMPLE 1.6 **Short-period aircraft dynamics**

A schematic diagram of an airplane is given in Fig. 1.12. To illustrate the effect of parameter variations, we consider the pitching motion of the aircraft. Introduce the pitch angle θ. Choose normal acceleration N_z, pitch rate $q = \dot{\theta}$,

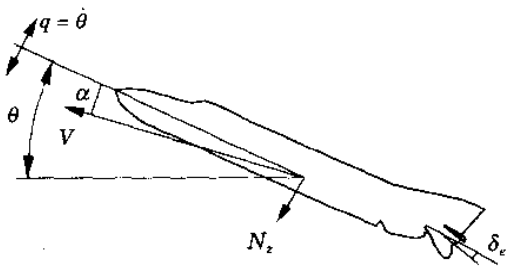

Figure 1.12 Schematic diagram of the aircraft in Example 1.6.

and elevon angle δ_e as state variables and the input to the elevon servo as the input signal u. The following model is obtained if we assume that the aircraft is a rigid body:

$$\frac{dx}{dt} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & -a \end{pmatrix} x + \begin{pmatrix} b_1 \\ 0 \\ a \end{pmatrix} u \tag{1.6}$$

where $x^T = \begin{pmatrix} N_z & \dot{\theta} & \delta_e \end{pmatrix}$. This model is called short-period dynamics. The parameters of the model given depend on the operating conditions, which can be described in terms of Mach number and altitude; see Fig. 1.13, which shows the flight envelope.

Table 1.1 shows the parameters for the four flight conditions (FC) indicated in Fig. 1.13. The data applies to the supersonic aircraft F4-E. The system has three eigenvalues. One eigenvalue, $-a = -14$, which is due to the elevon servo, is constant. The other eigenvalues, λ_1 and λ_2, depend on the flight conditions. Table 1.1 shows that the system is unstable for subsonic speeds (FC 1, 2, and 3) and stable but poorly damped for the supersonic condition FC 4. Because of these variations it is not possible to use a controller with the same parameters for all flight conditions. The operating condition is determined from air data sensors that measure altitude and Mach number. The controller parameters are then changed as a function of these parameters. How this is done is discussed in Chapter 9.

Much more complicated models will have to be considered in practice because the airframe is elastic and will bend. Notch prefilters on the command signal from the pilot are also used so that the control actions will not excite the bending modes of the airplane. □

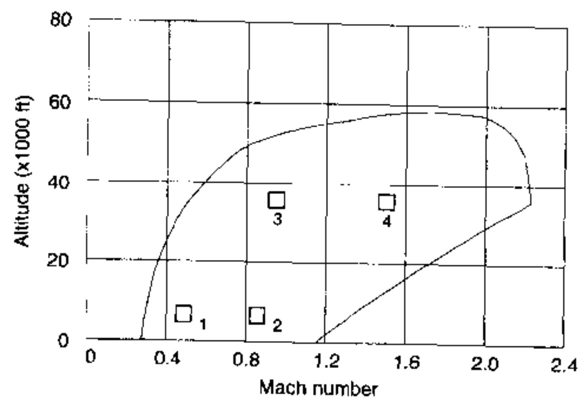

Figure 1.13 Flight envelope of the F4-E. Four different flight conditions are indicated. (From Ackermann (1983), courtesy of Springer-Verlag.)

Table 1.1 Parameters of the airplane state model of Eq. (1.6) for different flight conditions (FC).

	FC 1	FC 2	FC 3	FC 4
Mach	0.5	0.85	0.9	1.5
Altitude (feet)	5000	5000	35000	35000
a_{11}	−0.9896	−1.702	−0.667	−0.5162
a_{12}	17.41	50.72	18.11	26.96
a_{13}	96.15	263.5	84.34	178.9
a_{21}	0.2648	0.2201	0.08201	−0.6896
a_{22}	−0.8512	−1.418	−0.6587	−1.225
a_{23}	−11.39	−31.99	−10.81	−30.38
b_1	−97.78	−272.2	−85.09	−175.6
λ_1	−3.07	−4.90	−1.87	$-0.87 \pm 4.3i$
λ_2	1.23	1.78	0.56	

Variations in Disturbance Characteristics

So far, we have discussed effects of variations in process dynamics. There are also situations in which the key issue is variations in disturbance characteristics. Two examples follow.

EXAMPLE 1.7 Ship steering

A key problem in the design of an autopilot for ship steering is to compensate for the disturbing forces that act on the ship because of wind, waves, and current. The wave-generated forces are often the dominating forces. Waves have strong periodic components. The dominating wave frequency may change by a factor of 3 when the weather conditions change from light breeze to fresh gale. The frequency of the forces generated by the waves will change much more because it is also influenced by the velocity and heading of the ship. Examples of wave height and spectra for two weather conditions are shown in Fig. 1.14. It seems natural to take the nature of the wave disturbances into account in designing autopilots and roll dampers. Since the wave-induced forces change so much, it seems natural to adjust the controller parameters to cope with the disturbance characteristics. □

Positioning of ships and platforms is another example that is similar to ship steering. In this case the control system will typically have less control authority. This means that the platform to a greater extent has to "ride the waves" and can compensate only for a low-frequency component of the disturbances. This makes it even more critical to have a model for the disturbance pattern.

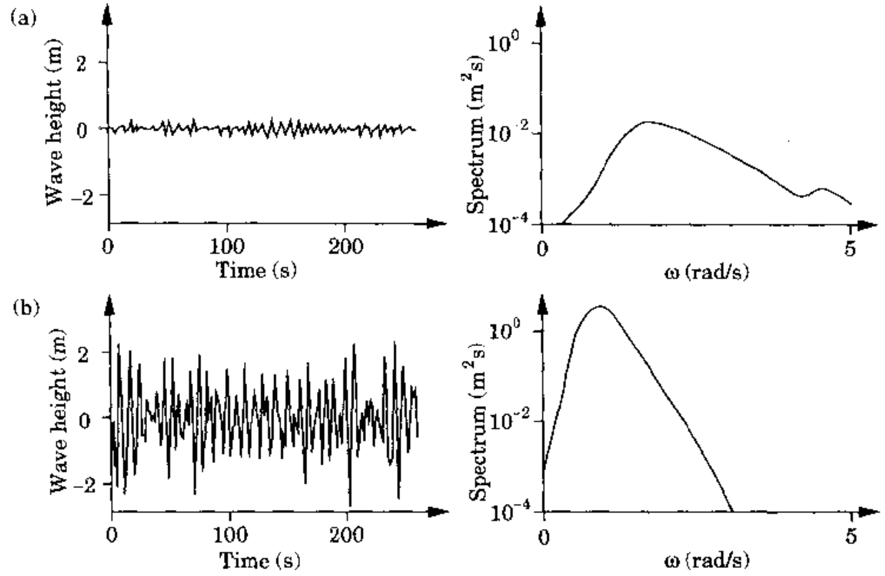

Figure 1.14 Measurements and spectra of waves at different weather conditions at Hoburgen, Sweden. (a) Wind speed 3–4 m/s. (b) Wind speed 18–20 m/s. (Courtesy of SSPA Maritime Consulting AB, Sweden.)

In process control the key issue is often to perform accurate regulation. For important quality variables, even moderate reductions in the fluctuation of a quality variable can give substantial savings. If the disturbances have some statistical regularity, it is possible to obtain significant improvements in control quality by having a controller that is tuned to the particular character of the disturbance. Such controllers can give much better performance than standard PI controllers. The consequences of compensating for disturbances are illustrated by an example.

EXAMPLE 1.8 Regulation of a quality variable in process control

Consider regulation of a quality variable of an industrial process in which there are disturbances whose characteristics are changing. A block diagram of the system is shown in Fig. 1.15. In the experiment it is assumed that the process dynamics are first order with time constant $T = 1$. It is assumed that the disturbance acts on the process input. The disturbance is simulated by sending white noise through a band-pass filter. The process dynamics are constant, but the frequency of the band-pass filter changes. Regulation can be done by a PI controller, but performance can be improved significantly by using a more complex controller that is tuned to the disturbance character. Such a

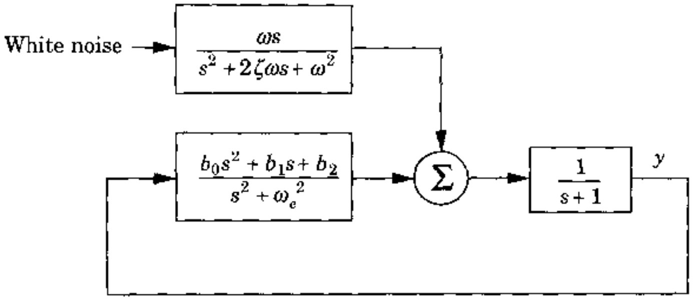

Figure 1.15 Block diagram of the system with disturbances used in Example 1.8.

controller has a very high gain at the center frequency of the disturbance. Figure 1.16 shows the control error under different conditions. The center frequency of the band-pass filter used to generate the disturbance is ω, and the corresponding value used in the design of the controller is ω_e. In Fig. 1.16(a) we show the control error obtained when the controller is tuned to the disturbance,

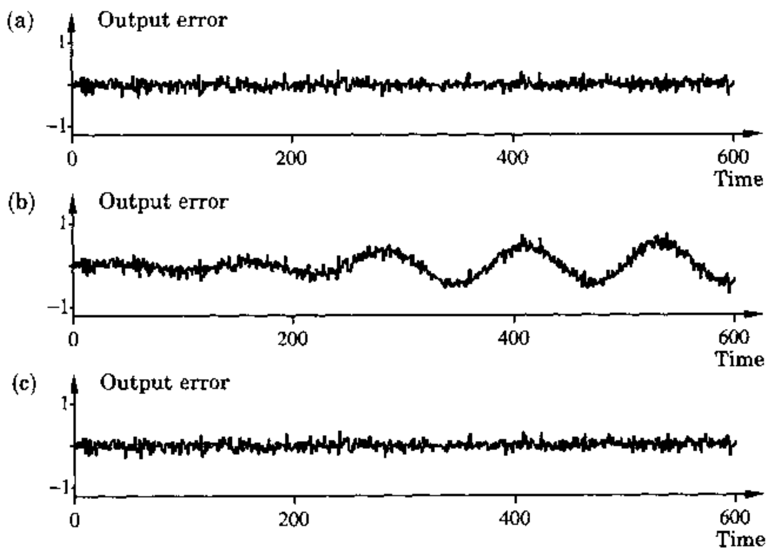

Figure 1.16 Illustrates performance of controllers that are tuned to the disturbance characteristics. Output error when (a) $\omega = \omega_e = 0.1$; (b) $\omega = 0.05$, $\omega_e = 0.1$; (c) $\omega = \omega_e = 0.05$.

that is, $\omega_e = \omega = 0.1$. In Fig. 1.16(b) we illustrate what happens when the disturbance properties change. Parameter ω is changed to 0.05, while $\omega_e = 0.1$. The performance of the control system now deteriorates significantly. In Fig. 1.16(c) we show the improvement obtained by tuning the controller to the new conditions, that is, $\omega = \omega_e = 0.05$. □

There are many other practical problems of a similar type in which there are significant variations in the disturbance characteristics. Having a controller that can adapt to changing disturbance patterns is particularly important when there is limited control authority or dead time in the process dynamics.

Summary

The examples in this section illustrate some mechanisms that can create variations in process dynamics. The examples have of necessity been very simple to show some of the difficulties that may occur. In some cases it is straightforward to reduce the variations by introducing nonlinear compensations in the controllers. For the nonlinear valve in Example 1.4 it is natural to introduce a nonlinear compensator at the controller output that is the inverse of the valve characteristics. This modification is done in Example 9.1. The variations in flow rate in Example 1.5 can be dealt with in a similar way by measuring the flow and changing the controller parameters accordingly. To compensate for the variations in dynamics in Example 1.6, it is necessary to measure the flight conditions. In Examples 1.7 and 1.8, in which the variations are due to changes in the disturbances, it is not possible to directly relate the variation to a measurable quantity. In these cases it may be very advantageous to use adaptive control.

In practice there are many different sources of variations, and there is usually a mixture of different phenomena. The underlying reasons for the variations are in most cases not fully understood. When the physics of the process is reasonably well known (as for airplanes), it is possible to determine suitable controller parameters for different operating conditions by linearizing the models and using some method for control design. This is the common way to design autopilots for airplanes. System identification is an alternative to physical modeling. Both approaches do, however, require a significant engineering effort.

Most industrial processes are very complex and not well understood; it is neither possible nor economical to make a thorough investigation of the causes of the process variations. Adaptive controllers can be a good alternative in such cases. In other situations, some of the dynamics may be well understood, but other parts are unknown. A typical example is robots, for which the geometry, motors, and gearboxes do not change but the load does change. In such cases it is of great importance to use the available *a priori* knowledge and estimate and adapt only to the unknown part of the process.

1.4 ADAPTIVE SCHEMES

In this section we describe four types of adaptive systems: gain scheduling, model-reference adaptive control, self-tuning regulators, and dual control.

Gain Scheduling

In many cases it is possible to find measurable variables that correlate well with changes in process dynamics. A typical case is given in Example 1.4. These variables can then be used to change the controller parameters. This approach is called *gain scheduling* because the scheme was originally used to measure the gain and then change, that is, schedule, the controller to compensate for changes in the process gain. A block diagram of a system with gain scheduling is shown in Fig. 1.17. The system can be viewed as having two loops. There is an inner loop composed of the process and the controller and an outer loop that adjusts the controller parameters on the basis of the operating conditions. Gain scheduling can be regarded as a mapping from process parameters to controller parameters. It can be implemented as a function or a table lookup.

The concept of gain scheduling originated in connection with the development of flight control systems. In this application the Mach number and the altitude are measured by air data sensors and used as scheduling variables. This was used, for instance, in the X-15 in Fig. 1.2. 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. Gain scheduling is thus a very useful technique for reducing the effects of parameter variations. Historically, it has been a matter of controversy whether gain scheduling should be considered an adaptive system or not. If we use the informal definition in Section 1.1 that an adaptive system is a controller with

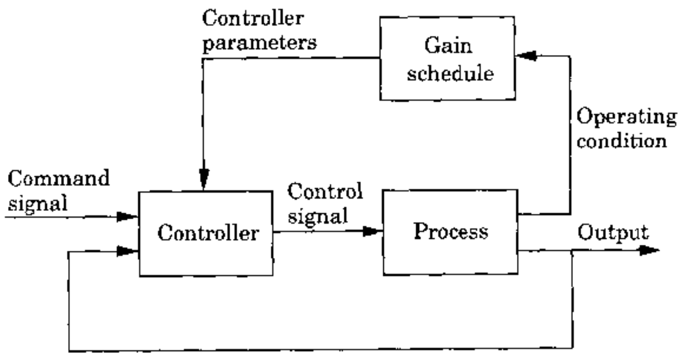

Figure 1.17 Block diagram of a system with gain scheduling.

adjustable parameters and an adjustment mechanism, it is clearly adaptive. An in-depth discussion of gain scheduling is given in Chapter 9.

Model-Reference Adaptive Systems (MRAS)

The *model-reference adaptive system* (MRAS) was originally proposed to solve a problem in which the performance specifications are given in terms of a reference model. This model tells how the process output ideally should respond to the command signal. A block diagram of the system is shown in Fig. 1.18. The controller can be thought of as consisting of two loops. The inner loop is an ordinary feedback loop composed of the process and the controller. The outer loop adjusts the controller parameters in such a way that the error, which is the difference between process output y and model output y_m, is small. The MRAS was originally introduced for flight control. In this case the reference model describes the desired response of the aircraft to joystick motions.

The key problem with MRAS is to determine the adjustment mechanism so that a stable system, which brings the error to zero, is obtained. This problem is nontrivial. The following parameter adjustment mechanism, called the *MIT rule*, was used in the original MRAS:

$$\frac{d\theta}{dt} = -\gamma e \frac{\partial e}{\partial \theta} \tag{1.7}$$

In this equation, $e = y - y_m$ denotes the model error and θ is a controller parameter. The quantity $\partial e/\partial\theta$ is the sensitivity derivative of the error with respect to parameter θ. The parameter γ determines the adaptation rate. In practice it is necessary to make approximations to obtain the sensitivity derivative. The MIT rule can be regarded as a gradient scheme to minimize the squared error e^2.

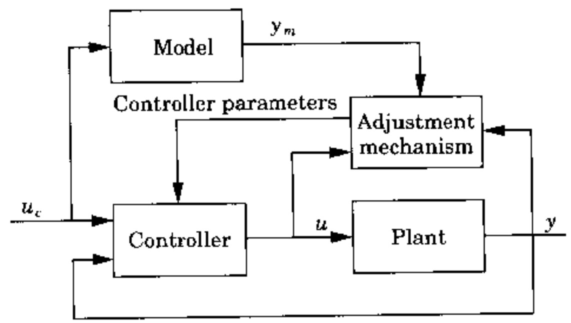

Figure 1.18 Block diagram of a model-reference adaptive system (MRAS).

Self-tuning Regulators (STR)

The adaptive schemes discussed so far are called *direct* methods, because the adjustment rules tell directly how the *controller* parameters should be updated. A different scheme is obtained if the estimates of the *process* parameters are updated and the controller parameters are obtained from the solution of a design problem using the estimated parameters. A block diagram of such a system is shown in Fig. 1.19. The adaptive controller can be thought of as being composed of two loops. The inner loop consists of the process and an ordinary feedback controller. The parameters of the controller are adjusted by the outer loop, which is composed of a recursive parameter estimator and a design calculation. It is sometimes not possible to estimate the process parameters without introducing probing control signals or perturbations. Notice that the system may be viewed as an automation of process modeling and design, in which the process model and the control design are updated at each sampling period. A controller of this construction is called a *self-tuning regulator (STR)* to emphasize that the controller automatically tunes its parameters to obtain the desired properties of the closed-loop system. Self-tuning regulators are discussed in detail in Chapters 3 and 4.

The block labeled "Controller design" in Fig. 1.19 represents an on-line solution to a design problem for a system with known parameters. This is the *underlying design problem*. Such a problem can be associated with most adaptive control schemes, but it is often given indirectly. To evaluate adaptive control schemes, it is often useful to find the underlying design problem, because it will give the characteristics of the system under the ideal conditions when the parameters are known exactly.

The STR scheme is very flexible with respect to the choice of the underlying design and estimation methods. Many different combinations have been

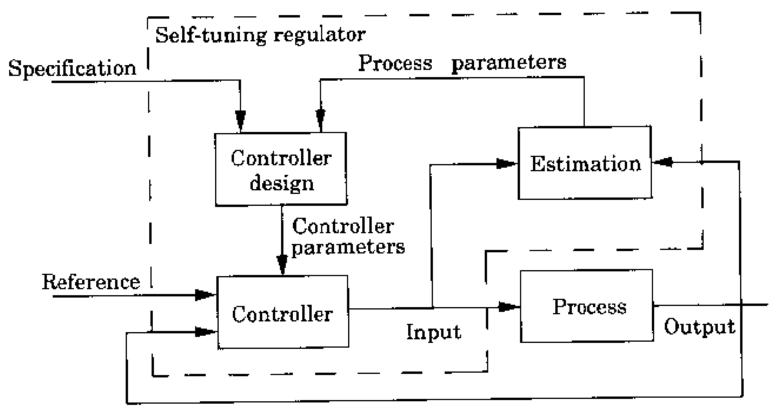

Figure 1.19 Block diagram of a self-tuning regulator (STR).

explored. The controller parameters are updated indirectly via the design calculations in the self-tuner shown in Fig. 1.19. It is sometimes possible to reparameterize the process so that the model can be expressed in terms of the controller parameters. This gives a significant simplification of the algorithm because the design calculations are eliminated. In terms of Fig. 1.19 the block labeled "Controller design" disappears, and the controller parameters are updated directly.

In the STR the controller parameters or the process parameters are estimated in real time. The estimates are then used as if they are equal to the true parameters (i.e., the uncertainties of the estimates are not considered). This is called the *certainty equivalence principle*. In many estimation schemes it is also possible to get a measure of the quality of the estimates. This uncertainty may then be used in the design of the controller. For example, if there is a large uncertainty, one may choose a conservative design. This is discussed in Chapter 7.

Dual Control

The schemes for adaptive control described so far look like reasonable heuristic approaches. Already from their description it appears that they have some limitations. For example, parameter uncertainties are not taken into account in the design of the controller. It is then natural to ask whether there are better approaches than the certainty equivalence scheme. We may also ask whether adaptive controllers can be obtained from some general principles. It is possible to obtain a solution that follows from an abstract problem formulation and use of optimization theory. The particular tool one could use is nonlinear stochastic control theory. This will lead to the notion of *dual control*. The approach will give a controller structure with interesting properties. A major consequence is that the uncertainties in the estimated parameters will be taken into account in the controller. The controller will also take special actions when it has poor knowledge about the process. The approach is so complicated, however, that so far it has not been possible to use it for practical problems. Since the ideas are conceptually useful, we will discuss them briefly in this section.

The first problem that we are faced with is to describe mathematically the idea that a constant or slowly varying parameter is unknown. An unknown constant can be modeled by the differential equation

$$\frac{d\theta}{dt} = 0 \tag{1.8}$$

with an initial distribution that reflects the parameter uncertainty. Parameter drift can be described by adding random variables to the right-hand side of Eq. (1.8). A model of a plant with uncertain parameters is thus obtained by augmenting the state variables of the plant and its environment by the parameter vector whose dynamics is given by Eq. (1.8). Notice that with this

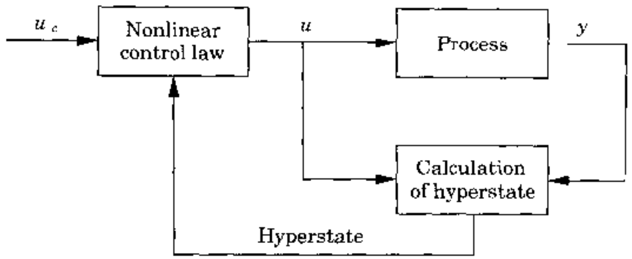

Figure 1.20 Block diagram of a dual controller.

formulation there is no distinction between these parameters and the other state variables. This means that the resulting controller can handle very rapid parameter variations. An augmented state $z = \begin{pmatrix} x^T & \theta^T \end{pmatrix}^T$ consisting of the state of the process and the parameters can now be introduced. The goal of the control is then formulated to minimize a loss function

$$V = E\left(G\left(z(T), u(T)\right) + \int_0^T g(z,u)\,dt\right)$$

where E denotes mathematical expectation, u is the control variable, and G and g are scalar functions of z and u. The expectation is taken with respect to the distribution of all initial values and all disturbances appearing in the models of the system. The criterion V should be minimized with respect to admissible controls that are such that $u(t)$ is a function of past and present measurements and the prior distributions. The problem of finding a controller that minimizes the loss function is difficult. By making sufficient assumptions a solution can be obtained by using dynamic programming. The solution is then given in terms of a functional equation that is called the *Bellman equation*. This equation is an extension of the Hamilton-Jacobi equation in the calculus of variations. It is very difficult and time-consuming, if at all possible, to solve the Bellman equation numerically.

Some structural properties are shown in Fig. 1.20. The controller can be regarded as being composed of two parts: a nonlinear estimator and a feedback controller. The estimator generates the conditional probability distribution of the state from the measurements, $p(z|y,u)$. This distribution is called the *hyperstate* of the problem. The feedback controller is a nonlinear function that maps the hyperstate into the space of control variables. This function could be computed off-line. The hyperstate must, however, be updated on-line. The structural simplicity of the solution is obtained at the price of introducing the hyperstate, which is a quantity of very high dimension. Updating of the hyperstate generally requires solution of a complicated nonlinear filtering problem. In simple cases the distribution can be characterized by its mean and covariance, as will be shown in Chapter 7.

The optimal controller sometimes has some interesting properties, which have been found by solving a number of specific problems. It attempts to drive the output to its desired value, but it will also introduce perturbations (probing) when the parameters are uncertain. This improves the quality of the estimates and the future performance of the closed-loop system. The optimal control gives the correct balance between maintaining good control and small estimation errors. The name *dual control* was coined to express this property.

It is interesting to compare the controller in Fig. 1.20 with the self-tuning regulator in Fig. 1.19. In the STR the states are separated into two groups: the ordinary state variables of the underlying constant parameter model and the parameters, which are assumed to vary slowly. The parameter estimator may be considered as an observer for the parameters. Notice that many estimators will also provide estimates of the uncertainties, although this is not used in calculating the control signal. The calculation of the hyperstate in the dual controller gives the conditional distribution of all states and all parameters of the process. The conditional mean value represents estimates, and the conditional covariances give the uncertainties of the estimates. Uncertainties are not used in computing the control signal in the self-tuning regulator. They are important for the dual controller because it may automatically introduce perturbations when the estimates are poor. Dual control is discussed in more detail in Chapter 7.

1.5 THE ADAPTIVE CONTROL PROBLEM

In this section we formulate the adaptive control problem. We do this by giving examples of process models, controller structures, and ways to adapt the controller parameters.

Process Descriptions

In this book the processes will mainly be described by linear single-input, single-output systems. In continuous time the process can be in state space form:

$$\begin{aligned}\frac{dx}{dt} &= Ax + Bu \\ y &= Cx\end{aligned} \tag{1.9}$$

or in transfer function form:

$$G_p(s) = \frac{B(s)}{A(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \cdots b_m}{s^n + a_1 s^{n-1} + \cdots a_n} \tag{1.10}$$

where s is the Laplace transform variable. Notice that A, B, and C are used for matrices as well as polynomials. In normal cases this will not cause any misunderstanding. In ambiguous cases the argument will be used in the polynomials.

In discrete time the process can be described in state space form:

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

where the sampling interval is taken as the time unit. The discrete time system can also be represented by the pulse transfer function

$$H_p(z) = \frac{B(z)}{A(z)} = \frac{b_0 z^m + b_1 z^{m-1} + \cdots b_m}{z^n + a_1 z^{n-1} + \cdots a_n} \qquad (1.11)$$

where z is the z-transform variable.

The parameters, $b_0, b_1, \ldots, b_m, a_1, \ldots, a_n$ of systems (1.10) and (1.11) as well as the orders m, n are often assumed to be unknown or partly unknown.

A Remark on Notation

Throughout this book we need a convenient notation for the time functions obtained in passing signals through linear systems. For this purpose we will use the *differential operator* $p = d/dt$. The output of the system with the transfer function $G(s)$ when the input signal is $u(t)$ will then be denoted by

$$y(t) = G(p)u(t)$$

The output will also depend on the initial conditions. In using the above notation it is assumed that all initial conditions are zero. To deal with discrete time systems, we introduce the *forward shift operator* q defined by

$$qy(t) = y(t+1)$$

The output of a system with input u and pulse transfer function $H(z)$ is denoted by

$$y(t) = H(q)u(t)$$

In this case it is also assumed that all initial conditions are zero.

Controller Structures

The process is controlled by a controller that has adjustable parameters. It is assumed that there exists some kind of design procedure that makes it possible to determine a controller that satisfies some design criteria if the process and its environment are known. This is called the *underlying design problem*. The adaptive control problem is then to find a method of adjusting the controller when the characteristics of the process and its environment are unknown or changing. In *direct adaptive control* the controller parameters are changed

directly without the characteristics of the process and its disturbances first being determined. In *indirect adaptive methods* the process model and possibly the disturbance characteristics are first determined. The controller parameters are designed on the basis of this information.

One key problem is the parameterization of the controller. A few examples are given to illustrate this.

EXAMPLE 1.9 Adjustment of gains in a state feedback

Consider a single-input, single-output process described by Eq. (1.9). Assume that the order n of the process is known and that the controller is described by

$$u = -Lx$$

In this case the controller is parameterized by the elements of the matrix L. □

EXAMPLE 1.10 A general linear controller

A general linear controller can be described by

$$R(s)U(s) = -S(s)Y(s) + T(s)U_c(s)$$

where R, S, and T are polynomials and U, Y, and U_c are the Laplace transform of the control signal, the process output, and the reference value, respectively. Several design methods are available to determine the parameters in the controller when the system is known. □

In Examples 1.9 and 1.10 the controller is linear. Of course, parameters can also be adjusted in nonlinear controllers. A common example is given next.

EXAMPLE 1.11 Adjustment of a friction compensator

Friction is common in all mechanical systems. Consider a simple servo drive. Friction can to some extent be compensated for by adding the signal u_{fc} to a controller, where

$$u_{fc} = \begin{cases} u_+ & \text{if } v > 0 \\ -u_- & \text{if } v < 0 \end{cases}$$

where v is the velocity. The signal attempts to compensate for Coulomb friction by adding a positive control signal u_+ when the velocity is positive and subtracting u_- when the velocity is negative. The reason for having two parameters is that the friction forces are typically not symmetrical. Since there are so many factors that influence friction, it is natural to try to find a mechanism that can adjust the parameters u_+ and u_- automatically. □

The Adaptive Control Problem

An adaptive controller has been defined as a controller with adjustable parameters and a mechanism for adjusting the parameters. The construction of an adaptive controller thus contains the following steps:

- Characterize the desired behavior of the closed-loop system.
- Determine a suitable control law with adjustable parameters.
- Find a mechanism for adjusting the parameters.
- Implement the control law.

In this book, different ways to derive the adjustment rule will be discussed.

1.6 APPLICATIONS

There have been a number of applications of adaptive feedback control since the mid-1950s. The early experiments, which used analog implementations, were plagued by hardware problems. Systems implemented by using minicomputers appeared in the early 1970s. The number of applications has increased drastically with the advent of the microprocessor, which has made the technology cost-effective. Adaptive techniques have been used in regular industrial controllers since the early 1980s. Today, a large number of industrial control loops are under adaptive control. These include a wide range of applications in aerospace, process control, ship steering, robotics, and automotive and biomedical systems. The applications have shown that there are many cases in which adaptive control is very useful, others in which the benefits are marginal, and yet others in which it is inappropriate. On the basis of the products and their uses, it is clear that adaptive techniques can be used in many different ways. In this section we give a brief discussion of some applications. More details are given in Chapter 12.

Automatic Tuning

The most widespread applications are in automatic tuning of controllers. By automatic tuning we mean that the parameters of a standard controller, for instance a PID controller, are tuned automatically at the demand of the operator. After the tuning, the parameters are kept constant. Practically all controllers can benefit from tools for automatic tuning. This will drastically simplify the use of controllers. Practically all adaptive techniques can be used for automatic tuning. There are also many special techniques that can be used for this purpose. Single-loop controllers and distributed systems for process control are important application areas. Most of these controllers are of the PID type. This is a vast application area because there are millions of controllers of this type in use. Many of them are poorly tuned.

Although automatic tuning is currently widely used in simple controllers, it is also beneficial for more complicated controllers. It is in fact a prerequisite for the widespread use of more advanced control algorithms. A mechanism for automatic tuning is often necessary to get the correct time scale and to find a starting value for a more complex adaptive controller. The main advantage of using an automatic tuner is that it simplifies tuning drastically and thus contributes to improved control quality. Tuners have also been developed for other standard applications such as motor control. This is also a case in which a fairly standardized system has to be applied to a wide variety of applications.

Gain Scheduling

Gain scheduling is a powerful technique that is straightforward and easy to use. The key problem is to find suitable scheduling variables, that is, variables that characterize the operating conditions (see Fig. 1.17). It may also be a significant engineering effort to determine the schedules. This effort can be reduced significantly by using automatic tuning because the schedules can then be determined experimentally. Auto-tuning or adaptive algorithms may be used to build gain schedules. A scheduling variable is first determined. Its range is quantized into a number of discrete operating conditions. The controller parameters are determined by automatic tuning when the system is running in one operating condition. The parameter values are stored in a table. The procedure is repeated until all operating conditions are covered. In this way it is easy to install and tune gain scheduling into a computer-controlled system. The only facility required is a table for storing and recalling controller parameters.

Gain scheduling is the standard technique used in flight control systems for high-performance aircrafts. An example is given in Fig. 1.21. A massive engineering effort is required to develop such systems. Gain scheduling is increasingly being used for industrial process control. A combination with automatic tuning makes it possible to significantly reduce the engineering effort in developing the systems.

Continuous Adaptation

There are several cases in which the process or the disturbance characteristics are changing continuously. Continuous adaptation of controller parameters is then needed. The MRAS and the STR are the most common approaches for parameter adjustment. There are many different ways to use the techniques. In some cases, it is natural to assume that the process is described by a general linear model. In other cases, parts of the model are known and only a few parameters are adjusted. In many situations it is possible to measure the disturbances acting on a system. A typical example is climate control in houses in which the outdoor temperature can be measured. The process of using the

Figure 1.21 Gain scheduling is an important ingredient in modern flight control systems. (By courtesy of Nawrocki Stock Photo, Inc., Neil Hargreave.)

measurable disturbance and compensating for its influence is called *feedforward*. Adaptation of feedforward compensators has been found particularly beneficial. One reason for this is that feedforward control requires good models. Another is that it is difficult and time consuming to tune feedforward loops because it is necessary to wait for a proper disturbance to appear. Adaptation is thus almost a prerequisite for using feedforward control.

Since adaptive control is a relatively new technology, there is limited experience of its use in products. One observation that has been made is that the human-machine interface is very important. Adaptive controllers also have their own parameters, which must be chosen. It has been our experience that controllers without any externally adjusted parameters can be designed for specific applications in which the purpose of control can be stated *a priori*. Autopilots for missiles and ships are typical examples. However, in many cases 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 *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. This is very convenient for applications to motor control. Another possibility would be to have a controller with a dial labeled with the weighting between state deviation and control action in a quadratic optimization 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. On the whole it appears that there is significant room for engineering ingenuity in the packaging of adaptive techniques.

Abuses of Adaptive Control

An adaptive controller, being inherently nonlinear, is more complicated than a fixed-gain controller. Before attempting to use adaptive control, it is therefore important to investigate whether the control problem might be solved by constant-gain feedback. In the literature on adaptive control there are many cases in which constant-gain feedback can do as well as an adaptive controller. This is one reason why we are discussing alternatives to adaptive control in this book. One way to proceed in deciding whether adaptive control should be used is sketched in Fig. 1.22.

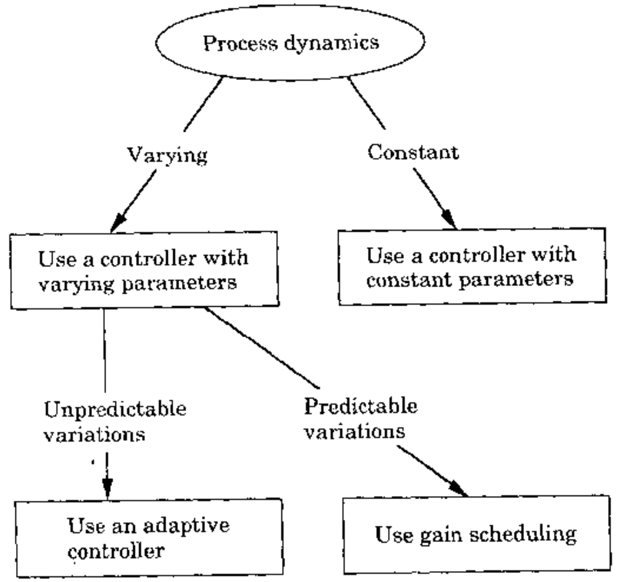

Figure 1.22 Procedure to decide what type of controller to use.

Industrial Products

The industrial products can, broadly speaking, be divided into three different categories: standard controllers, distributed control systems, and dedicated special-purpose systems.

Standard controllers form the largest category. They are typically based on some version of the PID algorithm. Currently, there is very vigorous development of these systems, which are manufactured in large quantities. Practically all new single-loop controllers introduced use some form of adaptation. Many different schemes are used. The single-loop controller is in fact becoming a proving ground for adaptive control. One example is shown in Fig. 1.23. This

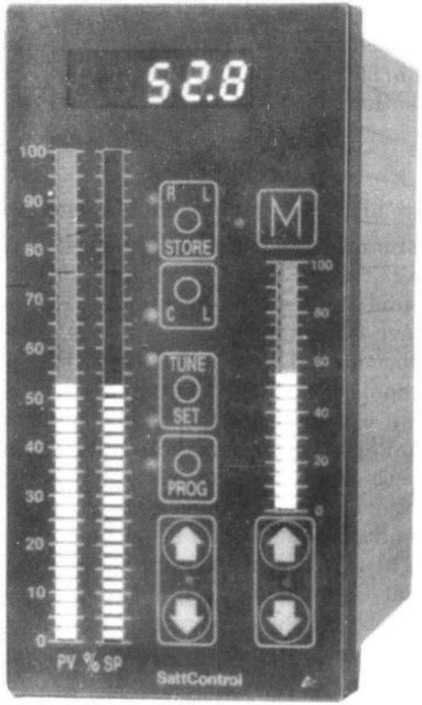

Figure 1.23 A commercial PID controller with automatic tuning, gain scheduling, and feedforward (SattControl Instruments ECA50). Tuning is performed on operator demand when the tune button is pushed. (By courtesy of SattControl Instrument.)

system has automatic tuning of the PID controller. The controller also has feedforward and gain scheduling. The automatic tuning is implemented in such a way that the user only has to push a button to execute the tuning.

A standard controller may be regarded as automation of the actions of a process operator. The controller shown in Fig. 1.23 may be viewed as the next level of automation, in which the actions of an instrument engineer are automated.

Distributed control systems are general-purpose systems primarily for process control applications. These systems may be viewed as a toolbox for implementing a wide variety of control systems. Typically, in addition to tools for PID control, alarm, and startup, more advanced control schemes are also incorporated. Adaptive techniques are now being introduced in the distributed systems, although the rate of development is not as rapid as for single-loop controllers.

There are many special-purpose systems for adaptive control. The applications range from space vehicles to automobiles and consumer electronics. The spacecraft Gemini, for example, has an adaptive notch filter and adaptive friction compensation. The following is another example of an adaptive controller.

EXAMPLE 1.12 An adaptive autopilot for ship steering

This is an example of a dedicated system for a special application. The adaptive autopilot is superior to a conventional autopilot for two reasons: It gives better performance, and it is easier to operate. A conventional autopilot has three dials, which have to be adjusted over a continuous scale. The adaptive autopilot has a performance-related switch with two positions (tight steering and economic propulsion). In the tight steering mode the autopilot gives good, fast response to commands with no consideration for propulsion efficiency. In the economic propulsion mode the autopilot attempts to minimize the steering loss. The control performance is significantly better than that of a well-adjusted conventional autopilot, as shown in Fig. 1.24. The figure shows heading deviations and rudder motions for an adaptive autopilot and a conventional autopilot. The experiments were performed under the same weather conditions. Notice that the heading deviations for the adaptive autopilot are much smaller than those for the conventional autopilot but that the rudder motions are of the same magnitude. The adaptive autopilot is better because

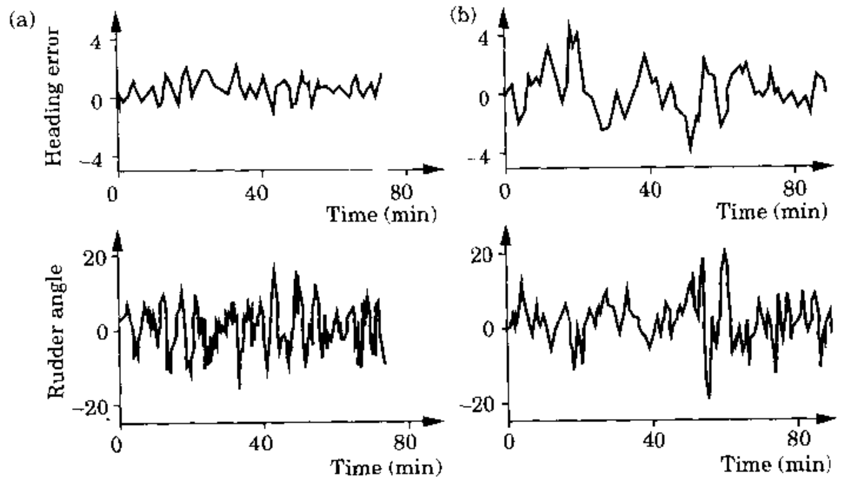

Figure 1.24 The figure shows the variations in heading and the corresponding rudder motions of a ship. (a) Adaptive autopilot. (b) Conventional autopilot based on a PID-like algorithm.

it uses a more complicated control law, which has eight parameters instead of three for the conventional autopilot. For example, the adaptive autopilot has an internal model of the wave motion. If the adaptation mechanism is switched off, the constant parameter controller obtained will perform well for a while, but its performance will deteriorate as the conditions change. Since it is virtually impossible to adjust eight parameters manually, adaptation is a necessity for using such a controller. The adaptive autopilot is discussed in more detail in Chapter 12. □

The next example illustrates a general-purpose adaptive system.

EXAMPLE 1.13 Novatune

The first general-purpose adaptive system was Novatune, announced by the Swedish company Asea in 1982. The system can be regarded as a software-configured toolbox for solving control problems. It broke with conventional process control by using a general-purpose discrete-time pulse transfer function as the building block. The system also has elements for conventional PI and PID control, lead-lag filter, logic, sequencing, and three modules for adaptive control. It has been used to implement control systems for a wide range of process control problems. The advantage of the system is that the control system designer has a simple means of introducing adaptation. The adaptive controller is now incorporated in ABB Master (see Chapter 12). □

1.7 CONCLUSIONS

The purpose of this chapter has been to introduce the notion of adaptive control, to describe some adaptive systems, and to indicate why adaptation is useful. An adaptive controller was defined as a controller with adjustable parameters and a mechanism for adjusting the parameters.

The key new element is the parameter adjustment mechanisms. Five ways of doing this were discussed: gain scheduling, auto tuning, model-reference adaptive control, self-tuning control, and dual control. To present a balanced account and to give the knowledge required to make complete systems, all aspects of the adaptive problem will be discussed in the book.

Some reasons for using adaptive control have also been discussed in this chapter. The key factors are

- variations in process dynamics,
- variations in the character of the disturbances, and
- engineering efficiency and ease of use.

Examples of mechanisms that cause variations in process dynamics have been given. The examples are simplistic; in many real-life problems it is difficult

to describe the mechanisms analytically. Variations in the character of disturbances is another strong reason for using adaptation.

Adaptive control is not the only way to deal with parameter variations. Robust control is an alternative. A robust controller is a controller that can satisfactorily control a class of system with specified uncertainties in the process model. To have a balanced view of adaptive techniques, it is therefore necessary to know these methods as well (see Chapter 10). Notice particularly that there are few alternatives to adaptation for feedforward control of processes with varying dynamics.

Engineering efficiency is an often overlooked argument in the choice between different techniques. It may be advantageous to trade engineering efforts against more "intelligence" in the controller. This tradeoff is one reason for the success of automatic tuning. When a control loop can be tuned simply by pushing a button, it is easy to commission control systems and to keep them running well. This also makes it possible to use a more complex controller like feedforward. With toolboxes for adaptive control (such as ABB Master) it is often a simple matter to configure an adaptive control system and to try it experimentally. This can be much less time-consuming than the alternative path of modeling, design, and implementation of a conventional control system. The knowledge required to build and use toolboxes for adaptive control is given in the chapters that follow. It should be emphasized that typical industrial processes are so complex that the parameter variations cannot be determined from first principles.

A more complex controller may be used on different processes, and the development expenses can be shared by many applications. However, it should be pointed out that the use of an adaptive controller will not replace good process knowledge, which is still needed to choose the specifications, the structure of the controller, and the design method.

PROBLEMS

1.1 Look up the definitions of "adaptive" and "learning" in a good dictionary. Compare the uses of the words in different fields.

1.2 Find descriptions of adaptive controllers from some manufacturers and browse through them.

1.3 Give some situations in which adaptive control may be useful. What factors would you consider when judging the need for adaptive control?

1.4 Make an assessment of the field of adaptive control by making a literature search. Look for the distribution of publications on adaptive control over the years. Can you see some pattern in the publications concerning uses of different methods, emphasis on theory and applications, and so on?

1.5 The system in Example 1.4 has the following characteristics:

$$G_0(s) = \frac{1}{(s+1)^3}$$
$$f(u) = u^4$$

The PI controller has the gain $K = 0.15$ and the reset time $T_i = 1$. Linearize the equations when the reference values are $u_c = 0.3$, 1.1, and 5.1. Determine the roots of the characteristic equation in the different cases. Determine a reference value such that the linearized equations just become unstable.

1.6 Consider the concentration control system in Example 1.5. Assume that $V_d = V_m = 1$ and that the nominal flow is $q = 1$. Determine PI controllers with the transfer function

$$K_c\left(1 + \frac{1}{T_i s}\right)$$

that give good closed-loop performance for the flows $q = 0.5$, 1, and 2. Test the controllers for the nominal flow.

1.7 Consider the following system with two inputs and two outputs:

$$\frac{dx}{dt} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & -1 \end{pmatrix} x + \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 0 & 1 \end{pmatrix} u$$

$$y = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} x$$

Assume that proportional feedback is introduced around the second loop:

$$u_2 = -k_2 y_2$$

(a) Determine the transfer function from u_1 to y_1, and determine how the steady-state gain depends on k_2.

(b) Simulate the response of y_1 and y_2 when u_1 is a step for different values of k_2.

1.8 A block diagram of a system used for metal cutting on a numerically controlled machine is shown in Fig. 1.25. The machine is equipped with a force sensor, which measures the cutting force. A controller adjusts the feedback to maintain a constant cutting force. The cutting force is approximately given by

$$F = k\,a\left(\frac{v}{N}\right)^{\alpha}$$

where a is the depth of the cut, v is the feed rate, N is the spindle speed, α is a parameter in the range $0.5 < \alpha < 1$, and k is a positive parameter.

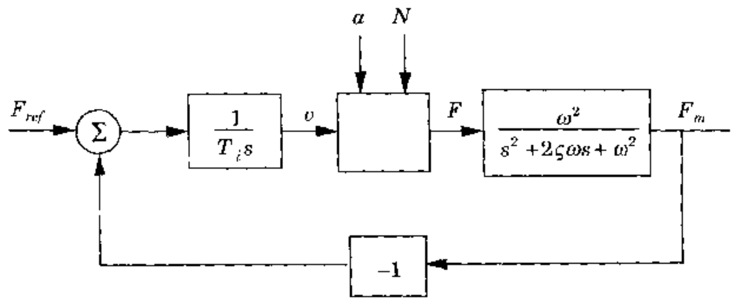

Figure 1.25 Block diagram of a control system for metal cutting.

The steady-state gain from feed rate to force is

$$K = k\,\alpha\,a\,v^{\alpha-1} N^{-\alpha}$$

The gain increases with increasing depth a, decreasing feed rate v, and decreasing spindle speed N. Assume that $\alpha = 0.7$, $k = 1$, $a = 1$, $\zeta = 0.7$, and $\omega = 5$. Determine T_i such that the closed-loop system shows good closed-loop behavior for $N = 1$ and $a = 1$.

(a) Investigate the performance of the closed-loop system when N varies between 0.2 and 2 and $a = 1$.

(b) Repeat part (a) but for a varying between 0.5 and 4 and $N = 1$.

1.9 Consider the system in Fig. 1.26. Let the process be

$$G_0(s) = \frac{K}{s+a}$$

where

$$K = K_0 + \Delta K \qquad K_0 = 1$$
$$a = a_0 + \Delta a \qquad a_0 = 1$$

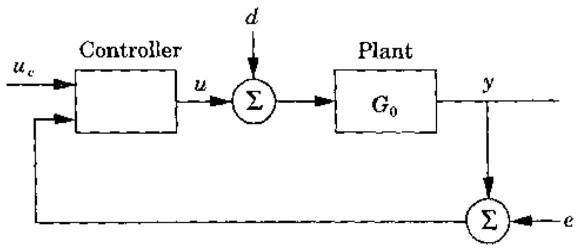

Figure 1.26 Block diagram for Problems 1.9 and 1.10.

and

$$-0.5 \le \Delta K \le 2.0$$
$$-2.0 \le \Delta a \le 2.0$$

Let the ideal closed-loop response be given by

$$Y_m(s) = \frac{1}{s+1} U_c(s)$$

(a) Simulate the open-loop responses for some values of K and a.

(b) Determine a controller for the nominal system such that the difference between step responses of the closed-loop system and of the desired system is less than 1% of the magnitude of the step.

(c) Use the controller from part (b) and investigate the sensitivity to parameter changes.

(d) Use the controller from part (b) and investigate the sensitivity to the disturbance $d(t)$ when

$$d(t) = \begin{cases} -1 & 0 \le t < 6 \\ 2 & 6 \le t < 15 \\ 1 & 15 \le t \end{cases}$$

(e) Use the controller from part (b) and investigate the influence of measurement noise, $e(t)$. Let $e(t)$ be zero mean white noise.

This problem and the next example are based on a special session at the 1988 American Control Conference in Atlanta, Georgia. A detailed discussion of the problem is found in *International Journal of Adaptive Control and Signal Processing*, No. 2, June 1989, which is entirely devoted to the problem.

1.10 Make the same investigation as in Problem 1.9 when the process is

$$G_0(s) = \frac{K}{s^2 + a_1 s + a_2}$$

where

$$K = K_0 + \Delta K \qquad K_0 = 1$$
$$a_1 = a_{10} + \Delta a_1 \qquad a_{10} = 1.4$$
$$a_2 = a_{20} + \Delta a_2 \qquad a_{20} = 1$$

and

$$-0.5 \le \Delta K \le 2.0$$
$$-2.0 \le \Delta a_1 \le 2.0$$
$$-3.0 \le \Delta a_2 \le 3.0$$

Let the desired closed-loop response be given by

$$Y_m(s) = \frac{1}{s^2 + 1.4s + 1} U_c(s)$$

REFERENCES

Many papers, books, and reports have been written on adaptive control. Some of the earlier developments are found in:

Kalman, R. E., 1958. "Design of Self-optimizing Control Systems." *ASME Transactions* **80**: 468–478.

Gregory, P. C., ed., 1959. *Proc. Self Adaptive Flight Control Symposium.* Wright-Patterson Air Force Base, Ohio: Wright Air Development Center.

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

Mishkin, E., and L. Braun, 1961. *Adaptive Control Systems.* New York: McGraw-Hill.

Tsypkin, Y. Z., 1971. *Adaptation and Learning in Automatic Systems.* New York: Academic Press.

The conference proceedings edited by Gregory is an interesting historical document. The papers and the discussions quoted give a good perspective on early research on adaptive control. Most schemes in the conference are also found in the book by Mishkin and Braun. Bellman's book is still interesting reading. The relation to learning is emphasized both in this book and in the book by Tsypkin. Reprints of many original papers are found in:

Gupta, M. M., ed., 1986. *Adaptive Methods for Control System Design.* New York: IEEE Press.

Narendra, K. S., R. Ortega, and P. Dorato, eds., 1991. *Advances in Adaptive Control.* New York: IEEE Press.

There are several good survey papers on adaptive control:

Åström, K. J., 1983. "Theory and applications of adaptive control—A survey." *Automatica* **19**: 471–486.

Kumar, P. R., 1985. "A survey of some results in stochastic adaptive control." *SIAM J. Control and Opt.* **23**: 329–380.

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.

Ioannou, P. A., and A. Datta, 1991. "Robust Adaptive Control: A Unified Approach." *Proc. IEEE* **79**: 1736–1768.

Among the textbooks in adaptive control we can mention:

Narendra, K. S., and R. V. Monopoli, eds., 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, 1981. *Self-tuning and Adaptive Control: Theory and Applications.* London: Peter Peregrinus.

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

Anderson, B. D. O., R. R. Bitmead, C. R. Johnson, 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.

Gawthrop, P. J., 1986. *Continuous Time Self-Tuning Control.* Letchworth, U.K.: Research Studies Press.

Narendra, K. S., and A. M. Annaswamy, 1989. *Stable Adaptive Systems.* Englewood Cliffs, N.J.: Prentice-Hall.

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

Wellstead, P. E., and M. B. Zarrop, 1991. *Selftuning Systems: Control and Signal Processing.* Chichester, U.K.: John Wiley & Sons.

Isermann, R., K.-H. Lachmann, and D. Matko, 1992. *Adaptive Control Systems.* Hemel Hempstead, U.K.: Prentice-Hall International.

Recent developments with particular emphasis on nonlinear systems are discussed in:

Kokotovic, P. V., ed., 1991. *Foundations of Adaptive Control.* Berlin: Springer-Verlag.

There are normally sessions on adaptive control at the major control conferences. The International Federation of Automatic Control is responsible for the Symposium on Adaptive Systems in Control and Signal Processing (ACASP), which is held every third year. The first symposium was held in San Francisco in 1983. These symposia provide up-to-date information about progess in the field. There are few discussions of when to use adaptive control in the literature. Some papers in which this is discussed are:

Åström, K. J., 1980. "Why use adaptive techniques for steering large tankers?" *Int. J. Control* **32**: 689–708.

Jacobs, O. L. R., 1981. "When is adaptive control useful?" *Proceedings Third IMA Conference on Control Theory.* New York: Academic Press.

Flight control systems are usually based on gain scheduling. Feasibility studies of using adaptive control for airplane control are reported in:

IEEE, 1977. "Mini-issue on NASA's advanced control law program for the F-8 DFBW aircraft." *IEEE Trans. Automat. Contr.* **AC-22**: 752–806.

A discussion of adaptive flight control is found 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.

The airplane problem in Example 1.6 is taken from:

Ackermann, J., 1983. *Abtastregelung Band II: Entwurf robuster Systeme.* Berlin: Springer-Verlag.

Robust high-gain control is thoroughly discussed in:

Horowitz, I. M., 1963. *Synthesis of Feedback Systems.* New York: Academic Press.

Horowitz's book contains the foundation of feedback control systems synthesis in the frequency domain, including benefits and disadvantages of feedback, parameter-uncertain systems, tolerances and specification, and reasoning about slowly varying parameters. Basic background material for feedback and sensitivity is found in:

Bode, H. W., 1945. *Network Analysis and Feedback Amplifier Design.* New York: Van Nostrand.

Unstructured perturbations are discussed 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.

A survey of linear quadratic Gaussian design and its robustness properties is found in:

Stein, G., and M. Athans, 1987. "The LQG/LTR procedure for multivariable feedback control design." *IEEE Trans. Automat. Contr.* **AC-32**: 105–114.

Other references on robustness and sensitivity are:

Zames, G., 1981. "Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms and approximate inverses." *IEEE Trans. Automat. Contr.* **AC-26**: 301–320.

Zames, G., and B. A. Francis, 1983. "Feedback, minimax sensitivity and optimal robustness." *IEEE Trans. Automat. Contr.* **AC-28**: 585–601.

Morari, M., and E. Zafiriou, 1989. *Robust Process Control.* Englewood Cliffs, N.J.: Prentice-Hall.

Doyle, J. C., B. A. Francis, and A. R. Tannenbaum, 1992. *Feedback Control Theory.* New York: Macmillan.

CHAPTER 2

REAL-TIME PARAMETER ESTIMATION

2.1 INTRODUCTION

On-line determination of process parameters is a key element in adaptive control. A recursive parameter estimator appears explicitly as a component of a self-tuning regulator (see Fig. 1.19). Parameter estimation also occurs implicitly in a model-reference adaptive controller (see Fig. 1.18). This chapter presents some methods for real-time parameter estimation. It is useful to view parameter estimation in the broader context of system identification. The key elements of system identification are selection of model structure, experiment design, parameter estimation, and validation. Since system identification is executed automatically in adaptive systems, it is essential to have a good understanding of all aspects of the problem. Selection of model structure and parameterization are fundamental issues. Simple transfer function models will be used in this chapter. The identification problems are simplified significantly if the models are linear in the parameters.

The experiment design is crucial for successful system identification. In control problems this boils down to selection of the input signal. Choosing an input signal requires some knowledge of the process and the intended use of the model. In adaptive systems there is an additional complication because the input signal to the plant is generated by feedback. In certain cases this does not permit the parameters to be determined uniquely, a situation that has far-reaching consequences. In some cases it may be necessary to introduce perturbation signals, as discussed in more detail in Chapters 6 and 7. In adaptive control the parameters of a process change continuously, so it is necessary to have estimation methods that update the parameters recursively.

In solving identification problems it is very important to validate the results. This is especially important for adaptive systems, in which identification is done automatically. Some validation techniques will therefore be discussed.

The least-squares method is a basic technique for parameter estimation. The method is particularly simple if the model has the property of being *linear in the parameters*. In this case the least-squares estimate can be calculated analytically. A compact presentation of the method of least squares is given in Section 2.2. The formulas for the estimate are derived, and geometric and statistical interpretations are given. It is shown how the computations can be done recursively. In Section 2.3 it is shown how the least-squares method can be used to estimate parameters in dynamical systems. Experimental conditions are discussed in Section 2.4. In particular we introduce the notion of persistent excitation. In using parameter estimation in adaptive control it is useful to have an intuitive insight into the properties of parameter estimators. To start to develop this, we give a number of simulations that illustrate the properties of the different algorithms in Section 2.5. More properties of different estimation schemes are given in Chapter 6 in connection with convergence and stability analysis of adaptive controllers.

2.2 LEAST SQUARES AND REGRESSION MODELS

Karl Friedrich Gauss formulated the principle of least squares at the end of the eighteenth century and used it to determine the orbits of planets and asteroids. Gauss stated that, according to this principle, the unknown parameters of a mathematical model should be chosen in such a way that

> the sum of the squares of the differences between the actually observed and the computed values, multiplied by numbers that measure the degree of precision, is a minimum.

The least-squares method can be applied to a large variety of problems. It is particularly simple for a mathematical model that can be written in the form

$$y(i) = \varphi_1(i)\theta_1^0 + \varphi_2(i)\theta_2^0 + \cdots + \varphi_n(i)\theta_n^0 = \varphi^T(i)\theta^0 \tag{2.1}$$

where y is the observed variable, $\theta_1^0, \theta_2^0, \ldots, \theta_n^0$ are parameters of the model to be determined, and φ_1, φ_2, ..., φ_n are known functions that may depend on other known variables. The vectors

$$\varphi^T(i) = \begin{pmatrix} \varphi_1(i) & \varphi_2(i) & \ldots & \varphi_n(i) \end{pmatrix}$$
$$\theta^0 = \begin{pmatrix} \theta_1^0 & \theta_2^0 & \ldots & \theta_n^0 \end{pmatrix}^T$$

have also been introduced. The model is indexed by the variable i, which often denotes time. It will be assumed initially that the index set is a discrete set. The variables φ_i are called the *regression variables* or the *regressors*, and the model in Eq. (2.1) is also called a *regression model*. Pairs of observations and regressors $\{(y(i), \varphi(i)), i = 1, 2, \ldots, t\}$ are obtained from an experiment. The problem is to determine the parameters in such a way that the outputs computed from the model in Eq. (2.1) agree as closely as possible with the measured variables $y(i)$ in the sense of least squares. That is, the parameter θ should be chosen to minimize the least-squares loss function

$$V(\theta, t) = \frac{1}{2}\sum_{i=1}^{t}\left(y(i) - \varphi^T(i)\theta\right)^2 \tag{2.2}$$

Since the measured variable y is linear in parameters θ^0 and the least-squares criterion is quadratic, the problem admits an analytical solution. Introduce the notations

$$Y(t) = \begin{pmatrix} y(1) & y(2) & \ldots & y(t) \end{pmatrix}^T$$

$$\mathcal{E}(t) = \begin{pmatrix} \varepsilon(1) & \varepsilon(2) & \ldots & \varepsilon(t) \end{pmatrix}^T$$

$$\Phi(t) = \begin{pmatrix} \varphi^T(1) \\ \vdots \\ \varphi^T(t) \end{pmatrix}$$

$$P(t) = \left(\Phi^T(t)\Phi(t)\right)^{-1} = \left(\sum_{i=1}^{t}\varphi(i)\varphi^T(i)\right)^{-1} \tag{2.3}$$

where the *residuals* $\varepsilon(i)$ are defined by

$$\varepsilon(i) = y(i) - \hat{y}(i) = y(i) - \varphi^T(i)\theta$$

With these notations the loss function (2.2) can be written as

$$V(\theta, t) = \frac{1}{2}\sum_{i=1}^{t}\varepsilon^2(i) = \frac{1}{2}\mathcal{E}^T\mathcal{E} = \frac{1}{2}\|\mathcal{E}\|^2$$

where $\mathcal{E}$ can be written as

$$\mathcal{E} = Y - \hat{Y} = Y - \Phi\theta \tag{2.4}$$

The solution to the least-squares problem is given by the following theorem.

THEOREM 2.1 Least-squares estimation

The function of Eq. (2.2) is minimal for parameters $\hat{\theta}$ such that

$$\Phi^T\Phi\hat{\theta} = \Phi^T Y \tag{2.5}$$

If the matrix $\Phi^T\Phi$ is nonsingular, the minimum is unique and given by

$$\hat{\theta} = (\Phi^T\Phi)^{-1}\Phi^T Y \tag{2.6}$$

Proof: The loss function of Eq. (2.2) can be written as

$$\begin{aligned} 2V(\theta, t) &= \mathcal{E}^T\mathcal{E} = (Y - \Phi\theta)^T(Y - \Phi\theta) \\ &= Y^TY - Y^T\Phi\theta - \theta^T\Phi^TY + \theta^T\Phi^T\Phi\theta \end{aligned} \tag{2.7}$$

Since the matrix $\Phi^T\Phi$ is always nonnegative definite, the function V has a minimum. The loss function is quadratic in θ. The minimum can be found in many ways. One way is to determine the gradient of Eq. (2.7) with respect to θ. (See Problem 2.1 at the end of the chapter). The gradient is zero when Eq. (2.5) is satisfied. Another way to find the minimum is by completing the square. We get

$$\begin{aligned} 2V(\theta, t) &= Y^TY - Y^T\Phi\theta - \theta^T\Phi^TY + \theta^T\Phi^T\Phi\theta \\ &\quad + Y^T\Phi(\Phi^T\Phi)^{-1}\Phi^TY - Y^T\Phi(\Phi^T\Phi)^{-1}\Phi^TY \\ &= Y^T\left(I - \Phi(\Phi^T\Phi)^{-1}\Phi^T\right)Y \\ &\quad + \left(\theta - (\Phi^T\Phi)^{-1}\Phi^TY\right)^T\Phi^T\Phi\left(\theta - (\Phi^T\Phi)^{-1}\Phi^TY\right) \end{aligned} \tag{2.8}$$

The first term on the right-hand side is independent of θ. The second term is always positive. The minimum is obtained for

$$\theta = \hat{\theta} = (\Phi^T\Phi)^{-1}\Phi^TY$$

and the theorem is proven. □

Remark 1. Equation (2.5) is called the *normal equation*. Equation (2.6) can be written as

$$\hat{\theta}(t) = \left(\sum_{i=1}^{t}\varphi(i)\varphi^T(i)\right)^{-1}\left(\sum_{i=1}^{t}\varphi(i)y(i)\right) = P(t)\left(\sum_{i=1}^{t}\varphi(i)y(i)\right) \tag{2.9}$$

Remark 2. The condition that the matrix $\Phi^T\Phi$ is invertible is called an *excitation* condition.

Remark 3. The least-squares criterion weights all errors $\varepsilon(i)$ equally, and this corresponds to the assumption that all measurements have the same precision. □

Different weighting of the errors can be accounted for by changing the loss function (2.2) to

$$V = \frac{1}{2}\mathcal{E}^TW\mathcal{E} \tag{2.10}$$

where W is a diagonal matrix with the weights in the diagonal. The least-squares estimate is then given by

$$\hat{\theta} = \left(\Phi^TW\Phi\right)^{-1}\Phi^TWY \tag{2.11}$$

EXAMPLE 2.1 Least-squares estimation of static system

Consider the system

$$y(i) = b_0 + b_1 u(i) + b_2 u^2(i) + e(i)$$

where $e(i)$ is zero mean Gaussian noise with standard deviation 0.1. The system is linear in the parameters and can be written in the form (2.1) with

$$\varphi^T(i) = \begin{pmatrix} 1 & u(i) & u^2(i) \end{pmatrix}$$
$$\theta^T = \begin{pmatrix} b_0 & b_1 & b_2 \end{pmatrix}$$

The output is measured for the seven different inputs shown by the dots in Fig. 2.1. In practice the model structure is usually unknown, and the user must decide on an appropriate model. We illustrate this by estimating parameters of the following models:

$$\begin{aligned} \text{Model 1}: \quad y(i) &= b_0 \\ \text{Model 2}: \quad y(i) &= b_0 + b_1 u \\ \text{Model 3}: \quad y(i) &= b_0 + b_1 u + b_2 u^2 \\ \text{Model 4}: \quad y(i) &= b_0 + b_1 u + b_2 u^2 + b_3 u^3 \end{aligned}$$

The different models give a polynomial dependence of different orders between y and u.

Table 2.1 shows the least-squares estimates of the different models together with the resulting loss function. Figure 2.1 also shows the estimated relation between u and y for the different models. From the table it is seen that about the same losses are obtained for Models 3 and 4. The fit to the data points is almost the same for these two models, as is seen in Fig. 2.1. □

The example shows that it is important to choose the correct model structure to get a good model. With few parameters it is not possible to get a good fit to the data. If too many parameters are used, the fit to the measured data will be very good but the fit to another data set may be very poor. This latter situation is called *overfitting*.

Table 2.1 Least-squares estimates and loss functions for the system in Example 2.1 using different model structures.

Model	$\hat{b}_0$	$\hat{b}_1$	$\hat{b}_2$	$\hat{b}_3$	V
1	3.85				34.46
2	0.57	1.09			1.01
3	1.11	0.45	0.11		0.031
4	1.13	0.37	0.14	−0.003	0.027

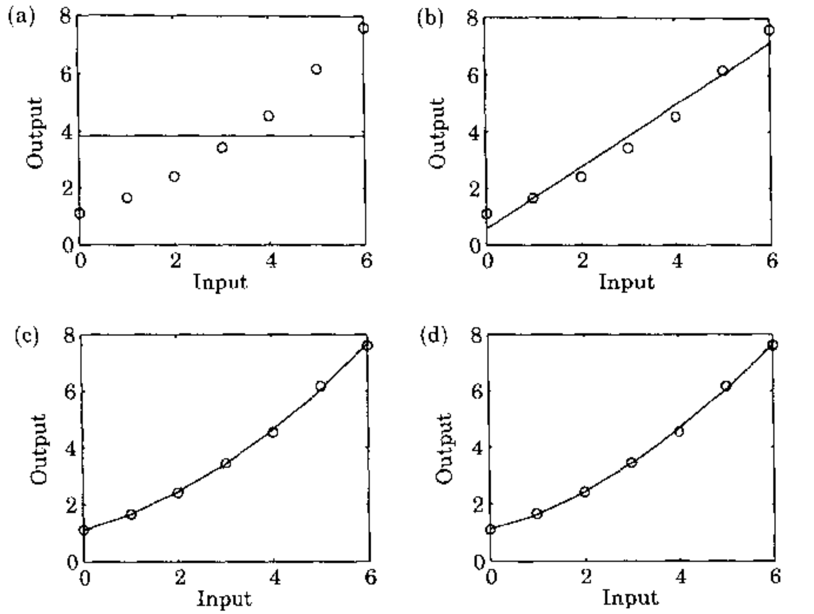

Figure 2.1 The dots represent the measured data points. Resulting models, indicated by the solid lines, based on the least-squares estimates are also given for (a) Model 1, (b) Model 2, (c) Model 3, (d) Model 4.

Geometric Interpretation

The least-squares problem can be interpreted as a geometric problem in R^t, where t is the number of observations. Figure 2.2 illustrates the situation with two parameters and three observations. The vectors φ^1 and φ^2 spans a plane if they are linearly independent. The predicted output $\hat{Y}$ lies in the plan spanned by φ^1 and φ^2. The error $\mathcal{E} = Y - \hat{Y}$ is smallest when $\mathcal{E}$ is orthogonal to this plane. In the general case, Eq. (2.4) can be written as

$$\begin{pmatrix} \varepsilon(1) \\ \varepsilon(2) \\ \vdots \\ \varepsilon(t) \end{pmatrix} = \begin{pmatrix} y(1) \\ y(2) \\ \vdots \\ y(t) \end{pmatrix} - \begin{pmatrix} \varphi_1(1) \\ \varphi_1(2) \\ \vdots \\ \varphi_1(t) \end{pmatrix} \theta_1 - \cdots - \begin{pmatrix} \varphi_n(1) \\ \varphi_n(2) \\ \vdots \\ \varphi_n(t) \end{pmatrix} \theta_n$$

or

$$\mathcal{E} = Y - \varphi^1\theta_1 - \varphi^2\theta_2 - \cdots - \varphi^n\theta_n$$

where φ^i are the columns of the matrix Φ. The least-squares problem can thus be interpreted as the problem of finding constants $\theta_1, \ldots, \theta_n$ such that

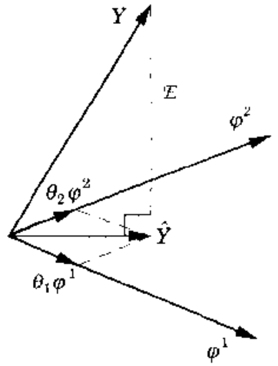

Figure 2.2 Geometric interpretation of the least-squares estimate.

the vector Y is approximated as well as possible by a linear combination of the vectors $\varphi^1, \varphi^2, \ldots, \varphi^n$. Let $\hat{Y}$ be the vector in the span of $\varphi^1, \varphi^2, \ldots, \varphi^n$, which is the best approximation, and let $\mathcal{E} = Y - \hat{Y}$. The vector $\mathcal{E}$ is smallest when it is orthogonal to all vectors φ^i. This gives

$$(\varphi^i)^T \left(Y - \theta_1\varphi^1 - \theta_2\varphi^2 - \cdots - \theta_n\varphi^n\right) = 0 \qquad i = 1, \ldots, t$$

which is identical to the normal equation (2.5). The vector θ is unique if the vectors $\varphi^1, \varphi^2, \ldots, \varphi^n$ are linearly independent.

Statistical Interpretation

The least-squares method can be interpreted in statistical terms. It is then necessary to make assumptions about how the data has been generated. Assume that the process is

$$y(i) = \varphi^T(i)\theta^0 + e(i) \tag{2.12}$$

where θ^0 is the vector of "true" parameters and $\{e(i), i = 1, 2, \ldots\}$ is a sequence of independent, equally distributed random variables with zero mean. It is also assumed that e is independent of φ. Equation (2.4) can be written as

$$Y = \Phi\theta^0 + \mathcal{E}$$

Multiplying by $(\Phi^T\Phi)^{-1}\Phi^T$ gives

$$\left(\Phi^T\Phi\right)^{-1}\Phi^T Y = \hat{\theta} = \theta^0 + \left(\Phi^T\Phi\right)^{-1}\Phi^T\mathcal{E} \tag{2.13}$$

Provided that $\mathcal{E}$ is independent of Φ^T, which is equivalent to saying that $e(i)$ is independent of $\varphi(i)$, the mathematical expectation of $\hat{\theta}$ is equal to θ^0. An estimate with this property is called unbiased. The following theorem is given without proof.

THEOREM 2.2 Statistical properties of least-squares estimation

Consider the estimate in Eq. (2.6) and assume that data is generated from Eq. (2.12), where $\{e(i), i = 1, 2, \ldots\}$ is a sequence of independent random variables with zero mean and variance σ^2. Let E denote mathematical expectation and cov the covariance of a random variable.

If $\Phi^T\Phi$ is nonsingular, then

(i) $E\hat{\theta}(t) = \theta^0$

(ii) $\operatorname{cov} \hat{\theta}(t) = \sigma^2(\Phi^T\Phi)^{-1}$

(iii) $\hat{\sigma}^2(t) = 2V(\hat{\theta}, t)/(t - n)$ is an unbiased estimate of σ^2

where n is the number of parameters in θ^0 and $\hat{\theta}$ and t is the number of data points. □

The theorem states that the estimates are unbiased, that is, $E\hat{\theta}(t) = \theta^0$. Further, it is desirable that an estimate converge to the true parameter value as the number of observations increases toward infinity. This property is called *consistency*. There are several notions of consistency corresponding to different convergence concepts for random variables. Mean square convergence is one possibility, which can be investigated simply by analyzing the variance of the estimate. The result (ii) can be used to determine how the variance of the estimate decreases with the number of observations. This is illustrated by an example.

EXAMPLE 2.2 Decrease of variance

Consider the case in which the model in Eq. (2.12) has only one parameter. Let t be the number of observations. It follows from (ii) of Theorem 2.2 that the variance of the estimate is given by

$$\operatorname{cov} \hat{\theta} = \frac{\sigma^2}{\sum_{k=1}^{t} \varphi^2(k)}$$

Several different cases can now be considered, depending on the asymptotic behavior of $\varphi(k)$ for large k. Introduce the notation $a \sim b$ to indicate that a and b are proportional.

(a) Assume that $\varphi(k) \sim e^{-\alpha k}, \alpha > 0$. The sum in the denominator above then converges, and the variance goes to a constant.

(b) Assume that $\varphi(k) \sim k^{-a}, a > 0$. Then

$$\sum_{k=1}^{t} \varphi^2(k) \sim \begin{cases} \text{const} & a > 0.5 \\ \log t & a = 0.5 \\ t^{1-2a} & a < 0.5 \end{cases}$$

The variance goes to zero if $a \leq 0.5$.

(c) Assume that $\varphi(k) \sim 1$. The variance then goes to zero as $1/t$.

(d) Assume that $\varphi(k) \sim k^a, a > 0$. The variance then goes to zero as $t^{-(1+2a)}$.

(e) Assume that $\varphi(k) \sim e^{\alpha k}, \alpha > 0$. The variance then goes to zero as $e^{-2\alpha t}$.

□

The example shows clearly how the precision of the estimate depends on the rate of growth of the regression vector. The variance does not go to zero with increasing number of observations if the regression variable decreases faster than $1/\sqrt{t}$. In the normal situation, when the regressors are of the same order of magnitude, the variance decreases as $1/t$. The variance decreases more rapidly if the regression variables increase with time.

When several parameters are estimated, the convergence rates may be different for different parameters. This is related to the structure of the matrix $(\Phi^T\Phi)^{-1}$ in Eq. (2.6).

Recursive Computations

In adaptive controllers the observations are obtained sequentially in real time. It is then desirable to make the computations recursively to save computation time. Computation of the least-squares estimate can be arranged in such a way that the results obtained at time $t-1$ can be used to get the estimates at time t. The solution in Eq. (2.6) to the least-squares problem will be rewritten in a recursive form. Let $\hat{\theta}(t-1)$ denote the least-squares estimate based on $t-1$ measurements. Assume that the matrix $\Phi^T\Phi$ is nonsingular for all t. It follows from the definition of $P(t)$ in Eq. (2.3) that

$$\begin{aligned} P^{-1}(t) &= \Phi^T(t)\Phi(t) = \sum_{i=1}^{t} \varphi(i)\varphi^T(i) \\ &= \sum_{i=1}^{t-1} \varphi(i)\varphi^T(i) + \varphi(t)\varphi^T(t) \\ &= P^{-1}(t-1) + \varphi(t)\varphi^T(t) \end{aligned} \tag{2.14}$$

The least-squares estimate $\hat{\theta}(t)$ is given by Eq. (2.9):

$$\hat{\theta}(t) = P(t)\left(\sum_{i=1}^{t} \varphi(i)y(i)\right) = P(t)\left(\sum_{i=1}^{t-1} \varphi(i)y(i) + \varphi(t)y(t)\right)$$

It follows from Eqs. (2.9) and (2.14) that

$$\sum_{i=1}^{t-1} \varphi(i)y(i) = P^{-1}(t-1)\hat{\theta}(t-1) = P^{-1}(t)\hat{\theta}(t-1) - \varphi(t)\varphi^T(t)\hat{\theta}(t-1)$$

The estimate at time t can now be written as

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) - P(t)\varphi(t)\varphi^T(t)\hat{\theta}(t-1) + P(t)\varphi(t)y(t) \\ &= \hat{\theta}(t-1) + P(t)\varphi(t)\left(y(t) - \varphi^T(t)\hat{\theta}(t-1)\right) \\ &= \hat{\theta}(t-1) + K(t)\varepsilon(t) \end{aligned}$$

where

$$\begin{aligned} K(t) &= P(t)\varphi(t) \\ \varepsilon(t) &= y(t) - \varphi^T(t)\hat{\theta}(t-1) \end{aligned}$$

The residual $\varepsilon(t)$ can be interpreted as the error in predicting the signal $y(t)$ one step ahead based on the estimate $\hat{\theta}(t-1)$.

To proceed, it is necessary to derive a recursive equation for $P(t)$ rather than for $P(t)^{-1}$ as in Eq. (2.14). The following lemma is useful.

LEMMA 2.1 Matrix inversion lemma

Let A, C, and $C^{-1} + DA^{-1}B$ be nonsingular square matrices. Then $A + BCD$ is invertible, and

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

Proof: By direct multiplication we find that

$$\begin{aligned} &(A + BCD)\left(A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}\right) \\ &\quad = I + BCDA^{-1} - B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \\ &\qquad - BCDA^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \\ &\quad = I + BCDA^{-1} - BC(C^{-1} + DA^{-1}B)(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \\ &\quad = I \end{aligned}$$

□

Applying Lemma 2.1 to $P(t)$ and using Eq. (2.14), we get

$$\begin{aligned} P(t) &= \left(\Phi^T(t)\Phi(t)\right)^{-1} = \left(\Phi^T(t-1)\Phi(t-1) + \varphi(t)\varphi^T(t)\right)^{-1} \\ &= \left(P(t-1)^{-1} + \varphi(t)\varphi^T(t)\right)^{-1} \\ &= P(t-1) - P(t-1)\varphi(t)\left(I + \varphi^T(t)P(t-1)\varphi(t)\right)^{-1}\varphi^T(t)P(t-1) \end{aligned}$$

This implies that

$$K(t) = P(t)\varphi(t) = P(t-1)\varphi(t)\left(I + \varphi^T(t)P(t-1)\varphi(t)\right)^{-1}$$

Notice that a matrix inversion is necessary to compute P. However, the matrix to be inverted is of the same dimension as the number of measurements. That is, for a single output system it is a scalar.

The recursive calculations are summarized in the following theorem.

THEOREM 2.3 Recursive least-squares estimation (RLS)
Assume that the matrix $\Phi(t)$ has full rank, that is, $\Phi^T(t)\Phi(t)$ is nonsingular, for all $t \geq t_0$. Given $\hat{\theta}(t_0)$ and $P(t_0) = (\Phi^T(t_0)\Phi(t_0))^{-1}$, the least-squares estimate $\hat{\theta}(t)$ then satisfies the recursive equations

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\Big(y(t) - \varphi^T(t)\hat{\theta}(t-1)\Big) \tag{2.15}$$

$$K(t) = P(t)\varphi(t) = P(t-1)\varphi(t)\Big(I + \varphi^T(t)P(t-1)\varphi(t)\Big)^{-1} \tag{2.16}$$

$$\begin{aligned} P(t) &= P(t-1) - P(t-1)\varphi(t)\Big(I + \varphi^T(t)P(t-1)\varphi(t)\Big)^{-1}\varphi^T(t)P(t-1) \\ &= \Big(I - K(t)\varphi^T(t)\Big)P(t-1) \end{aligned} \tag{2.17}$$

□

Remark 1. Equation (2.15) has strong intuitive appeal. The estimate $\hat{\theta}(t)$ is obtained by adding a correction to the previous estimate $\hat{\theta}(t-1)$. The correction is proportional to $y(t) - \varphi^T(t)\hat{\theta}(t-1)$, where the last term can be interpreted as the value of y at time t predicted by the model of Eq. (2.1). The correction term is thus proportional to the difference between the measured value of $y(t)$ and the prediction of $y(t)$ based on the previous parameter estimate. The components of the vector $K(t)$ are weighting factors that tell how the correction and the previous estimate should be combined.

Remark 2. The least-squares estimate can be interpreted as a Kalman filter for the process

$$\begin{aligned} \theta(t+1) &= \theta(t) \\ y(t) &= \varphi^T(t)\theta(t) + e(t) \end{aligned} \tag{2.18}$$

Remark 3. The recursive equations can also be derived by starting with the loss function of Eq. (2.2). Using Eqs. (2.8) and (2.6) gives

$$\begin{aligned} 2V(\theta,t) &= 2V(\theta,t-1) + \varepsilon^2(\theta,t) \\ &= Y^T(t-1)\left(I - \Phi(t-1)\Big(\Phi^T(t-1)\Phi(t-1)\Big)^{-1}\Phi(t-1)\right)Y(t-1) \\ &\quad + \Big(\theta - \hat{\theta}(t-1)\Big)^T\Phi^T(t-1)\Phi(t-1)\Big(\theta - \hat{\theta}(t-1)\Big) \\ &\quad + \Big(y(t) - \varphi^T(t)\theta\Big)^T\Big(y(t) - \varphi^T(t)\theta\Big) \end{aligned} \tag{2.19}$$

The first term on the right-hand side is independent of θ, and the remaining two terms are quadratic in θ. $V(\theta,t)$ can then easily be minimized with respect to θ. □

Notice that the matrix $P(t)$ is defined only when the matrix $\Phi^T(t)\Phi(t)$ is nonsingular. Since

$$\Phi^T(t)\Phi(t) = \sum_{i=1}^{t}\varphi(i)\varphi^T(i)$$

it follows that $\Phi^T\Phi$ is always singular if $t < n$. To obtain an initial condition for P, it is thus necessary to choose $t = t_0$ such that $\Phi^T(t_0)\Phi(t_0)$ is nonsingular. The initial conditions are then

$$\begin{aligned} P(t_0) &= \Big(\Phi^T(t_0)\Phi(t_0)\Big)^{-1} \\ \hat{\theta}(t_0) &= P(t_0)\Phi^T(t_0)Y(t_0) \end{aligned}$$

The recursive equations can then be used for $t > t_0$. It is, however, often convenient to use the recursive equations in all steps. If the recursive equations are started with the initial condition

$$P(0) = P_0$$

where P_0 is positive definite, then

$$P(t) = \Big(P_0^{-1} + \Phi^T(t)\Phi(t)\Big)^{-1}$$

Notice that $P(t)$ can be made arbitrarily close to $\big(\Phi^T(t)\Phi(t)\big)^{-1}$ by choosing P_0 sufficiently large.

By using the Kalman filter interpretation of the least-squares method, it may be seen that this way of starting the recursion corresponds to the situation in which the parameters have an initial distribution with mean θ_0 and covariance P_0.

Time-Varying Parameters

In the least-squares model (2.1) the parameters θ_i^0 are assumed to be constant. In several adaptive problems it is of interest to consider the situation in which the parameters are time-varying. Two cases can be covered by simple extensions of the least-squares method. In one such case parameters are assumed to change abruptly but infrequently; in the other case the parameters are changing continuously but slowly. The case of abrupt parameter changes can be covered by *resetting*. The matrix P in the least-squares algorithm (Theorem 2.3) is then periodically reset to αI, where α is a large number. This implies that the gain $K(t)$ in the estimator becomes large and the estimate can be updated with a larger step. A more sophisticated version is to run n estimators in parallel, which are reset sequentially. The estimate is then chosen by using some decision logic. (See Chapter 6.) The case of slowly time-varying parameters can be covered by relatively simple mathematical models. One pragmatic approach is simply to replace the least-squares criterion of Eq. (2.2) with

$$V(\theta,t) = \frac{1}{2}\sum_{i=1}^{t}\lambda^{t-i}\Big(y(i) - \varphi^T(i)\theta\Big)^2 \tag{2.20}$$

where λ is a parameter such that $0 < \lambda \leq 1$. The parameter λ is called the *forgetting factor* or *discounting factor*. The loss function of Eq. (2.20) implies that a time-varying weighting of the data is introduced. The most recent data is given unit weight, but data that is n time units old is weighted by λ^n. The method is therefore called *exponential forgetting* or *exponential discounting*. By repeating the calculations leading to Theorem 2.3 for the loss function of Eq. (2.20), the following result is obtained.

THEOREM 2.4 Recursive least squares with exponential forgetting

Assume that the matrix $\Phi(t)$ has full rank for $t \geq t_0$. The parameter θ, which minimizes Eq. (2.20), is given recursively by

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + K(t)\Big(y(t) - \varphi^T(t)\hat{\theta}(t-1)\Big) \\ K(t) &= P(t)\varphi(t) = P(t-1)\varphi(t)\left(\lambda I + \varphi^T(t)P(t-1)\varphi(t)\right)^{-1} \qquad (2.21) \\ P(t) &= \left(I - K(t)\varphi^T(t)\right)P(t-1)/\lambda \end{aligned}$$

□

A disadvantage of exponential forgetting is that data is discounted even if $P(t)\varphi(t) = 0$. This condition implies that $y(t)$ does not contain any new information about the parameter θ. In this case it follows from Eqs. (2.21) that the matrix P increases exponentially with rate λ. Several ways to avoid this are discussed in detail in Chapter 11.

An alternative method of dealing with time-varying parameters is to assume a time-varying mathematical model. Time-varying parameters can be obtained by replacing the first equation of Eqs. (2.18) with the model

$$\theta(t+1) = \Phi_v\theta(t) + v(t)$$

where Φ_v is a known matrix and $v(t)$ is discrete-time white noise. The filtering interpretation of the least-squares problem given in Remark 2 of Theorem 2.3 can now easily be generalized. The least-squares estimator will then be the Kalman filter. The case $\Phi_v = I$ corresponds to a model in which the parameters are drifting Wiener processes.

Simplified Algorithms

The recursive least-squares algorithm given by Theorem 2.3 has two sets of state variables, $\hat{\theta}$ and P, which must be updated at each step. For large n the updating of the matrix P dominates the computing effort. There are several simplified algorithms that avoid updating the P matrix at the cost of slower convergence. Kaczmarz's *projection algorithm* is one simple solution. To describe this algorithm, consider the unknown parameter as an element of R^n. One measurement

$$y(t) = \varphi^T(t)\theta \qquad (2.22)$$

determines the projection of the parameter vector θ on the vector $\varphi(t)$. From this it is immediately clear that n measurements, where $\varphi(1), \ldots, \varphi(n)$ span

R^n, are required to determine the parameter vector θ uniquely. Assume that an estimate $\hat{\theta}(t-1)$ is available and that a new measurement such as Eq. (2.22) is obtained. Since the measurement $y(t)$ contains information only in the direction $\varphi(t)$ in parameter space, it is natural to choose as the new estimate the value $\hat{\theta}(t)$ that minimizes $\|\hat{\theta}(t) - \hat{\theta}(t-1)\|$ subject to the constraint $y(t) = \varphi^T(t)\hat{\theta}(t)$. Introducing a Lagrangian multiplier $\bar{\alpha}$ to handle the constraint, we thus have to minimize the function

$$V = \frac{1}{2}\left(\hat{\theta}(t) - \hat{\theta}(t-1)\right)^T\left(\hat{\theta}(t) - \hat{\theta}(t-1)\right) + \bar{\alpha}\left(y(t) - \varphi^T(t)\hat{\theta}(t)\right)$$

Taking derivatives with respect to $\hat{\theta}(t)$ and $\bar{\alpha}$, we get

$$\begin{aligned} \hat{\theta}(t) - \hat{\theta}(t-1) - \bar{\alpha}\varphi(t) &= 0 \\ y(t) - \varphi^T(t)\hat{\theta}(t) &= 0 \end{aligned}$$

Solving these equations gives

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\varphi(t)}{\varphi^T(t)\varphi(t)}\left(y(t) - \varphi^T(t)\hat{\theta}(t-1)\right) \qquad (2.23)$$

The updating formula is called *Kaczmarz's algorithm*. It is useful to be able to change the step length of the parameter adjustment by introducing a factor γ. This gives

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\gamma\varphi(t)}{\varphi^T(t)\varphi(t)}\left(y(t) - \varphi^T(t)\hat{\theta}(t-1)\right)$$

To avoid a potential problem that occurs when $\varphi(t) = 0$, the denominator in the correction term is changed from $\varphi^T(t)\varphi(t)$ to $\varphi^T(t)\varphi(t) + \alpha$, where α is a positive constant. The following algorithm is then obtained.

ALGORITHM 2.1 Projection algorithm

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\gamma\varphi(t)}{\alpha + \varphi^T(t)\varphi(t)}\left(y(t) - \varphi^T(t)\hat{\theta}(t-1)\right) \qquad (2.24)$$

where $\alpha \geq 0$ and $0 < \gamma < 2$. □

Remark 1. In some textbooks this is called the normalized projection algorithm.

Remark 2. The bound for the parameter γ is obtained from the following analysis. Assume that data has been generated by Eq. (2.22) with parameter $\theta = \theta^0$. It then follows from Eq. (2.24) that the parameter error

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

satisfies the equation

$$\tilde{\theta}(t) = A(t)\tilde{\theta}(t-1)$$

where

$$A(t) = I - \frac{\gamma\varphi(t)\varphi^T(t)}{\alpha + \varphi^T(t)\varphi(t)}$$

The matrix $A(t)$ has one eigenvalue,

$$\lambda = \frac{\alpha + (1-\gamma)\varphi^T\varphi}{\alpha + \varphi^T\varphi}$$

This value is less than 1 in magnitude if $0 < \gamma < 2$. The other eigenvalues of A are all equal to 1. □

The projection algorithm assumes that the data is generated by Eq. (2.22) with no error. When the data is generated by Eq. (2.12) with additional random error, a simplified algorithm is given by

$$\hat\theta(t) = \hat\theta(t-1) + P(t)\varphi(t)\left(y(t) - \varphi^T(t)\hat\theta(t-1)\right) \tag{2.25}$$

where

$$P(t) = \left(\sum_{i=1}^{t} \varphi^T(i)\varphi(i)\right)^{-1} \tag{2.26}$$

This is the *stochastic approximation* (SA) *algorithm*. Notice that $P(t) = \Phi\Phi^T$ is now a scalar when $y(t)$ is a scalar. A further simplification is the *least mean square (LMS) algorithm* in which the parameter updating is done by using

$$\hat\theta(t) = \hat\theta(t-1) + \gamma\varphi(t)\left(y(t) - \varphi^T(t)\hat\theta(t-1)\right)$$

where γ is a constant.

Continuous-Time Models

In the recursive schemes the variables have so far been indexed by a discrete parameter t. The notation t was chosen because in many applications it denotes time. In some cases it is natural to use continuous-time observations. It is straightforward to generalize the results to this case. Equation (2.1) is still used, but i is now assumed to be a real variable. Assuming exponential forgetting, the parameter should be determined such that the criterion

$$V(\theta) = \int_0^t e^{-\alpha(t-\tau)}\left(y(\tau) - \varphi^T(\tau)\theta\right)^2 d\tau \tag{2.27}$$

is minimized. The parameter α, where $\alpha \geq 0$, corresponds to the forgetting factor λ in Eq. (2.20). A straightforward calculation shows that the criterion is minimized if (see Problem 2.15 at the end of the chapter)

$$\left(\int_0^t e^{-\alpha(t-\tau)}\varphi(\tau)\varphi^T(\tau)\,d\tau\right)\hat\theta(t) = \int_0^t e^{-\alpha(t-\tau)}\varphi(\tau)y(\tau)\,d\tau \tag{2.28}$$

which is the *normal equation*. The estimate is unique if the matrix

$$R(t) = \int_0^t e^{-\alpha(t-\tau)}\varphi(\tau)\varphi^T(\tau)\,d\tau \tag{2.29}$$

is invertible. It is also possible to obtain recursive equations by differentiating Eq. (2.28). The estimate is given by the following theorem.

THEOREM 2.5 Continuous-time least-squares estimation

Assume that the matrix $R(t)$ given by Eq. (2.29) is invertible for all t. The estimate that minimizes Eq. (2.27) satisfies

$$\frac{d\hat\theta}{dt} = P(t)\varphi(t)e(t) \tag{2.30}$$

$$e(t) = y(t) - \varphi^T(t)\hat\theta(t) \tag{2.31}$$

$$\frac{dP(t)}{dt} = \alpha P(t) - P(t)\varphi(t)\varphi^T(t)P(t) \tag{2.32}$$

Proof: The theorem is proved by differentiating Eq. (2.28). □

Remark 1. The matrix $R(t) = P(t)^{-1}$ satisfies

$$\frac{dR}{dt} = -\alpha R + \varphi\varphi^T$$

Remark 2. There are also continuous-time versions of the simplified algorithms. The projection algorithm corresponding to Eqs. (2.25) and (2.26) is given by Eq. (2.30) with

$$P(t) = \left(\int_0^t \varphi^T(\tau)\varphi(\tau)\,d\tau\right)^{-1}$$

where $P(t)$ is now a scalar. □

2.3 ESTIMATING PARAMETERS IN DYNAMICAL SYSTEMS

We now show how the least-squares method can be used to estimate parameters in models of dynamical systems. The particular way to do this will depend on the character of the model and its parameterization.

Finite-Impulse Response (FIR) Models

A linear time-invariant dynamical system is uniquely characterized by its impulse response. The impulse response is in general infinite-dimensional. For stable systems the impulse response will go to zero exponentially fast and may then be truncated. Notice, however, that a large number of parameters may be required if the sampling interval is short in comparison to the slowest time constant of the system. This results in the so-called finite impulse response (FIR) model, which is also called a transversal filter. The model can be described by the equation

$$y(t) = b_1u(t-1) + b_2u(t-2) + \cdots + b_nu(t-n) \tag{2.33}$$

or

$$y(t) = \varphi^T(t-1)\theta$$

where

$$\theta^T = \begin{pmatrix} b_1 & \ldots & b_n \end{pmatrix}$$
$$\varphi^T(t-1) = \begin{pmatrix} u(t-1) & \ldots & u(t-n) \end{pmatrix}$$

This model is identical to the regression model of Eq. (2.1), except for the index t of the regression vector, which is different. The reason for this change of notation is that it will be convenient to label the regression vector with the time of the most recent data that appears in the regressor. The model of Eq. (2.33) clearly fits the least-squares formulation, and the estimator is then given by Theorem 2.3.

The parameter estimator can be represented by the block diagram in Fig. 2.3. The estimator may be regarded as a system with inputs u and y and output θ. Since the signal

$$\hat{y}(t) = \hat{b}_1(t-1)u(t-1) + \cdots + \hat{b}_n(t-1)u(t-n)$$

is available in the system, we can also consider $\hat{y}(t)$ as an output. Since $\hat{y}(t)$ is a predicted estimate of y, the recursive estimator can also be interpreted as an *adaptive filter* to predict y. The use of this filter is discussed in Chapter 13.

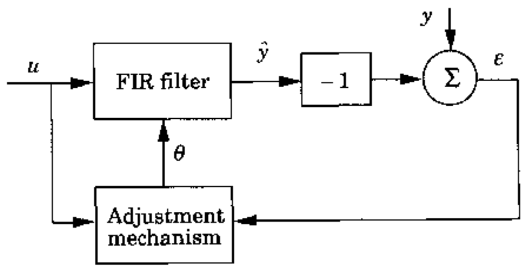

Figure 2.3 Block diagram representation of a recursive parameter estimator for an FIR model.

Transfer Function Models

The least-squares method can be used to identify parameters in dynamical systems. Let the system be described by the model

$$A(q)y(t) = B(q)u(t) \tag{2.34}$$

where q is the forward shift operator and $A(q)$ and $B(q)$ are the polynomials

$$A(q) = q^n + a_1q^{n-1} + \ldots + a_n$$
$$B(q) = b_1q^{m-1} + b_2q^{m-2} + \ldots + b_m$$

Equation (2.34) can be written as the difference equation

$$y(t) + a_1y(t-1) + \cdots + a_ny(t-n) = b_1u(t+m-n-1) + \cdots + b_mu(t-n)$$

Assume that the sequence of inputs $\{u(1), u(2), \ldots, u(t)\}$ has been applied to the system and the corresponding sequence of outputs $\{y(1), y(2), \ldots, y(t)\}$ has been observed. Introduce the parameter vector

$$\theta^T = \begin{pmatrix} a_1 & \ldots & a_n & b_1 & \ldots & b_m \end{pmatrix} \tag{2.35}$$

and the regression vector

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

Notice that the output signal appears delayed in the regression vector. The model is therefore called an *autoregressive model*. The way in which the elements are ordered in the matrix θ is, of course, arbitrary, provided that $\varphi(t-1)$ is also similarly reordered. Later, in dealing with adaptive control, it will be natural to reorder the terms. The convention that the time index of the φ vector will refer to the time when all elements in the vector are available will also be adopted. The model can formally be written as the regression model

$$y(t) = \varphi^T(t-1)\theta$$

Parameter estimates can be obtained by applying the least-squares method (Theorem 2.1). The matrix Φ is given by

$$\Phi = \begin{pmatrix} \varphi^T(n) \\ \vdots \\ \varphi^T(t-1) \end{pmatrix}$$

If we use the statistical interpretation of the least-squares estimate given by Theorem 2.2, it follows that the method described will work well when the disturbances can be described as white noise added to the right-hand side of Eq. (2.34). This leads to the least-squares model

$$A(q)y(t) = B(q)u(t) + e(t+n)$$

(Compare with Eq. (2.12).) The method is therefore called an *equation error* method. A slight variation of the method is better if the disturbances are described instead as white noise added to the system output, that is, when the model is

$$y(t) = \frac{B(q)}{A(q)}u(t) + e(t)$$

The method obtained is then called an *output error* method. To describe such a method, let u be the input and $\hat{y}$ be the output of a system with the input-output relation

$$\hat{y}(t) + a_1\hat{y}(t-1) + \cdots + a_n\hat{y}(t-n) = b_1u(t+m-n-1) + \cdots + b_mu(t-n)$$

that is,

$$\hat{y}(t) = \frac{B(q)}{A(q)}u(t)$$

Determine the parameters that minimize the criterion

$$\sum_{k=1}^{t}(y(k) - \hat{y}(k))^2$$

where $y(t) = \hat{y}(t) + e(t)$. This problem can be interpreted as a least-squares problem whose solution is given by

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi(t-1)\varepsilon(t)$$

where

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

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

Compare with Theorem 2.1. The recursive estimator obtained can be represented by the block diagram in Fig. 2.4.

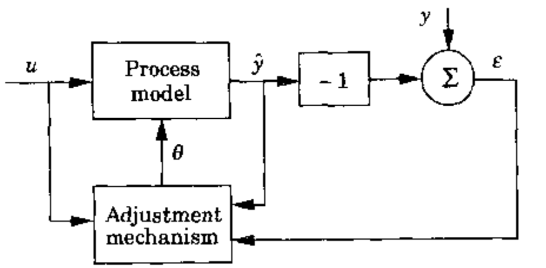

Figure 2.4 Block diagram of a least-squares estimator based on the output error.

Continuous-Time Transfer Functions

We now show that the least-squares method can also be used to estimate parameters in continuous-time transfer functions. For instance, consider a continuous-time model of the form

$$\frac{d^ny}{dt^n} + a_1\frac{d^{n-1}y}{dt^{n-1}} + \cdots + a_ny = b_1\frac{d^{m-1}u}{dt^{m-1}} + \cdots + b_mu$$

which can also be written as

$$A(p)y(t) = B(p)u(t) \tag{2.36}$$

where $A(p)$ and $B(p)$ are polynomials in the differential operator $p = d/dt$. In most cases we cannot conveniently compute $p^ny(t)$ because it would involve taking n derivatives of a signal. The model of Eq. (2.36) is therefore rewritten as

$$A(p)y_f(t) = B(p)u_f(t) \tag{2.37}$$

where

$$y_f(t) = H_f(p)y(t)$$
$$u_f(t) = H_f(p)u(t)$$

and $H_f(p)$ is a stable transfer function with a pole excess of n or more. See Fig. 2.5. If we introduce

$$\theta = \begin{pmatrix} a_1 & \ldots & a_n & b_1 & \ldots & b_m \end{pmatrix}^T$$

$$\begin{aligned}\varphi^T(t) &= \begin{pmatrix} -p^{n-1}y_f & \ldots & -y_f & p^{m-1}u_f & \ldots & u_f \end{pmatrix}\\ &= \begin{pmatrix} -p^{n-1}H_f(p)y & \ldots & -H_f(p)y & p^{m-1}H_f(p)u & \ldots & H_f(p)u \end{pmatrix}\end{aligned}$$

the model expressed by Eq. (2.37) can be written as

$$p^ny_f(t) = p^nH_f(p)y(t) = \varphi^T(t)\theta$$

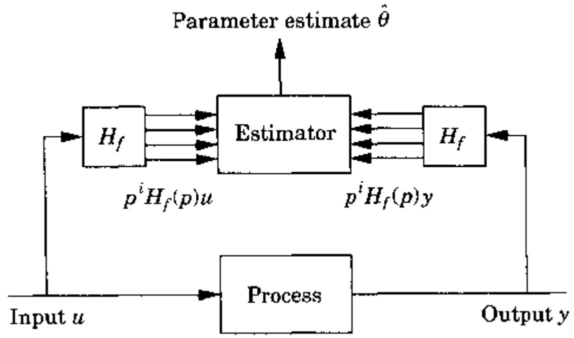

Figure 2.5 Block diagram of estimator with filters H_f.

By a proper realization of the filter H_f it is possible to use one filter to generate all the signals $p^i H_f(p)y$, $i = 0, \ldots, n$, and another filter to generate $p^i H_f(p)u$, $i = 0, \ldots, m-1$. Standard least squares can now be applied, since this is a regression model. A recursive estimate is given by Theorem 2.5. With the restriction on H_f there will not be any pure differentiation of the output or the input to the system.

Nonlinear Models

Least squares can also be applied to certain nonlinear models. The essential restriction is that the models be linear in the parameters so that they can be written as linear regression models. Notice that the regressors do not need to be linear in the inputs and outputs. An example illustrates the idea.

EXAMPLE 2.3 Nonlinear system

Consider the model

$$y(t) + ay(t-1) = b_1 u(t-1) + b_2 \sin(u(t-1))$$

By introducing

$$\theta = \begin{pmatrix} a & b_1 & b_2 \end{pmatrix}^T$$

and

$$\varphi^T(t) = \begin{pmatrix} -y(t) & u(t) & \sin(u(t-1)) \end{pmatrix}$$

the model can be written as

$$y(t) = \varphi^T(t-1)\theta$$

The model is linear in the parameters, and the least-squares method can be used to estimate θ. □

Stochastic Models

The least-squares estimate is biased when it is used on data generated by Eq. (2.12), where the errors $e(i)$ are correlated. The reason is that $E\varphi^T(i)e(i) \neq 0$ (compare Eq. (2.13)). A possibility to cope with this problem is to model the correlation of the disturbances and to estimate the parameters describing the correlations. Consider the model

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \tag{2.38}$$

where $A(q), B(q)$, and $C(q)$ are polynomials in the forward shift operator and $\{e(t)\}$ is white noise. The parameters of the polynomial C describe the correlation of the disturbance. The model of Eq. (2.38) cannot be converted

directly to a regression model, since the variables $\{e(t)\}$ are not known. A regression model can, however, be obtained by suitable approximations. To describe these, introduce

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

where

$$\theta = \begin{pmatrix} a_1 & \ldots & a_n & b_1 & \ldots & b_n & c_1 & \ldots & c_n \end{pmatrix}$$

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

The variables $e(t)$ are approximated by the prediction errors $\varepsilon(t)$. The model can then be approximated by

$$y(t) = \varphi^T(t-1)\theta + e(t)$$

and standard recursive least squares can be applied. The method obtained is called *extended least squares* (ELS). The equations for updating the estimates are given by

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + P(t)\varphi(t-1)\varepsilon(t) \\ P^{-1}(t) &= P^{-1}(t-1) + \varphi(t-1)\varphi^T(t-1) \end{aligned} \tag{2.39}$$

(Compare with Theorem 2.3.) Another method of estimating the parameters in Eq. (2.38) is to use Eqs. (2.39) and let the residual be defined by

$$\hat{C}(q)\varepsilon(t) = \hat{A}(q)y(t) - \hat{B}(q)u(t) \tag{2.40}$$

and regression vector φ in Eqs. (2.39) be replaced by φ_f, where

$$\hat{C}(q)\varphi_f(t) = \varphi(t) \tag{2.41}$$

The most recent estimates should be used in these updates. The method obtained is then not truly recursive, since Eqs. (2.41) and (2.40) have to be solved from $t = 1$ for each measurement. The following approximations can be made:

$$\varepsilon(t) = y(t) - \varphi_f^T(t-1)\hat{\theta}(t-1)$$

This algorithm is called the *recursive maximum likelihood (RML) method*.

It is advantageous for both ELS and RML to replace the residual in the regression vector by the *posterior residual* defined as

$$\varepsilon_p(t) = y(t) - \varphi^T(t-1)\hat{\theta}(t)$$

that is, the latest value of $\hat{\theta}$ is used to compute ε_p.

Another possibility to model the correlated noise is to use the model

$$y(t) = \frac{B(q)}{A(q)}u(t) + \frac{C(q)}{D(q)}e(t)$$

instead of Eq. (2.38). Recursive parameter estimates for this model can be derived in the same way as for Eq. (2.38).

Details about the extended least-squares method and the recursive maximum likelihood method are found in the references at the end of the chapter.

Unification

The different recursive algorithms discussed are quite similar. They can all be described by the equations

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi(t-1)\varepsilon(t)$$

$$P(t) = \frac{1}{\lambda}\left(P(t-1) - \frac{P(t-1)\varphi(t-1)\varphi^T(t-1)P(t-1)}{\lambda + \varphi^T(t-1)P(t-1)\varphi(t-1)}\right)$$

where θ, φ, and ε are different for the different methods.

2.4 EXPERIMENTAL CONDITIONS

The properties of the data used in parameter estimation are crucial for the quality of the estimates. For example, it is obvious that no useful parameter estimates can be obtained if all signals are identically zero. In this section we discuss the influence of the experimental conditions on the quality of the estimates. In performing system identification automatically, as in an adaptive system, it is essential to understand these conditions, as well as the mechanisms that can interfere with proper identification. The notion of persistent excitation, which is one way to characterize process inputs, is introduced. In adaptive systems the plant input is generated by feedback. Difficulties caused by this are also discussed.

Persistent Excitation

Let us first consider estimation of parameters in a FIR model given by Eq. (2.33). The parameters of the model cannot be determined unless some conditions are imposed on the input signal. It follows from the condition for uniqueness of the least-squares estimate given by Theorem 2.1 that the minimum is unique if the matrix

$$\Phi^T\Phi = \begin{pmatrix} \sum_{n+1}^{t} u^2(k-1) & \sum_{n+1}^{t} u(k-1)u(k-2) & \dots & \sum_{n+1}^{t} u(k-1)u(k-n) \\ \sum_{n+1}^{t} u(k-1)u(k-2) & \sum_{n+1}^{t} u^2(k-2) & \dots & \sum_{n+1}^{t} u(k-2)u(k-n) \\ \vdots & & & \\ \sum_{n+1}^{t} u(k-1)u(k-n) & & & \sum_{n+1}^{t} u^2(k-n) \end{pmatrix} \tag{2.42}$$

has full rank. This condition is called an *excitation condition*. For long data sets, all sums in Eq. (2.42) can be taken from 1 to t. We then get

$$C_n = \lim_{t\to\infty} \frac{1}{t}\Phi^T\Phi = \begin{pmatrix} c(0) & c(1) & \dots & c(n-1) \\ c(1) & c(0) & \dots & c(n-2) \\ \vdots & & & \\ c(n-1) & c(n-2) & \dots & c(0) \end{pmatrix} \tag{2.43}$$

where $c(k)$ are the empirical covariances of the input, that is,

$$c(k) = \lim_{t\to\infty} \frac{1}{t}\sum_{i=1}^{t} u(i)u(i-k) \tag{2.44}$$

For long data sets the condition for uniqueness can thus be expressed as the matrix in Eq. (2.43) being positive definite. This leads to the following definition.

DEFINITION 2.1 Persistent excitation

A signal u is called *persistently exciting* (PE) of order n if the limits (2.44) exist and if the matrix C_n given by Eq. (2.43) is positive definite.

Remark 1. In the adaptive control literature an alternative definition of PE is often used. The signal u is said to be persistently exciting of order n if for all t there exists an integer m such that

$$\rho_1 I > \sum_{k=t}^{t+m} \varphi(k)\varphi^T(k) > \rho_2 I$$

where $\rho_1, \rho_2 > 0$ and the vector $\varphi(t)$ is given by

$$\varphi(t) = \begin{pmatrix} u(t-1) & u(t-2) & \dots & u(t-n) \end{pmatrix}$$

Notice that the matrix (2.43) can be written as

$$C_n = \lim_{t\to\infty} \frac{1}{t}\sum_{k=1}^{t} \varphi(k)\varphi^T(k)$$

Remark 2. Notice that no mean value is included in the definition of the empirical covariance $c(k)$ in Eq. (2.44). □

The following result can be established.

THEOREM 2.6 Consistency for FIR models

Consider least-squares estimation of the parameters of a finite impulse response model with n parameters. The estimate is consistent and the variance of the estimates goes to zero as $1/t$ if the input signal is persistently exciting of order n.

Proof: The result follows from Definition 2.1 and Theorem 2.2. □

We now introduce the following theorem.

THEOREM 2.7 Persistently exciting signals

The signal u with the property (2.44) is persistently exciting of order n if and only if

$$U = \lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t} (A(q)u(k))^2 > 0 \tag{2.45}$$

for all nonzero polynomials A of degree $n-1$ or less.

Proof: Let the polynomial A be

$$A(q) = a_0 q^{n-1} + a_1 q^{n-2} + \cdots + a_{n-1}$$

A straightforward calculation shows that

$$U = \lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t} \left(a_0 u(k+n-1) + \cdots + a_{n-1}u(k)\right)^2 = a^T C_n a$$

where C_n is the matrix given by Eq. (2.43). If C_n is positive definite, the right-hand side is positive for all a, and so is the left-hand side. Conversely, if the left-hand side is positive for all a, so is the right-hand side. □

The result is useful in investigating whether special signals are persistently exciting.

EXAMPLE 2.4 Pulse

It follows from Eq. (2.45) that $C_n \to 0$ for all n if u is a pulse. A pulse thus is not PE for any n. □

EXAMPLE 2.5 Step

Let $u(t) = 1$ for $t > 0$ and zero otherwise. It follows that

$$(q-1)u(t) = \begin{cases} 1 & t = 0 \\ 0 & t \neq 0 \end{cases}$$

A step can thus at most be PE of order 1. Since

$$C_1 = \frac{1}{t}\sum_{k=1}^{t} u^2(k) = 1$$

it follows that it is PE of order 1. □

EXAMPLE 2.6 Sinusoid

Let $u(t) = \sin \omega t$. It follows that

$$\left(q^2 - 2q\cos\omega + 1\right) u(t) = 0$$

A sinusoid can thus at most be PE of order 2. Since

$$C_2 = \frac{1}{2}\begin{pmatrix} 1 & \cos\omega \\ \cos\omega & 1 \end{pmatrix}$$

it follows that a sinusoid is actually PE of order 2. □

EXAMPLE 2.7 Periodic signal

Let $u(t)$ be periodic with period n. It then follows that

$$(q^n - 1)\, u(t) = 0$$

The signal can thus at most be PE of order n. □

EXAMPLE 2.8 Random signals

Consider the stochastic process

$$u(t) = H(q)e(t)$$

where $e(t)$ is white noise and $H(q)$ is a pulse transfer function. It follows from the definition of white noise that Eq. (2.45) is satisfied for the signal e for any nonzero polynomial $A(q)$. This property also holds for the signal u. The signal u is thus PE of any order. □

To be able to give a frequency domain interpretation of PE, it is useful to use the following theorem, which is given without proof.

THEOREM 2.8 Parseval's theorem

Let

$$H(q^{-1}) = \sum_{k=0}^{\infty} h_k q^{-k}$$

$$G(q^{-1}) = \sum_{k=0}^{\infty} g_k q^{-k}$$

be two stable transfer functions, and let $e(t)$ be white noise of zero mean and covariance σ^2. Then

$$\sigma^2 \sum_{k=0}^{\infty} h_k g_k = \frac{\sigma^2}{2\pi}\int_{-\pi}^{\pi} H(e^{i\omega})G(e^{i\omega})\, d\omega$$

□

Remark. The left-hand side can be interpreted as $E\left(H(q^{-1})e(t) \cdot G(q^{-1})e(t)\right)$, that is, the covariance of the two signals obtained by sending white noise through the transfer functions $H(q^{-1})$ and $G(q^{-1})$. □

EXAMPLE 2.9 Frequency domain characterization

Consider a quasi-stationary signal $u(t)$ with spectrum $\Phi_u(\omega)$. It follows from Parseval's theorem that

$$\lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t} (A(q)u(k))^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |A(e^{i\omega})|^2 \Phi_u(\omega)\, d\omega \tag{2.46}$$

This equation gives considerable insight into the notion of persistent excitation. A polynomial of degree $n-1$ can at most vanish in $n-1$ points. The right-hand side of Eq. (2.46) will thus be positive if $\Phi_u(\omega) \neq 0$ for at least n points in the interval $-\pi \le \omega \le \pi$. A signal whose spectrum is different from zero in an interval is thus persistently exciting of any order. □

A sinusoid has a point spectrum that differs from zero at two points. It is thus persistently exciting of second order. A signal that is a sum of k sinusoids is persistently exciting of order $2k$. The frequency domain characterization also makes it possible to derive the following result.

THEOREM 2.9 PE of filtered signals

Let the signal u be persistently exciting of order n. Assume that $A(q)$ is a polynomial of degree $m < n$. The signal v defined by

$$v(t) = A(q)u(t)$$

is then persistently exciting of order ℓ with $n - m \le \ell \le n$. Assuming that A is stable, the signal w defined by

$$w(t) = \frac{1}{A(q)} u(t)$$

is persistently exciting of order n. □

Transfer Function Models

The properties of parameter estimates for discrete-time transfer functions will now be discussed. The uniqueness of the estimates will first be explored. For this purpose it is assumed that the data is actually generated by

$$A^0(q)y(t) = B^0(q)u(t) + e(t+n) \tag{2.47}$$

where A^0 and B^0 are relatively prime. Let A and B be the estimates of A^0 and B^0, respectively. If $e = 0$, $\deg A > \deg A^0$, and $\deg B > \deg B^0$, it follows from Theorem 2.1 that the estimate is not unique because the columns of the matrix Φ are linearly dependent. However, we have the following result.

THEOREM 2.10 Transfer function estimation

Consider data generated by the model of Eq. (2.47), with A^0 stable and $e = 0$. Let the parameters of the polynomials A and B be fitted by least squares. Assume that the input u is persistently exciting of order $\deg A + \deg B + 1$. If it is further assumed that $\deg A = \deg A^0$ and $\deg B \ge B^0$, then $\lim_{t\to\infty} \Phi^T\Phi/t$ is positive definite.

Proof: Consider

$$V(\theta) = \theta^T \lim_{t\to\infty} \frac{1}{t} \Phi^T \Phi \theta = \lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t} \left(\varphi^T(k)\theta\right)^2$$

Introduce

$$\begin{aligned} v(t) &= \varphi^T(t+n-1)\theta = B(q)u(t) - (A(q) - q^n)\,y(t) \\ &= B(q)u(t) - \frac{A(q) - q^n}{A^0(q)} B^0(q)u(t) \\ &= \left\{A^0(q)B(q) - \left(A(q) - q^n\right)B^0(q)\right\} \frac{1}{A^0(q)} u(t) \end{aligned}$$

Since A^0 is stable, it follows from Theorem 2.9 that the signal $1/A^0(q) \cdot u(t)$ is persistently exciting of order $\deg A + \deg B + 1$. Since the polynomial in braces has a degree lower than or equal to $\deg A + \deg B$, it follows that the signal $v(t)$ does not vanish in the mean square sense unless the polynomial is identically zero. This happens if

$$\frac{B^0(q)}{A^0(q)} = \frac{B(q)}{A(q) - q^n}$$

Since $\deg A = \deg A^0$, the denominator on the right-hand side thus has degree $\deg A - 1 = \deg A^0 - 1$. The rational functions are then not identical, and the theorem is proved. □

Remark 1. Notice that $\deg A + \deg B + 1$ is equal to the number of parameters in the model of Eq. (2.47). The order of PE required is thus equal to the number of estimated parameters.

Remark 2. If the data is generated by Eq. (2.47), where $\{e(t)\}$ is white noise (i.e., a sequence of uncorrelated random variables), then the matrix

$$\lim_{t\to\infty} \frac{1}{t} \Phi^T \Phi$$

is positive definite for models of all orders provided that the input is persistently exciting of order $\deg B + 1$. □

Theorem 2.2 does not automatically apply to estimation of parameters of a transfer function, because the output y appears in the regression vector. A consequence of this is that theoretical properties of the estimates can be established asymptotically only for large number of observations.

Identification in Closed Loop

In adaptive control, system identification is often performed under closed-loop conditions, which may give rise to certain difficulties. Consider, for example, the estimation of the coefficients of a transfer function model as in Eq. (2.34). The matrix Φ is then

$$\Phi = \begin{pmatrix} -y(n) & \dots & -y(1) & u(n) & \dots & u(1) \\ -y(n+1) & \dots & -y(2) & u(n+1) & \dots & u(2) \\ \vdots & & & & & \vdots \\ -y(t-1) & \dots & -y(t-n) & u(t-1) & \dots & u(t-n) \end{pmatrix} \tag{2.48}$$

A linear feedback of sufficiently low order introduces linear dependencies among the columns of the matrix Φ. This means that the parameters cannot be determined uniquely. A simple example shows what may happen.

EXAMPLE 2.10 Loss of identifiability due to feedback

Consider a system described by

$$y(t+1) + ay(t) = bu(t) \tag{2.49}$$

Assume that the parameters a and b should be estimated in the presence of the feedback

$$u(t) = -ky(t) \tag{2.50}$$

Multiplying Eq. (2.50) by α and adding to Eq. (2.49) give

$$y(t+1) + (a + \alpha k)y(t) = (b - \alpha)u(t)$$

This shows that any parameters such that

$$\hat{a} = a + \alpha k$$
$$\hat{b} = b - \alpha$$

give the same input-output relation. The above equation represents a straight line

$$\hat{b} = b + \frac{1}{k}(a - \hat{a}) \tag{2.51}$$

in parameter space (see Fig. 2.6). The least-squares loss function (2.2) has the same value for all parameters on this line. □

The problem with lack of identifiability due to feedback disappears if a linear feedback of sufficiently high order is used. Then the columns of the matrix Φ given by Eq. (2.48) are no longer linearly dependent. Another possibility is to have a time-varying feedback. For example, in Example 2.10 it is sufficient to have a feedback of the form

$$u(t) = -k_1 y(t) - k_2 y(t-1)$$

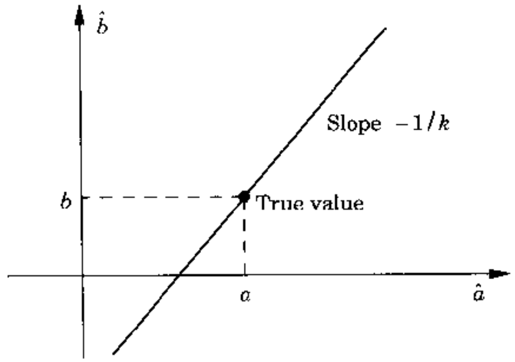

Figure 2.6 Illustration of lack of uniqueness in closed-loop identification.

with $k_2 \neq 0$. Another possibility is to use a feedback law

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

where k varies with time. For instance, in Example 2.10 it is sufficient to use two values of the gain. Each value of the gain corresponds to a straight line with slope $-1/k$ in parameter space. Two lines give a unique intersection.

In adaptive systems there is a natural time variation in the feedback because the feedback gains are based on parameter estimates. In a typical case the variance of the parameters decreases as $1/t$, but more complex behavior is also possible. The following example shows what can happen.

EXAMPLE 2.11 Convergence rate

Consider data generated by

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

with a feedback of the form

$$u(t) = -k\left(1 + \frac{v(t)}{\sqrt{t}}\right)y(t) \tag{2.52}$$

where $\{v(t)\}$ is a sequence of independent random variables that are also independent of $\{e(t)\}$. With the feedback law of Eq. (2.52) the closed-loop system becomes

$$y(t+1) = -\left(a + bk + \frac{bkv(t)}{\sqrt{t}}\right)y(t) + e(t+1)$$

Given measurements up to $t+1$, the matrix $\Phi^T\Phi$ of the estimation problem is

$$\Phi^T\Phi = \begin{pmatrix} \sum_{j=1}^{t} y^2(j) & \sum_{j=1}^{t} y(j)u(j) \\ \sum_{j=1}^{t} y(j)u(j) & \sum_{j=1}^{t} u^2(j) \end{pmatrix}$$

It follows that

$$\begin{aligned} \sum_{j=1}^{t} y(j)u(j) &= -k\sum_{j=1}^{t} y^2(j) - k\sum_{j=1}^{t} \frac{v(j)y^2(j)}{\sqrt{j}} \approx -k\sum_{j=1}^{t} y^2(j) \approx -kt\sigma_y^2 \\ \sum_{j=1}^{t} u^2(j) &= k^2\left(\sum_{j=1}^{t} y^2(j) + 2\sum_{j=1}^{t}\frac{v(j)y^2(j)}{\sqrt{j}} + \sum_{j=1}^{t}\frac{v^2(j)y^2(j)}{j}\right) \\ &\approx k^2\left(\sum_{j=1}^{t} y^2(j) + \sum_{j=1}^{t}\frac{v^2(j)y^2(j)}{j}\right) \approx k^2\sigma_y^2\left(t + \sigma_v^2\log t\right) \end{aligned}$$

Hence for large t,

$$\Phi^T\Phi \approx \sigma_y^2 \begin{pmatrix} t & -kt \\ -kt & k^2\left(t+\sigma_v^2\log t\right) \end{pmatrix}$$

The covariance matrix of the estimate is thus

$$\sigma_e^2\left(\Phi^T\Phi\right)^{-1} \approx \frac{\sigma_e^2}{\sigma_y^2\sigma_v^2}\begin{pmatrix} \dfrac{1}{\log t} + \dfrac{\sigma_v^2}{t} & \dfrac{1}{k\log t} \\ \dfrac{1}{k\log t} & \dfrac{1}{k^2\log t} \end{pmatrix}$$

It now follows that

$$\begin{aligned} \operatorname{cov}\left(\hat a - k\hat b\right) &= \sigma_e^2\begin{pmatrix} 1 & -k \end{pmatrix}\left(\Phi^T\Phi\right)^{-1}\begin{pmatrix} 1 & -k \end{pmatrix}^T \approx \frac{\sigma_e^2}{t\sigma_y^2} \\ \operatorname{cov}\left(k\hat a + \hat b\right) &= \sigma_e^2\begin{pmatrix} k & 1 \end{pmatrix}\left(\Phi^T\Phi\right)^{-1}\begin{pmatrix} k & 1 \end{pmatrix}^T \approx \frac{\sigma_e^2}{\sigma_y^2\sigma_v^2\log t} \end{aligned}$$

The estimate will thus approach the line (2.51) at the rate $1/t$. The estimate will then converge toward the correct values at the rate $1/\log t$. The convergence along the line (2.51) is slower than convergence toward the line. □

2.5 SIMULATION OF RECURSIVE ESTIMATION

In this section, different properties of the recursive least-squares (RLS) method are illustrated through simulations. Throughout the section, data is generated by

$$y(t) + ay(t-1) = bu(t-1) + e(t) + ce(t-1) \tag{2.53}$$

where $a = -0.8, b = 0.5$, and $e(t)$ is zero mean white noise with standard deviation $\sigma = 0.5$. Furthermore, $c = 0$, $P(0) = 100 \cdot I$, and $\hat\theta(0) = 0$ except when indicated. In most cases we use

$$\hat\theta = \begin{pmatrix} \hat a \\ \hat b \end{pmatrix} \qquad \varphi(t-1) = \begin{pmatrix} -y(t-1) & u(t-1) \end{pmatrix}$$

Only in Example 2.13 is the parameter c estimated.

EXAMPLE 2.12 Excitation

The need for persistency of excitation is illustrated in this example. A simulation of the estimates when the input is a unit pulse at $t = 50$ is shown in Fig. 2.7(a). The estimate $\hat a$ appears to converge to the correct value, but the estimate $\hat b$ does not. The reason for this is that information about the a parameter is obtained through the excitation by the noise. Information about the b parameter is obtained only through the pulse that is not persistently exciting.

In Fig. 2.7(b) the experiment is repeated, but the input is now a square wave of unit amplitude and a period of 100 samples. Both $\hat a$ and $\hat b$ will converge to their true values, because the input is persistently exciting. The absolute values of the elements of $P(t)$ are decreasing with time. For the simulation in

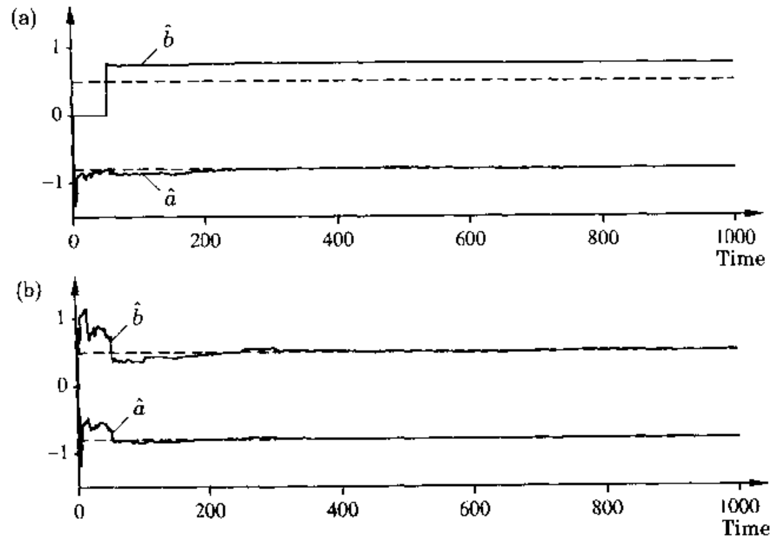

Figure 2.7 The estimated (solid line) and true (dashed line) parameter values in estimating the parameters in the model (2.53). The input signal $u(t)$ is (a) a unit pulse at $t = 50$, (b) a unit amplitude square wave with period 100.

Fig. 2.7(b) we have

$$\begin{pmatrix} \hat{a}(1000) \\ \hat{b}(1000) \end{pmatrix} = \begin{pmatrix} -0.796 \\ 0.511 \end{pmatrix} \qquad P(1000) = \begin{pmatrix} 0.550 & 1.114 \\ 1.114 & 3.258 \end{pmatrix} \cdot 10^{-3}$$

According to Theorem 2.2 this implies the following standard deviations for the estimates:

$$\sigma_{\hat{a}} = 0.5\sqrt{5.50} \cdot 10^{-2} = 0.012$$

$$\sigma_{\hat{b}} = 0.5\sqrt{32.58} \cdot 10^{-2} = 0.029$$

The estimates are thus well within one standard deviation of their true values. □

EXAMPLE 2.13 Model structure

In this example, parameter c in Eq. (2.53) has the value −0.5. Figure 2.8(a) shows the estimates of parameters a and b. The estimates do not converge to their true values. This is because the equation error $e(t) + ce(t-1)$ is not white noise. The assumptions in Theorem 2.2 are thus violated. Figure 2.8(b) shows the estimates when the extended least squares (ELS) method is used. All three parameters a, b, and c are then estimated, and the estimates converge to the true values. When only a and b are estimated by using the least-squares

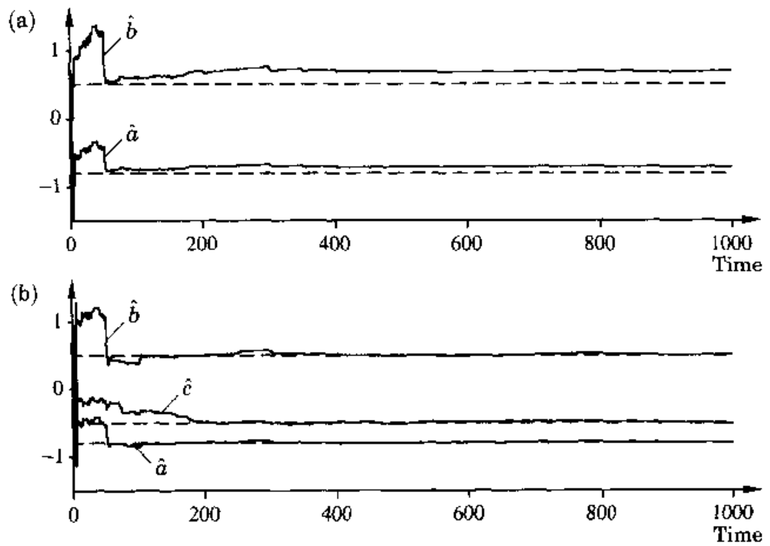

Figure 2.8 Estimated parameters when the model (2.53) is simulated with $c = -0.5$ by using (a) LS and (b) ELS.

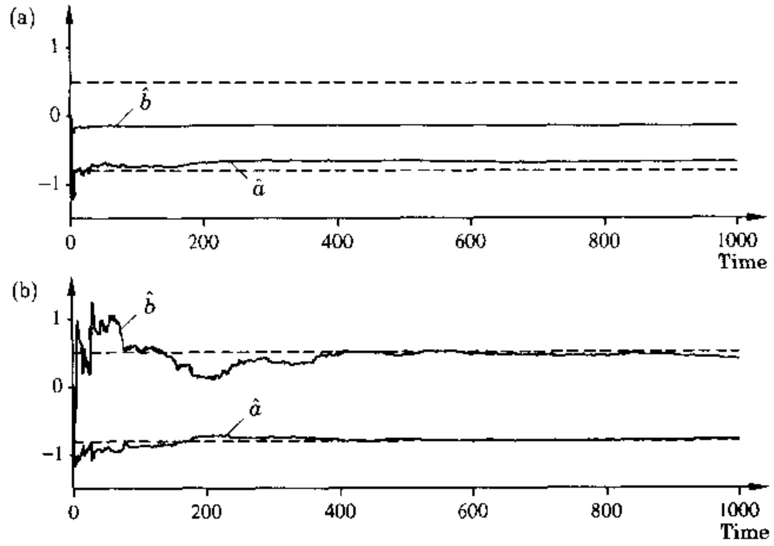

Figure 2.9 Estimates when the control signal is generated through feedback (a) $u(t) = -0.2y(t)$ and (b) $u(t) = -0.32y(t-1)$.

method, the estimates and the P-matrix at time $t = 1000$ are

$$\begin{pmatrix} \hat{a}(1000) \\ \hat{b}(1000) \end{pmatrix} = \begin{pmatrix} -0.702 \\ 0.697 \end{pmatrix} \qquad P(1000) = \begin{pmatrix} 0.710 & 1.435 \\ 1.435 & 3.903 \end{pmatrix} \cdot 10^{-3}$$

The elements in the P-matrix are small. This would indicate good accuracy if the process had fulfilled the assumptions about the noise structure. Theorem 2.2 gives the following estimates of the standard deviation of $\hat{a}$ and $\hat{b}$:

$$\sigma_{\hat{a}} = 0.5\sqrt{7.10} \cdot 10^{-2} = 0.013$$

$$\sigma_{\hat{b}} = 0.5\sqrt{39.03} \cdot 10^{-2} = 0.031$$

It is thus deceptive to judge the accuracy of the estimates by only looking at the P-matrix. It is necessary that the data be generated from a model of the form (2.12) to use the P-matrix for accuracy estimates.

If we did not observe that the equation error is not white, we could thus be strongly misled. One possibility to avoid mistakes is to also compute the correlation of the equation error and check whether it is white noise. □

EXAMPLE 2.14 Closed-loop estimation

Example 2.10 showed that identifiability can be lost owing to feedback. The

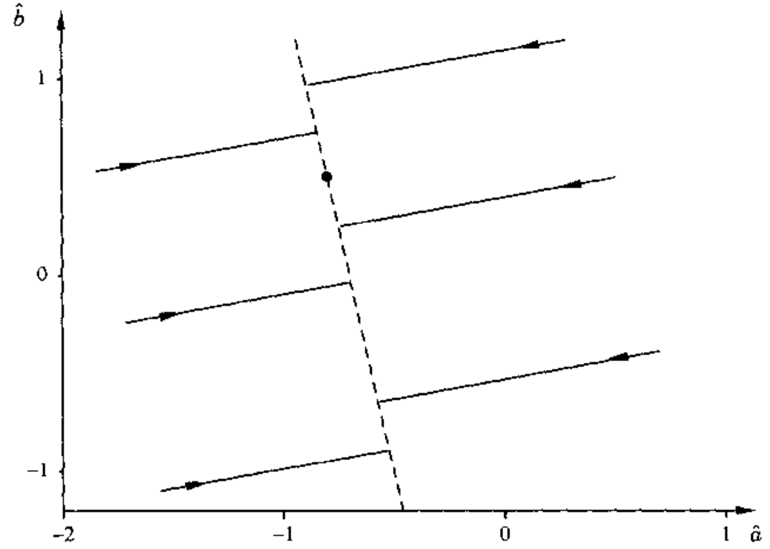

Figure 2.10 Phase plane of the estimates when the system (2.53) is simulated for different initial conditions when $u(t) = -0.2y(t)$. The dashed line shows the identifiable subspace. The dot shows the true parameter values.

estimates when the input is generated through the feedback

$$u(t) = -0.2y(t)$$

are shown in Fig. 2.9(a). The estimates converge to the wrong values. Notice, however, that the estimates are on the straight line (2.51). In Fig. 2.9(b) the feedback is more complex:

$$u(t) = -0.32y(t-1)$$

The two control laws give approximately the same speed and output variance of the closed-loop system. Identifiability is now regained, and the estimates converge to the correct values. The phase plane, that is, $\hat{b}$ as a function of $\hat{a}$, is shown in Fig. 2.10 for different initial conditions when $u(t) = -0.2y(t)$. The initial value of the P-matrix is $P(0) = 0.01I$, and 20,000 steps have been simulated for each initial condition. The estimates converge to the identifiable subspace determined by

$$\hat{a} + 0.2\hat{b} + 0.7 = 0$$

(Compare Eq. (2.51).) This line is dashed in the phase plane. The estimates are approaching the identifiable subspace along straight lines. The same initial conditions are simulated for the control law $u(t) = -0.32y(t-1)$ in Fig. 2.11. The estimates converge to the correct value $(-0.8, 0.5)$, independent of the initial values. □

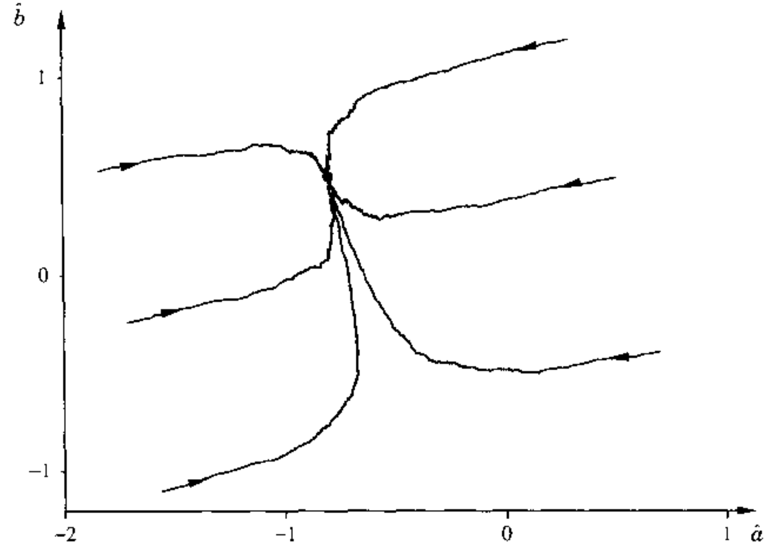

Figure 2.11 Phase plane of the estimates when the system (2.53) is simulated for different initial conditions when $u(t) = -0.32y(t-1)$. The dot shows the true parameter values.

EXAMPLE 2.15 Influence of forgetting factor

The recursive least-squares algorithm (2.21) has a forgetting factor λ. The influence of the forgetting factor is shown in Figure 2.12. When $\lambda = 1$, the estimates become smoother and smoother, since the gain $K(t)$ goes to zero. When $\lambda < 1$, the estimator gain $K(t)$ does not go to zero, and the estimates will always fluctuate. The fluctuations increase with decreasing λ. As a rule of thumb the "memory" of the estimator is

$$N = \frac{2}{1-\lambda}$$

For $\lambda = 0.99$ the estimates are based on approximately the last 200 steps. □

EXAMPLE 2.16 Different estimation methods

In the previous examples the RLS and ELS methods were used. Simplified estimation methods based on projection were discussed in Section 2.2. Three different projection algorithms will now be compared with the RLS method. All have the following form:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi(t)\left(y(t) - \varphi^T(t)\hat{\theta}(t-1)\right) \tag{2.54}$$

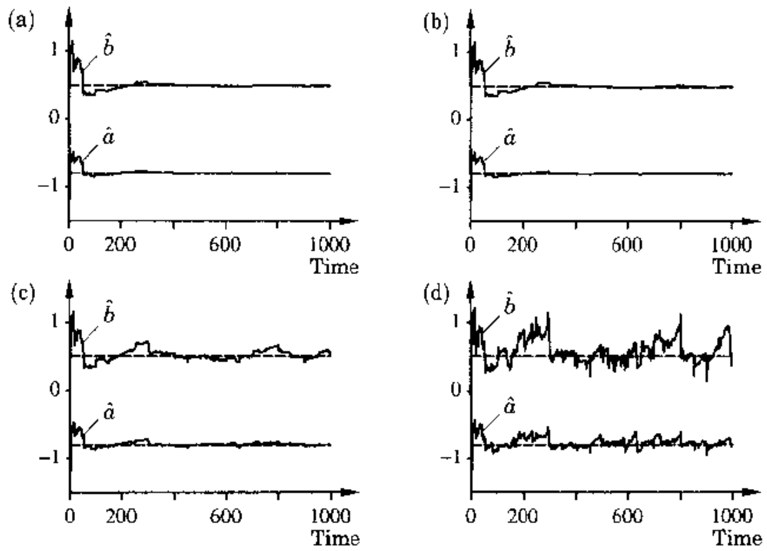

Figure 2.12 Estimates of the parameters in the process (2.53) when RLS is used and (a) $\lambda = 1$, (b) $\lambda = 0.999$, (c) $\lambda = 0.99$, and (d) $\lambda = 0.95$.

Compare with Eq. (2.24). The scalar gain $P(t)$ is given by the following algorithms.

Least mean squares (LMS):

$$P(t) = \gamma$$

Projection algorithm (PA):

$$P(t) = \frac{\gamma}{\alpha + \varphi^T(t)\varphi(t)} \qquad \alpha \geq 0, \quad 0 < \gamma < 2$$

Stochastic approximation (SA):

$$P(t) = \frac{\gamma}{\sum_{i=1}^{t} \varphi^T(i)\varphi(i)}$$

The convergence properties of the four algorithms RLS, LMS, PA, and SA are compared in Fig. 2.13. All algorithms are initialized with $\theta(0) = 0$. The RLS method in Fig. 2.13(a) uses $P(0) = 100I$ and $\lambda = 1$. Notice that the estimates move very quickly initially. The LMS method used in Fig. 2.13(b) has a constant gain $\gamma = 0.01$. The estimates approach values that are close to the correct ones relatively quickly, but the estimates do not converge, since the gain is not decreasing. In the PA method in Fig. 2.13(c) the gain is normalized with $\varphi^T(t)\varphi(t)$. Further, $\alpha = 0.1$ and $\gamma = 0.01$ are used. The approach toward

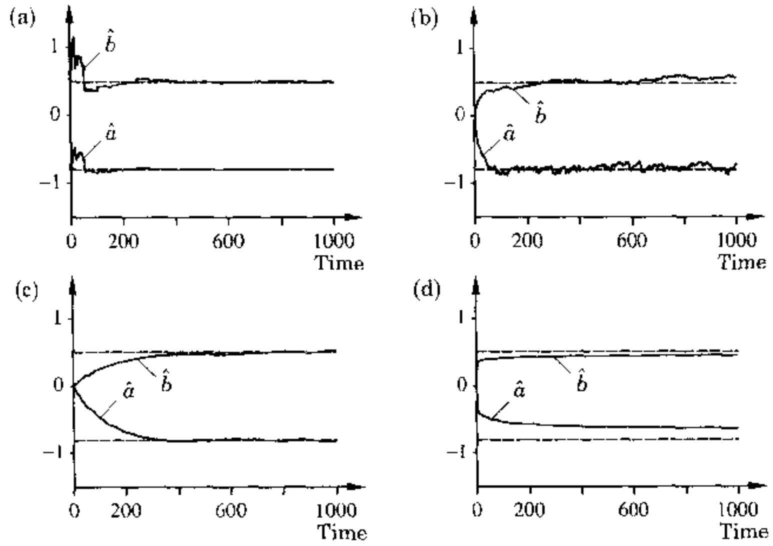

Figure 2.13 The estimates of the parameters in the process for different estimation methods. (a) Recursive least squares (RLS) with $P(0) = 100$ and $\lambda = 1$. (b) Least mean squares (LMS) with $\gamma = 0.01$. (c) Projection algorithm (PA) with $\alpha = 0.1$ and $\gamma = 0.01$. (d) Stochastic approximation (SA) with $\gamma = 0.2$.

the correct values is slower than for the LMS algorithm. However, the PA method is less sensitive than the LMS method to the size of the signals. The SA method is used in Fig. 2.13(d) with $\gamma = 0.2$, and the estimates converge to the correct values even if the convergence is very slow. About 25,000–50,000 steps are needed before the estimates are close to the correct values. From the simulations it is seen that the recursive least-squares method has superior convergence properties. The price for this is the increase in computations. □

The examples show that there are many things that influence the performance of the estimators. In adaptive control it is important to remember that the estimation is done in closed loop.

2.6 PRIOR INFORMATION

There is a significant advantage in incorporating available prior information. It reduces the number of parameters that have to be estimated, improves the precision of the estimates, and reduces the requirements on excitation.

Prior information typically relates to properties of a model. It can, for instance, represent knowledge of time constants of an actuator. This type of knowledge is easy to incorporate in an indirect adaptive algorithm. However, it may be difficult to incorporate in a direct adaptive algorithm, since process parameters influence controller parameters in a complicated fashion. Since prior knowledge is often related to the continuous-time models it is easier to use for continuous time than for discrete time self-tuners. These properties are highlighted by a few examples.

EXAMPLE 2.17 Prior information in continuous time

Consider the continuous-time system with the transfer function

$$G(s) = \frac{\theta_3}{(1+\theta_1 s)(1+\theta_2 s)}$$

The parameter θ_1 is assumed to be known; θ_2 and θ_3 are unknown. If we introduce the filtered signal $\bar{u}$ defined by

$$\bar{u} = \frac{1}{1+\theta_1 p} u$$

the input-output relation may be written as

$$y + \theta_2 \frac{dy}{dt} = \theta_3 \bar{u} \tag{2.55}$$

The estimation problem thus reduces to estimation of parameters θ_3 and θ_2 of the first-order system given by Eq. (2.55). □

The example thus shows that it is straightforward to handle prior information for the continuous-time model. The next example illustrates some complications that occur when the model is sampled.

EXAMPLE 2.18 Prior information in sampled models

Consider the system in Example 2.17. Sampling the system with sampling period h gives the pulse transfer operator

$$H(q) = \frac{b_1 q + b_2}{q^2 + a_1 q + a_2}$$

where

$$b_1 = \theta_3 \frac{\theta_1\left(1 - e^{-h/\theta_1}\right) - \theta_2\left(1 - e^{-h/\theta_2}\right)}{\theta_1 - \theta_2}$$

$$b_2 = \theta_3 \frac{\theta_2\left(1 - e^{-h/\theta_2}\right)e^{-h/\theta_1} - \theta_1\left(1 - e^{-h/\theta_1}\right)e^{-h/\theta_2}}{\theta_1 - \theta_2}$$

$$a_1 = -\left(e^{-h/\theta_1} + e^{-h/\theta_2}\right)$$

$$a_2 = e^{-(1/\theta_1 + 1/\theta_2)h}$$

The pulse transfer function is nonlinear in θ_1, θ_2, and θ_3. Further, both parameters appear in *all* the coefficients of the discrete-time pulse transfer function. This implies that a change in the unknown time constant θ_2 will influence all the coefficients in the sampled data model. There is, however, some structure in the parameter dependence. The denominator polynomial can be written as

$$\begin{aligned} q^2 + a_1 q + a_2 &= \left(q - e^{-h/\theta_1}\right)\left(q - e^{-h/\theta_2}\right) \\ &= \left(q - \alpha_1\right)\left(q - \alpha_2\right) \end{aligned}$$

When θ_1 is known, one factor of $A(q)$ is thus known. By reparameterization the sampled model can be written as

$$H(q) = \frac{b_1 q + b_2}{(q-\alpha_1)(q-\alpha_2)}$$

The prior information can be used to reduce the estimated parameters from 4 to 3. Further simplifications can be made when the sampling interval is small in comparison with θ_1 and θ_2. A series approximation of b_1 and b_2 in h gives

$$b_1 \approx \frac{\theta_3}{2\theta_1\theta_2} h^2 - \frac{\theta_3(\theta_1+\theta_2)}{6\theta_1^2\theta_2^2} h^3$$

$$b_2 \approx \frac{\theta_3}{2\theta_1\theta_2} h^2 - \frac{\theta_3(\theta_1+\theta_2)}{\theta_1^2\theta_2^2} h^3$$

For short sampling periods we have

$$b_1 \approx b_2 \approx \frac{\theta_3}{2\theta_1\theta_2} h^2$$

The model can now be described by

$$H(q) = \frac{k(q+1)}{(q-\alpha_1)(q-\alpha_2)}$$

where parameter α_1 is known and α_2 and $k = \theta_3/(h^2\theta_1\theta_2)$ are unknown. □

The observation about the structure of the sampled model for small sampling periods is a consequence of a general result about how poles and zeros are transformed by sampling. If α_i is a pole of a continuous-time system, then the sampled system has a pole at $\exp(\alpha_i h)$. There are no simple, exact formulas for transforming the zeros. For short sampling periods, however, a zero β_i is approximately transformed to $\exp(\beta_i h)$. If d is the pole excess of the continuous-time system, there will be $d-1$ additional zeros of the sampled system. The limiting positions of these zeros as the sampling period goes to zero are given by Theorem 6.9 in Chapter 6. In this way it is possible to use prior information in terms of poles and zeros both for continuous-time self-tuners and for discrete-time self-tuners with a short sampling period.

Example 2.18 shows that how the process model is parameterized is crucial. Different parameterizations can be attempted. This is illustrated by an example.

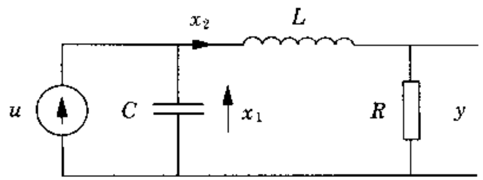

Figure 2.14 The circuit in Example 2.19.

EXAMPLE 2.19 Reparameterization

Consider the circuit in Fig. 2.14. The state space representation is

$$\frac{dx}{dt} = \begin{pmatrix} 0 & -1/C \\ 1/L & -R/L \end{pmatrix} x + \begin{pmatrix} 1/C \\ 0 \end{pmatrix} u$$
$$y = \begin{pmatrix} 0 & R \end{pmatrix} x$$

and the transfer function is

$$G(s) = \frac{\dfrac{R}{LC}}{s^2 + \dfrac{R}{L}s + \dfrac{1}{LC}}$$

Let $\theta_1 = R$, $\theta_2 = 1/L$, and $\theta_3 = 1/C$. Then

$$G(s) = \frac{\theta_1\theta_2\theta_3}{s^2 + \theta_1\theta_2 s + \theta_2\theta_3}$$

The coefficients are nonlinear (although of special structure) in the physical parameters R, $1/L$, and $1/C$. The system can be written as

$$G(s) = \frac{k_1}{s^2 + k_2 s + k_3} \tag{2.56}$$

and it is possible to make an estimation of θ_1, θ_2, and θ_3 by using Eq. (2.56). However, the estimates must be constrained such that the relations

$$k_1 = \theta_1\theta_2\theta_3$$
$$k_2 = \theta_1\theta_2$$
$$k_3 = \theta_2\theta_3$$

are fulfilled. □

For indirect self-tuning regulators it is possible to estimate the continuous-time process parameters from discrete-time measurements. The model can then be sampled and the controller designed for the chosen sampling interval.

2.7 CONCLUSIONS

In this chapter we have introduced recursive parameter estimation, which is a key ingredient in adaptive control. The presentation has been focused on the least-squares method, which is a simple but useful technique. In the next chapter we will show how the method is used in adaptive systems. System identification involves several important issues that we have not discussed. One is model validation; another is computational aspects. These issues are discussed in detail in Chapter 11.

PROBLEMS

2.1 Consider the function

$$V(x) = x^T Ax + b^T x + c$$

where x and b are column vectors, A is a matrix, and c is a scalar. Show that the gradient of function V with respect to x is given by

$$\text{grad}_x V = (A + A^T)x + b$$

This can be used to find the minimum of Eq. (2.7).

2.2 Consider the FIR model

$$y(t) = b_0 u(t) + b_1 u(t-1) + e(t) \qquad t = 1, 2, \ldots$$

where $\{e(t)\}$ is a sequence of independent normal $N(0, \sigma)$ random variables.

(a) Determine the least-squares estimate of the parameters b_0 and b_1 when the input signal u is a step. Analyze the covariance of the estimate when the number of observations goes to infinity. Relate the results to the notion of persistent excitation.

(b) Make the same investigation as in part (a) when the input signal is white noise with unit variance.

2.3 Consider data generated by the discrete-time system

$$y(t) = b_0 u(t) + b_1 u(t-1) + e(t)$$

where $\{e(t)\}$ is a sequence of independent $N(0, 1)$ random variables. Assume that the parameter b of the model

$$y(t) = bu(t)$$

is determined by least squares.

(a) Determine the estimates obtained for large observation sets when the input u is a step. (This is a simple illustration of the problem of fitting a low-order model to data generated by a complex model. The result obtained will critically depend on the character of the input signal.)

(b) Make the same investigation as in part (a) when the input signal is a sequence of independent $N(0, \sigma)$ random variables.

2.4 Determine which of the input signals below are persistently exciting of at least order 4.

(a)

$$u(t) = a_0 + a_1 \sin \omega t \qquad a_i \neq 0, \quad i = 0, 1$$

(b)

$$u(t) = \frac{q - 0.5}{(q - 0.4)(q - 0.6)} v(t)$$

where $v(t)$ is persistently exciting of order 5.

(c)

$$u(t) = \frac{q - 0.5}{(q - 0.4)(q - 0.6)} v(t)$$

where $v(t)$ has a spectrum $\Phi_v(\omega)$ that is not equal to zero in the interval $1 < \omega < 2$.

2.5 Consider the discrete-time system

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

where the input signal u and the noise e are sequences of independent random variables with zero mean values and standard deviation σ and 1. Determine the covariance of the estimates obtained for large observation sets.

2.6 Consider data generated by the least-squares model

$$y(t + 1) + ay(t) = bu(t) + e(t + 1) + ce(t) \qquad t = 1, 2, \ldots$$

where $\{u(t)\}$ and $\{e(t)\}$ are sequences of independent random variables with zero mean values and standard deviations 1 and σ. Assume that parameters a and b of the model

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

are estimated by least squares. Determine the asymptotic values of the estimates.

2.7 Consider least-squares estimation of the parameters b_1 and b_2 in

$$y(t) = b_1 u(t) + b_2 u(t - 1)$$

Assume that the following measurements are obtained:

t	u	y
1	1000	–
2	1001	2001
3	1000	2001

Discuss the numerical properties of computing the estimates directly and by the normal equations.

2.8 Consider the model

$$y(t) = a + b \cdot t + e(t) \qquad t = 1, 2, 3, \ldots$$

where $\{e(t)\}$ is a sequence of uncorrelated $N(0, 1)$ random variables. Determine the least-squares estimate of the parameters a and b. Also determine the covariance of the estimate. Discuss the behavior of the covariance as the number of estimates increases.

2.9 Consider the model in Problem 2.8, but assume continuous-time observation, where $e(t)$ is white noise, that is, a random function with covariance $\delta(t)$. Determine the estimate and its covariance. Analyze the behavior of the covariance for large observation intervals.

2.10 Consider data generated by

$$y(t) = b + e(t) \qquad t = 1, 2, \ldots, N$$

where $\{e(t); t = 1, 3, 4, \ldots\}$ is a sequence of independent random variables. Furthermore, assume that there is a large error at $t = 2$, that is,

$$e(2) = a$$

where a is a large number. Assume that the parameter b in the model

$$y(t) = b$$

is estimated by least squares. Determine the estimate obtained, and discuss how it depends on a. (This is a simple example that shows how sensitive the least-squares estimate is with respect to occasional large errors.)

2.11 Consider Example 2.12. Analyze the asymptotic properties of the P-matrix and explain the simulation in Figs. 2.7(a) and 2.7(b).

2.12 Show that Eq. (2.11) minimizes the weighted least-squares loss function (2.10).

2.13 Consider Eqs. (2.21) with the initial condition $\hat{\theta}(0) = \theta_0$ and $P(0) = P_0$. Show that $\hat{\theta}(t)$ minimizes the criterion

$$V(\theta, t) = \frac{1}{2}\sum_{i=1}^{t} \lambda^{t-i}\left(y(i) - \varphi^T(i)\theta\right)^2 + \frac{\lambda^t}{2}(\theta - \theta_0)^T P_0^{-1}(\theta - \theta_0)$$

Compare Theorem 2.4.

2.14 Consider the following model of time-varying parameters:

$$\theta(t) = \Phi_v \theta(t-1) + v(t)$$
$$y(t) = \varphi^T(t)\theta(t) + e(t)$$

where $\{v(t), t = 1, 2, \ldots\}$ and $\{e(t), t = 1, 2, \ldots\}$ are sequences of independent, equally distributed random vectors with zero mean values and covariances R_1 and R_2, respectively. Show that the recursive estimates of θ are given by

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\left(y(t) - \varphi^T(t)\hat{\theta}(t-1)\right)$$
$$K(t) = \Phi_v P(t-1)\varphi(t-1)\left(R_2 + \varphi^T(t-1)P(t-1)\varphi(t-1)\right)^{-1}$$
$$P(t) = \Phi_v P(t-1)\Phi_v^T + R_1 - \Phi_v P(t-1)\varphi(t)\left(R_2 + \varphi^T(t)P(t-1)\varphi(t)\right)^{-1}\varphi^T(t)P(t-1)\Phi_v^T$$

2.15 Show that Eq. (2.28) minimizes Eq. (2.27), and use this to prove Theorem 2.5. *Hint:* Use Remark 1 in Theorem 2.5 and that the time derivative of the identity $I = PP^{-1}$ is

$$\frac{dP}{dt} = -P\frac{d(P^{-1})}{dt}P$$

2.16 In an adaptive controller the process parameters are estimated according to the model

$$y(t) + a_1 y(t-1) + a_2 y(t-2) = b_0 u(t-1) + b_1 u(t-2) + e(t)$$

The controller has the structure

$$u(t) + r_1 u(t-1) = -s_0 y(t) - s_1 y(t-1)$$

The reference value is thus zero. Consider the case in which the controller parameters are constant.

(a) Show that the parameters a_1, a_2, b_0, and b_1 cannot be uniquely determined.

(b) Characterize the parameter combinations that can be determined.

(c) Show that with the controller structure

$$u(t) + r_1 u(t-1) + r_2 u(t-2) = -s_0 y(t) - s_1 y(t-1) - s_2 y(t-2)$$

all process parameters can be estimated uniquely.

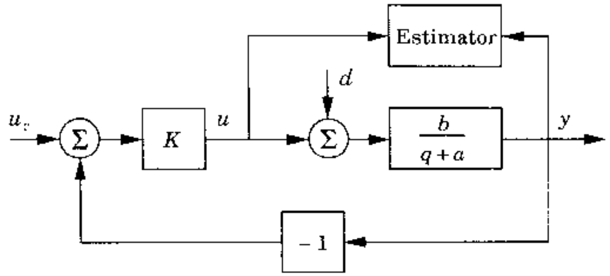

Figure 2.15 Closed-loop estimation scheme for Problem 2.17.

2.17 Figure 2.15 shows a closed-loop system for estimation of the unknown constant b in the pulse transfer function $H(q) = b/(q+a)$. The constant a is known and is such that $|a| > 1$. This means that the open-loop system is unstable, and to have bounded signals for the estimation, we need to stabilize the system with a controller. This is done with a P controller with gain K such that $|a + Kb| < 1$. The estimator is a least-squares (LS) estimator that is based on the regression model

$$\bar{y}(t) = \varphi^T(t-1)\theta$$

where

$$\bar{y}(t) = y(t) + ay(t-1)$$
$$\varphi(t-1) = u(t-1)$$
$$\theta = b$$

(a) Determine the asymptotic LS estimate of $\theta = b$ when $d = 0$ and $\{u_c\}$ is a sequence of independent, equally distributed random variables with zero mean and variance σ^2 (i.e., u_c is a white noise signal).

(b) Determine the asymptotic LS estimate of b when $u_c = 0$ and $d = d_0 =$ constant.

(c) Discuss the case of least-squares estimation of b when u_c is as in part (a) and $d = d_0 =$ constant. What should be done to avoid a biased estimate of b?

2.18 Write a computer program to simulate the recursive least-squares estimation problem. Write the program so that arbitrary input signals can be used. Use the program to investigate the effects of initial values on the estimate.

2.19 Use the program from Problem 2.18 to estimate the parameters a and b in Problem 2.6. Investigate how the bias of the estimate depends on c.

2.20 Consider the estimation problem in Problem 2.6. Use the computer program developed in Problem 2.18 to explore what happens when the control signal u is generated by the feedback

$$u(t) = -ky(t)$$

Try to support your observations by analysis.

2.21 Consider the open-loop system in Section 2.5 when $c = 0$. Let the input signal be a square wave with unit amplitude and a period of 100 samples. Investigate through simulations the convergence and behavior of the parameter estimates when varying:

(a) The initial value $\hat{\theta}(0)$.

(b) The initial value of the covariance matrix $P(0)$.

(c) The forgetting factor λ.

(d) The period of the input signal.

REFERENCES

The following textbooks can be recommended for those who would like to learn more about system identification:

Norton, J. P., 1986. *An Introduction to Identification.* London: Academic Press.

Ljung, L., 1987. *System Identification—Theory for the User.* Englewood Cliffs, N.J.: Prentice-Hall.

Söderström, T., and P. Stoica, 1988. *System Identification.* Hemel Hempstead, U.K.: Prentice-Hall International.

Johansson, R., 1992. *System Modeling and Identification.* Englewood Cliffs, N.J.: Prentice-Hall.

The regression model is commonly used in many branches of applied mathematics. See, for example:

Draper, N. R., and H. Smith, 1981. *Applied Regression Analysis,* 2nd edition. New York: John Wiley.

Recursive identification and properties of recursive estimators are treated in depth in:

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.

Properties of identification in closed-loop systems are found in:

Wellstead, P. E., and J. M. Edmunds, 1975. "Least-squares identification of closed loop systems." *Int. J. of Control* **21**: 689–699.

Gustafsson, I., L. Ljung, and T. Söderström, 1977. "Identification of processes in closed-loop—Identification and accuracy aspects." *Automatica* **13**: 59–75.

Good sources are also the proceedings of the IFAC symposia on system identification that have been held every third year since 1967.

The least-squares method was first presented in:

Gauss, K. F., 1809. *Theoria motus corposum coelestium,* (In Latin). English translation: *Theory of the Motion of the Heavenly Bodies.* New York: Dover, 1963.

The numerical solution to least-squares problems is well treated in:

Lawson, C. L., and R. J. Hanson, 1974. *Solving Least Squares Problems.* Englewood Cliffs, N.J.: Prentice-Hall.

Recursive square root algorithms are discussed in:

Bierman, G. J., 1977. *Factorization Methods for Discrete Sequential Estimation.* New York: Academic Press.

The exponential weighting of data in the least-squares estimation was first introduced in:

Plackett, R. L., 1950. "Some theorems in least squares." *Biometrika* **37**: 149–157.

Different ways to modify recursive estimators to follow time-varying parameters are suggested in:

Irving, E., 1980. "New developments in improving power network stability with adaptive generator control." In *Applications of Adaptive Control,* eds. K. S. Narendra and R. V. Monopoli. New York: Academic Press.

Fortescue, T. R., L. S. Kershenbaum, and B. E. Ydstie, 1981. "Implementation of self-tuning regulators with variable forgetting factors." *Automatica* **17**: 831–835.

Kulhavý, R., and M. Kárný, 1984. "Tracking of slowly varying parameters by directional forgetting." Paper 14.4/E-4, *9th IFAC World Congress.* Budapest.

Hägglund, T., 1985. "Recursive estimation of slowly time-varying parameters." *Preprints 7th IFAC Symposium on Identification and System Parameter Estimation,* pp. 1255–1260. York, U.K.

Kulhavý, R., 1987. "Restricted exponential forgetting in real-time identification." *Automatica* **23**: 589–600.

The Kaczmarz's algorithm was first published in German in 1937

Kaczmarz, S., 1937. "Angenäherte Auflösung von Systemen linearer Gleichunger." *Bulletin International de l'Academie Polonaise des Sciences. Lett A*: 355–357.

An English translation of the original paper is found in

Kaczmarz, S., 1993. "Approximate solution of systems of linear equations." *Int. J. Control* **57**: 1269–1271.

Estimation of continuous-time models is treated in, for instance:

Young, P. C., 1981. "Parameter estimation for continuous-time models: A survey." *Automatica* **17**: 23–29.

Unbehauen, H., and G. P. Rao, 1987. *Identification of Continuous-Time Systems.* Amsterdam: North-Holland.

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

The LMS method is extensively treated in:

Widrow, B., and S. D. Stearns, 1985. *Adaptive Signal Processing.* Englewood Cliffs, N.J.: Prentice-Hall.

Haykins, S., 1991. *Adaptive Filter Theory,* 2nd edition. Englewood Cliffs, N.J.: Prentice-Hall.

A tutorial survey of algorithms for tracking time-varying systems is found in:

Ljung, L., and S. Gunnarsson, 1990. "Adaptation and tracking in system identification–A survey." *Automatica* **26**: 7–21.

CHAPTER 3

DETERMINISTIC SELF-TUNING REGULATORS

3.1 INTRODUCTION

Development of a control system involves many tasks such as modeling, design of a control law, implementation, and validation. The *self-tuning regulator* (STR) attempts to automate several of these tasks. This is illustrated in Fig. 3.1, which shows a block diagram of a process with a self-tuning regulator. It is assumed that the structure of a process model is specified. Parameters of the model are estimated on-line, and the block labeled "Estimation" in Fig. 3.1 gives an estimate of the process parameters. This block is a recursive estimator of the type discussed in Chapter 2. The block labeled "Controller design" contains computations that are required to perform a design of a controller with a specified method and a few design parameters that can be chosen externally. The design problem is called the *underlying design problem* for systems with known parameters. The block labeled "Controller" is an implementation of the controller whose parameters are obtained from the control design.

The name "self-tuning regulator" comes from one of the early papers. The main reason for using an adaptive controller is that the process or its environment is changing continuously. It is difficult to analyze such systems. To simplify the problem, it can be assumed that the process has constant but unknown parameters. The term *self-tuning* was used to express the property that the controller parameters converge to the controller that was designed if the process was known. An interesting result was that this could happen even if the model structure was incorrect.

The tasks shown in the block diagram can be performed in many different ways. There are many possible choices of model and controller structures.

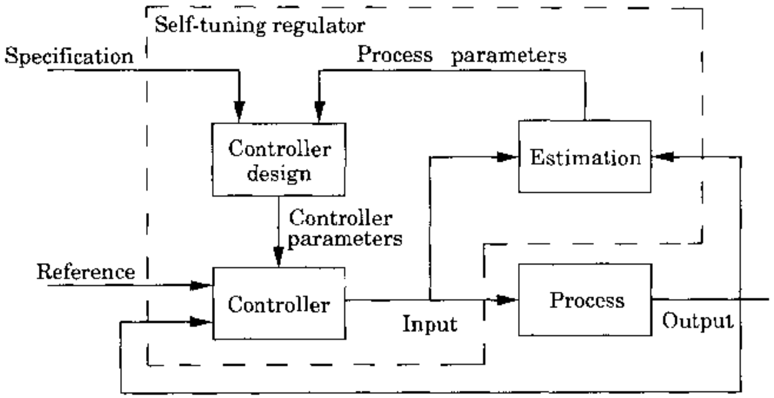

Figure 3.1 Block diagram of a self-tuning regulator.

Estimation can be performed continuously or in batches. In digital implementations, which are most common, different sampling rates can be used for the controller and the estimator. It is also possible to use hybrid schemes in which control is performed continuously and the parameters are updated discretely. Parameter estimation can be done in many ways, as was discussed in Chapter 2. There is also a large variety of techniques that can be used for control system design. It is also possible to consider nonlinear models and nonlinear design techniques. Although many estimation methods will provide estimates of parameter uncertainties, these are typically not used in the control design. The estimated parameters are treated as if they are true in designing the controller. This is called the *certainty equivalence principle*.

The controller shown in Fig. 3.1 is thus a very rich structure. Only a few possibilities have been investigated. The choice of model structure and its parameterization are important issues for self-tuning regulators. A straightforward approach is to estimate the parameters of the transfer function of the process. This gives an *indirect adaptive algorithm*. The controller parameters are not updated directly, but rather indirectly via the estimation of the process model.

Often, the model can be reparameterized such that the controller parameters can be estimated directly. That is, a *direct adaptive algorithm* is obtained (compare with the discussion of the direct MRAS in Section 1.4). There has been some confusion in the nomenclature. In the self-tuning context, indirect methods have often been called *explicit* self-tuning control, since the process parameters have been estimated. Direct updating of the controller parameters has been called *implicit* self-tuning control. In the early papers on adaptive control a direct adaptive controller was often referred to as an adaptive controller without identification. It is convenient to divide the algorithms into indirect

and direct self-tuners, but the distinction should not be overemphasized. The basic idea in both types of algorithms is to identify some parameters that are related to the process and/or the specifications of the closed-loop system.

The purpose of this chapter is to present the basic ideas and to illustrate some properties of self-tuning regulators. It is assumed that the process model and the controller are linear systems. The discussion will also be restricted to single-input, single-output (SISO) systems. In most cases we will assume that the controller is sampled and that estimation and control are performed with the same sampling rates. Recursive least squares will be used for parameter estimation, and the design method is a deterministic pole placement. The reasons for these choices are mostly didactic; we would like to present simple methods that can be used in practice. Least-squares estimation was discussed in Chapter 2. In Section 3.2 we present the design method used in a simple setting. A straightforward combination of least-squares estimation and pole placement design gives an indirect self-tuning regulator. The sampled version is described in Section 3.3, and the continuous-time version is described in Section 3.4. In Section 3.5 we show how a direct self-tuning regulator is obtained. In this section we also discuss hybrid algorithms that combine features of direct and indirect algorithms. In Section 3.6 we discuss how to modify the adaptive controllers so that they can deal with disturbances.

3.2 POLE PLACEMENT DESIGN

A simple method for control design will now be presented. The idea is to determine a controller that gives desired closed-loop poles. In addition it is required that the system follows command signals in a specified manner. This is a simple method that, properly applied, can give practically useful controllers as well as useful understanding of adaptive control. It is also the key to understand the similarities between the self-tuning regulator and the model reference adaptive controller.

Process Model

It is assumed that the process is described by the single-input, single-output (SISO) system

$$A(q)y(t) = B(q)\,(u(t) + v(t))$$

where y is the output, u is the input of the process, and v is a disturbance. The disturbances can enter the system in many ways. Here it has been assumed that they enter at the process input. For linear systems in which the superposition principle holds, an equivalent input disturbance can always be found. Furthermore, A and B are polynomials in the forward shift operator q. The polynomials have the degrees $\deg A = n$ and $\deg B = \deg A - d_0$. Pa-

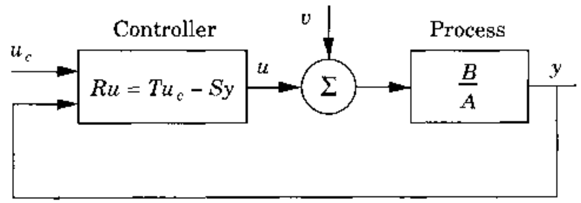

Figure 3.2 A general linear controller with two degrees of freedom.

rameter d_0, which is called the *pole excess*, represents the integer part of the ratio of time delay and sampling period. It is sometimes convenient to write the process model in the delay operator q^{-1}. This can be done by introducing the *reciprocal polynomial*

$$A^*(q^{-1}) = q^{-n}A(q)$$

where $n = \deg A$. The model can then be written as

$$A^*(q^{-1})y(t) = B^*(q^{-1})\,(u(t-d_0) + v(t-d_0))$$

where

$$A^*(q^{-1}) = 1 + a_1q^{-1} + \ldots + a_nq^{-n}$$
$$B^*(q^{-1}) = b_0 + b_1q^{-1} + \ldots + b_mq^{-m}$$

with $m = n - d_0$. Notice that since n was defined as the degree of the system, we have $n \geq m + d_0$, and trailing coefficients of A^* may thus be zero.

We will mostly deal with discrete time systems. Since the design method is purely algebraic, we can handle continuous systems simultaneously by writing the model

$$Ay(t) = B\,(u(t) + v(t)) \tag{3.1}$$

where A and B denote polynomials in either the differential operator $p = d/dt$ or the forward shift operator q. It is assumed that A and B are *relatively prime*, that is, that they do not have any common factors. Further, it is assumed that A is *monic*, that is, that the coefficient of the highest power in A is unity.

A general linear controller can be described by

$$Ru(t) = Tu_c(t) - Sy(t) \tag{3.2}$$

where R, S, and T are polynomials. This control law represents a negative feedback with the transfer operator S/R and a feedforward with the transfer operator T/R. It thus has two degrees of freedom. A block diagram of the closed-loop system is shown in Fig. 3.2. Elimination of u between Eqs. (3.1)

and (3.2) gives the following equations for the closed-loop system:

$$\begin{aligned} y(t) &= \frac{BT}{AR+BS}u_c(t) + \frac{BR}{AR+BS}v(t) \\ u(t) &= \frac{AT}{AR+BS}u_c(t) - \frac{BS}{AR+BS}v(t) \end{aligned} \tag{3.3}$$

The closed-loop characteristic polynomial is thus

$$AR + BS = A_c \tag{3.4}$$

The key idea of the design method is to specify the desired closed-loop characteristic polynomial A_c. The polynomials R and S can then be solved from Eq. (3.4). Notice that in the design procedure we consider polynomial A_c to be a design parameter that is chosen to give desired properties to the closed-loop system. Equation (3.4), which plays a fundamental role in algebra, is called the *Diophantine equation*. It is also called the *Bezout identity* or the *Aryabhatta equation*. The equation always has solutions if the polynomials A and B do not have common factors. The solution may be poorly conditioned if the polynomials have factors that are close. The solution can be obtained by introducing polynomials with unknown coefficients and solving the linear equations obtained. The solution of the equation is discussed in detail in Chapter 11.

Model-Following

The Diophantine equation (3.4) determines only the polynomials R and S. Other conditions must be introduced to also determine the polynomial T in the controller (3.2). To do this, we will require that the response from the command signal u_c to the output be described by the dynamics

$$A_my_m(t) = B_mu_c(t) \tag{3.5}$$

It then follows from Eqs. (3.3) that the following condition must hold:

$$\frac{BT}{AR+BS} = \frac{BT}{A_c} = \frac{B_m}{A_m} \tag{3.6}$$

This model-following condition says that the response of the closed-loop system to command signals is as specified by the model (3.5). Whether model-following can be achieved depends on the model, the system, and the command signal. If it is possible to make the error equal to zero for all command signals, then *perfect model-following* is achieved.

The consequences of the model-following condition will now be explored. Equation (3.6) implies that there are cancellations of factors of BT and A_c. Factor the B polynomial as

$$B = B^+B^- \tag{3.7}$$

where B^+ is a monic polynomial whose zeros are stable and so well damped that they can be canceled by the controller and B^- corresponds to unstable or poorly damped factors that cannot be canceled. It thus follows that B^- must be a factor of B_m. Hence

$$B_m = B^- B'_m \tag{3.8}$$

Since B^+ is canceled, it must be a factor of A_c. Furthermore, it follows from Eq. (3.6) that A_m must also be a factor of A_c. The closed-loop characteristic polynomial thus has the form

$$A_c = A_o A_m B^+ \tag{3.9}$$

Since B^+ is a factor of B and A_c, it follows from Eq. (3.4) that it also divides R. Hence

$$R = R' B^+ \tag{3.10}$$

and the Diophantine equation (3.4) reduces to

$$AR' + B^- S = A_o A_m = A'_c \tag{3.11}$$

Introducing Eqs. (3.7), (3.8), and (3.9) into Eq. (3.6) gives

$$T = A_o B'_m \tag{3.12}$$

Causality Conditions

To obtain a controller that is causal in the discrete-time case or proper in the continuous-time case, we must impose the conditions

$$\begin{aligned} \deg S &\le \deg R \\ \deg T &\le \deg R \end{aligned} \tag{3.13}$$

The Diophantine equation (3.4) has many solutions because if R^0 and S^0 are solutions, then so are

$$\begin{aligned} R &= R^0 + QB \\ S &= S^0 - QA \end{aligned} \tag{3.14}$$

where Q is an arbitrary polynomial. Since there are many solutions, we may select the solution that gives a controller of lowest degree. We call this the minimum-degree solution. Since $\deg A > \deg B$, the term of highest order on the left-hand side of Eq. (3.4) is AR. Hence

$$\deg R = \deg A_c - \deg A$$

Because of Eqs. (3.14) there is always a solution such that $\deg S < \deg A = n$. We can thus always find a solution in which the degree of S is at most $\deg A - 1$. This is called the *minimum-degree solution* to the Diophantine equation. The condition $\deg S \le \deg R$ thus implies that

$$\deg A_c \ge 2 \deg A - 1$$

It follows from Eq. (3.12) that the condition $\deg T \le \deg R$ implies that

$$\deg A_m - \deg B'_m \ge \deg A - \deg B^+$$

Adding $\deg B^-$ to both sides, we find that this is equivalent to $\deg A_m - \deg B_m \ge d_0$. This means that in the discrete-time case the time delay of the model must be at least as large as the time delay of the process, which is a very natural condition. Summarizing, we find that the causality conditions (3.13) can be written as

$$\begin{aligned} \deg A_c &\ge 2 \deg A - 1 \\ \deg A_m - \deg B_m &\ge \deg A - \deg B = d_0 \end{aligned} \tag{3.15}$$

It is natural to choose a solution in which the controller has the lowest possible degree. In the discrete-time case it is also reasonable to require that there be no extra delay in the controller. This implies that polynomials R, S, and T should have the same degrees. The following design procedure is then obtained.

ALGORITHM 3.1 Minimum-degree pole placement (MDPP)

Data: Polynomials A, B.

Specifications: Polynomials A_m, B_m, and A_o.

Compatibility Conditions:

$$\begin{aligned} \deg A_m &= \deg A \\ \deg B_m &= \deg B \\ \deg A_o &= \deg A - \deg B^+ - 1 \\ B_m &= B^- B'_m \end{aligned}$$

Step 1: Factor B as $B = B^+ B^-$, where B^+ is monic.

Step 2: Find the solution R' and S with $\deg S < \deg A$ from

$$AR' + B^- S = A_o A_m$$

Step 3: Form $R = R' B^+$ and $T = A_o B'_m$, and compute the control signal from the control law

$$Ru = Tu_c - Sy$$

□

There are special cases of the design procedure that are of interest.

All zeros are canceled The design procedure simplifies significantly in the special case in which all process zeros are canceled; then $\deg A_o = \deg A - \deg B - 1$. It is natural to choose $B_m = A_m(1) q^{d_0}$. Then the factorization in Step 1 is very simple, and we get $B^- = b_0$, $B^+ = B/b_0$. Furthermore,

$T = A_m(1)q^{d_0}/b_0$, and the closed-loop characteristic polynomial becomes $A_c = B^+A_c'$. The Diophantine equation in Step 2 reduces to

$$AR' + b_0S = A_c' = A_oA_m$$

This equation is easy to solve because R' is the quotient and b_0S is the remainder when A_oA_m is divided by A. However, all process zeros must be stable and well damped to allow cancellation.

No zeros are canceled The factorization in Step 2 also becomes very simple if no zeros are canceled. We have $B^+ = 1$, $B^- = B$, and $B_m = \beta B$, where $\beta - A_m(1)/B(1)$. Furthermore, $\deg A_o = \deg A - \deg B - 1$ and $T = \beta A_o$. The closed-loop characteristic polynomial is $A_c = A_oA_m$, and the Diophantine equation in Step 2 becomes

$$AR + BS = A_c = A_oA_m$$

Examples

The model-following design is illustrated by three examples

EXAMPLE 3.1 Model-following with zero cancellation

Consider a continuous-time process described by the transfer function

$$G(s) = \frac{1}{s(s+1)} \tag{3.16}$$

This can be regarded as a normalized model for a motor. The pulse transfer operator for the sampling period $h = 0.5$ s is

$$H(q) = \frac{B(q)}{A(q)} = \frac{b_0q + b_1}{q^2 + a_1q + a_2} = \frac{0.1065q + 0.0902}{q^2 - 1.6065q + 0.6065} \tag{3.17}$$

We have $\deg A = 2$ and $\deg B = 1$. The design procedure thus gives a first-order controller, and the closed-loop system will be of third order. The sampled data system has a zero in -0.84 and poles in 1 and 0.61. Let the desired closed-loop system be

$$\frac{B_m(q)}{A_m(q)} = \frac{b_{m0}q}{q^2 + a_{m1}q + a_{m2}} = \frac{0.1761q}{q^2 - 1.3205q + 0.4966} \tag{3.18}$$

This corresponds to a natural frequency of 1 rad/s and a relative damping of 0.7. Parameter b_{m0} is chosen so that the static gain is unity. This model satisfies the compatibility conditions because it has the same pole excess as the process and the process zero is stable although poorly damped. To apply the design procedure in Algorithm 3.1, we first factor the polynomial B, and we obtain

$$B^+(q) = q + b_1/b_0$$
$$B^-(q) = b_0$$
$$B_m'(q) = b_{m0}q/b_0$$

Since the process is of second order, the polynomials R, S, and T will all be of first order. Polynomial R' is thus of degree zero. Since the polynomial is monic, we have $R' = 1$. Since $\deg B^+ = 1$, it follows from the compatibility conditions that $\deg A_o - 0$. Choose

$$A_o(q) = 1$$

The Diophantine equation (3.11) then becomes

$$(q^2 + a_1q + a_2)\cdot 1 + b_0(s_0q + s_1) = q^2 + a_{m1}q + a_{m2}$$

Equating coefficients of equal power of q gives

$$a_1 + b_0s_0 = a_{m1}$$
$$a_2 + b_0s_1 = a_{m2}$$

These equations can be solved if $b_0 \neq 0$. The solution is

$$s_0 = \frac{a_{m1} - a_1}{b_0}$$
$$s_1 = \frac{a_{m2} - a_2}{b_0}$$

The controller is thus characterized by the polynomials

$$R(q) = B^+ = q + \frac{b_1}{b_0}$$
$$S(q) = s_0q + s_1$$
$$T(q) = A_oB_m' = \frac{b_{m0}q}{b_0}$$

□

The process in Example 3.1 has a zero that is stable but poorly damped. The continuous-time equivalent corresponds to a zero with relative damping $\zeta = 0.06$. We will therefore also determine a controller that does not cancel the zero. This is done in the next example.

EXAMPLE 3.2 Model-following without zero cancellation

Consider the same process as in Example 3.1, but use a control design in which there is no cancellation of the process zero. Since the process is of second order, the minimum-degree solution has polynomials R, S, and T of first order and the closed-loop system will be of third order. Since no zero is canceled, it follows from the compatibility condition in Algorithm 3.1 that $\deg A_0 = 1$. Since no process zeros are canceled, we have

$$B^+ = 1$$
$$B^- = B = b_0q + b_1$$

It also follows from the compatibility conditions that the model must have the same zero as the process. The desired closed-loop transfer operator is thus

$$H_m(q) = \beta\frac{b_0q + b_1}{q^2 + a_{m1}q + a_{m2}} = \frac{b_{m0}q + b_{m1}}{q^2 + a_{m1}q + a_{m2}}$$

where $b_{m0} = \beta b_0$ and

$$\beta = \frac{1 + a_{m1} + a_{m2}}{b_0 + b_1}$$

which gives unit steady state gain. The Diophantine equation (3.4) becomes

$$(q^2 + a_1 q + a_2)(q + r_1) + (b_0 q + b_1)(s_0 q + s_1) = (q^2 + a_{m1} q + a_{m2})(q + a_o) \quad (3.19)$$

Putting $q = -b_1/b_0$ and solving for r_1, we get

$$\begin{aligned} r_1 &= \frac{b_1}{b_0} + \frac{(b_1^2 - a_{m1}b_0b_1 + a_{m2}b_0^2)(-b_1 + a_o b_0)}{b_0(b_1^2 - a_1b_0b_1 + a_2b_0^2)} \\ &= \frac{a_o a_{m2} b_0^2 + (a_2 - a_{m2} - a_o a_{m1})b_0b_1 + (a_o + a_{m1} - a_1)b_1^2}{b_1^2 - a_1b_0b_1 + a_2b_0^2} \end{aligned} \quad (3.20)$$

Notice that the denominator is zero if polynomials $A(q)$ and $B(q)$ have a common factor. Equating coefficients of terms q^2 and q^0 in Eq. (3.19) gives

$$\begin{aligned} s_0 &= \frac{b_1(a_o a_{m1} - a_2 - a_{m1}a_1 + a_1^2 + a_{m2} - a_1 a_o)}{b_1^2 - a_1b_0b_1 + a_2b_0^2} \\ &\quad + \frac{b_0(a_{m1}a_2 - a_1a_2 - a_o a_{m2} + a_o a_2)}{b_1^2 - a_1b_0b_1 + a_2b_0^2} \\ s_1 &= \frac{b_1(a_1a_2 - a_{m1}a_2 + a_o a_{m2} - a_o a_2)}{b_1^2 - a_1b_0b_1 + a_2b_0^2} \\ &\quad + \frac{b_0(a_2 a_{m2} - a_2^2 - a_o a_{m2} a_1 + a_o a_2 a_{m1})}{b_1^2 - a_1b_0b_1 + a_2b_0^2} \end{aligned} \quad (3.21)$$

Furthermore, it follows from Eq. (3.12) that

$$T(q) = \beta A_o(q) = \beta(q + a_o)$$ □

Since the design method is purely algebraic, there is no difference between discrete-time systems and continuous-time systems. We illustrate this by an example.

EXAMPLE 3.3 Continuous-time system

The process discussed in Examples 3.1 and 3.2 has the transfer function

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

with $a = 1$ and $b = 1$. The design procedure given by Algorithm 3.1 will now be used to find a continuous-time controller. Since the process is of second order, the closed-loop system will be of third order and the minimum-degree controller is of first order. Polynomial A_m has degree two, B_m is a constant, and A_o has degree one. We choose

$$A_o(s) = s + a_o$$

and let the desired response be specified by the transfer function.

$$\frac{B_m(s)}{A_m(s)} = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

The Diophantine equation (3.4) becomes

$$s(s + a)(s + r_1) + b(s_0 s + s_1) = (s^2 + 2\zeta\omega s + \omega^2)(s + a_o)$$

Equating coefficients of equal powers of s gives the equations

$$\begin{aligned} a + r_1 &= 2\zeta\omega + a_o \\ ar_1 + bs_0 &= \omega^2 + 2\zeta\omega a_o \\ bs_1 &= \omega^2 a_o \end{aligned}$$

If $b \neq 0$, these equations can be solved, and we get

$$\begin{aligned} r_1 &= 2\zeta\omega + a_o - a \\ s_0 &= \frac{a_o 2\zeta\omega + \omega^2 - ar_1}{b} \\ s_1 &= \frac{\omega^2 a_o}{b} \end{aligned}$$

Furthermore, we have $B^+ = 1$, $B^- = b$, and $B'_m = \omega^2/b$. It then follows from Eq. (3.12) that

$$T(s) = B'_m(s)A_o(s) = \frac{\omega^2}{b}(s + a_o)$$ □

An Interpretation of Polynomial A_o

It is possible to give an interpretation of the polynomial A_o that appears in the minimum-degree pole placement solution in the case in which no process zeros are canceled. To do this, we observe that the pole placement problem can also be solved with state feedback and an observer. The closed-loop dynamics are then composed of two parts: one that corresponds to the state feedback and another that corresponds to the observer dynamics. For a system of degree n it is also known that it is sufficient to use an observer of degree $n - 1$. When no process zeros are canceled, the closed-loop characteristic polynomial in our case is A_mA_o, where A_m is of degree n and A_o is of degree $n - 1$. By this analogy we can interpret the polynomial A_m as being associated with the state feedback and A_o as being associated with the observer. We will therefore call A_o the *observer polynomial*. In a system with state feedback it is also natural to introduce the command signals in such a way that they do not generate observer errors. This means that the observer polynomial is canceled in the transfer function from command signal to process output.

Relations to Model-Following

Many other design methods can be related to pole placement. We will now show that pole placement can be interpreted as a model-following design. This is of interest because much work on MRAS is formulated in terms of model-following. Model-following generally means that the response of a closed-loop system to command signals is specified by a given model. This means that both poles and zeros of the model are specified by the user. Pole placement, on the other hand, specifies only the closed-loop poles. In the minimum-degree pole placement procedure we did, however, introduce some auxiliary conditions that included the process zeros. We will now show that the control law given by Eq. (3.2) can be interpreted as model-following. It follows from Eqs. (3.11) and (3.12) that

$$\frac{T}{R} = \frac{A_o B'_m}{R} = \frac{(AR' + B^- S)B'_m}{A_m R} = \frac{AB_m}{BA_m} + \frac{SB_m}{RA_m}$$

The control law of Eq. (3.2) can be written as

$$\begin{aligned} u &= \frac{T}{R}u_c - \frac{S}{R}y = \frac{AB_m}{BA_m}u_c + \frac{SB_m}{RA_m}u_c - \frac{S}{R}y \\ &= \frac{AB_m}{BA_m}u_c - \frac{S}{R}(y - y_m) \end{aligned}$$

A block diagram representation of this controller is given in Fig. 3.3. The figure shows that the controller can be interpreted as a combination of a feedforward controller and a feedback controller. The feedforward controller attempts to cancel the plant dynamics and replace it with the response of the model B_m/A_m. Also the feedback attempts to make the output follow this model. It is thus clear that the control law (3.2) can indeed be interpreted as a model-following algorithm.

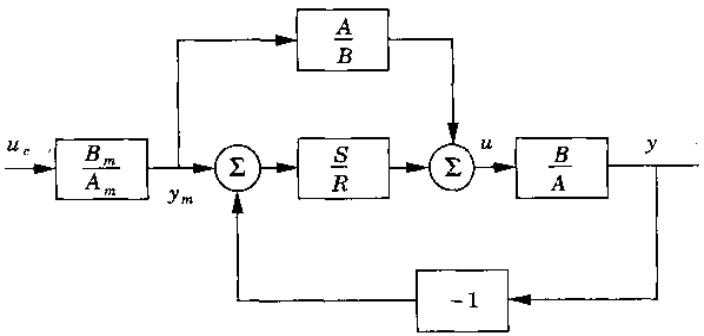

Figure 3.3 Alternative representation of model-following based on output feedback.

Notice that Fig. 3.3 is useful for the purpose of giving insight but that the controller cannot be implemented as shown in the figure because the inverse process model A/B is generally not realizable. Furthermore, A/B will be unstable if the system is non-minimum phase. However, the cascade combination of the reference model and the inverse process model is realizable if the model-following problem is well posed, that is, if Eqs. (3.13) are satisfied. Notice that the reference model and the inverse process model can be nonlinear without causing any stability problems because they appear only as part of a feedforward compensator.

Summary

In this section we have presented a straightforward design procedure that is relatively easy to use. The key problem in applying pole placement is to choose the desired closed-loop poles and the desired response to command signals. The choice is easy for low-order systems, but it may be difficult for systems of high order when many poles must be specified. Bad choices may result in a closed-loop system with poor sensitivity. In later chapters we will discuss this problem in more detail.

In the sampled-data case the sampling interval is a crucial design parameter. It is important to choose the sampling interval in relation to the desired closed-loop poles.

3.3 INDIRECT SELF-TUNING REGULATORS

Methods for estimating parameters of the model given by Eq. (3.1) were presented in Chapter 2. These methods will now be combined with the design method of Section 3.2 to obtain a simple self-tuning regulator. For simplicity it will be assumed that the disturbance v in Eq. (3.1) is zero.

Estimation

Several of the recursive estimation methods outlined in Chapter 2 can be used to estimate the coefficients of the A and B polynomials. The equations for recursive least-squares estimation will be used. The process model (3.1) can be written explicitly as

$$\begin{aligned} y(t) = &- a_1 y(t-1) - a_2 y(t-2) - \ldots - a_n y(t-n) \\ &+ b_0 u(t-d_0) + \ldots + b_m u(t-d_0-m) \end{aligned}$$

Notice that the degree of the system is $\max(n, d_0 + m)$. The model is linear in the parameters and can written as

$$y(t) = \varphi^T(t-1)\theta$$

where

$$\theta^T = \begin{pmatrix} a_1 & a_2 & \dots & a_n & b_0 & \dots & b_m \end{pmatrix}$$
$$\varphi^T(t-1) = \begin{pmatrix} -y(t-1) & \dots & -y(t-n) & u(t-d_0) & \dots & u(t-d_0-m) \end{pmatrix}$$

The least-squares estimator with exponential forgetting is given by

$$\begin{aligned}
\hat{\theta}(t) &= \hat{\theta}(t-1) + K(t)\varepsilon(t) \\
\varepsilon(t) &= y(t) - \varphi^T(t-1)\hat{\theta}(t-1) \\
K(t) &= P(t-1)\varphi(t-1)\left(\lambda + \varphi^T(t-1)P(t-1)\varphi(t-1)\right)^{-1} \\
P(t) &= \left(I - K(t)\varphi^T(t-1)\right)P(t-1)/\lambda
\end{aligned} \tag{3.22}$$

(Compare with Eq. (2.21).) If the input signal to the process is sufficiently exciting and the structure of the estimated model is compatible with the process, the estimates will converge to their true values. It takes $\max(n, m + d_0)$ sampling periods before the regression vector is defined. In the deterministic case it takes at least $n + m + 1$ additional sampling periods to determine the $n + m + 1$ parameters of the model, assuming that the process input is persistently exciting. It thus takes at least

$$N = n + m + 1 + \max(n, m + d_0) \tag{3.23}$$

sampling periods for the algorithm to converge. With recursive least squares initialized with a large P-matrix it may take a few more steps. Since the process input is generated by feedback, it may be difficult to assert that it is persistently exciting. Presence of process noise may also make convergence much slower. Convergence issues will be discussed further in Chapter 6.

An Indirect Self-Tuner

Combining the recursive least squares (RLS) estimator given by Eqs. (3.22) with the minimum-degree pole placement method (MDPP) for controller design given by Algorithm 3.1, we obtain the following self-tuning regulator.

ALGORITHM 3.2 Indirect self-tuning regulator using RLS and MDPP

Data: Given specifications in the form of a desired closed-loop pulse transfer operator B_m/A_m and a desired observer polynomial A_o.

Step 1: Estimate the coefficients of the polynomials A and B in Eq. (3.1) using the recursive least-squares method given by Eqs. (3.22).

Step 2: Apply the minimum-degree pole placement method given by Algorithm 3.1 where polynomials A and B are the estimates obtained in Step 1. The polynomials R, S, and T of the control law are then obtained.

Step 3: Calculate the control variable from Eq. (3.2), that is,

$$Ru(t) = Tu_c(t) - Sy(t)$$

Repeat Steps 1, 2, and 3 at each sampling period. Notice that there are some variations in the algorithm depending on the cancellations of the process zeros. Also notice that it is not necessary to perform Steps 1 and 2 at each sampling interval. □

Examples

The properties of indirect self-tuning regulators are illustrated by the following two examples.

EXAMPLE 3.4 Indirect self-tuner with cancellation of process zero

Let the process be the same as in Example 3.1 and assume that the process zero is canceled. The specifications are the same as in Example 3.1, that is, to obtain a closed-loop characteristic polynomial A_m. The parameters of the model

$$y(t) + a_1 y(t-1) + a_2 y(t-2) = b_0 u(t-1) + b_1 u(t-2)$$

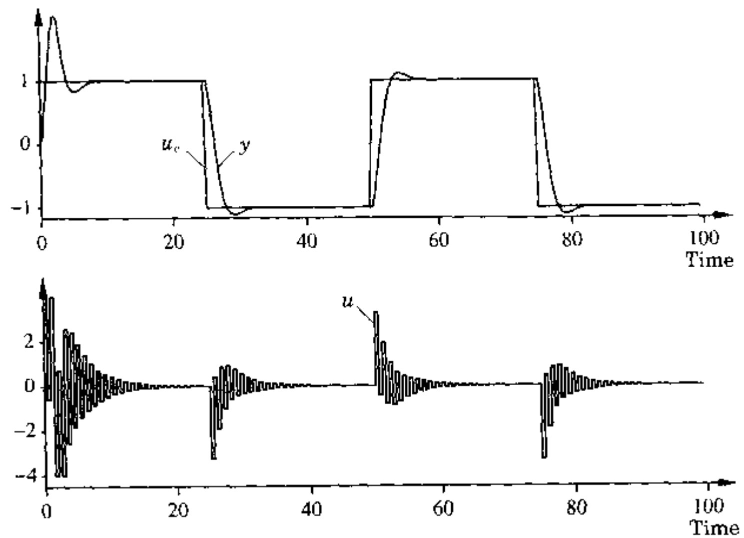

Figure 3.4 Output and input in using an indirect self-tuning regulator to control the system in Example 3.1. Notice the "ringing" in the control signal due to cancellation of the zero at −0.84.

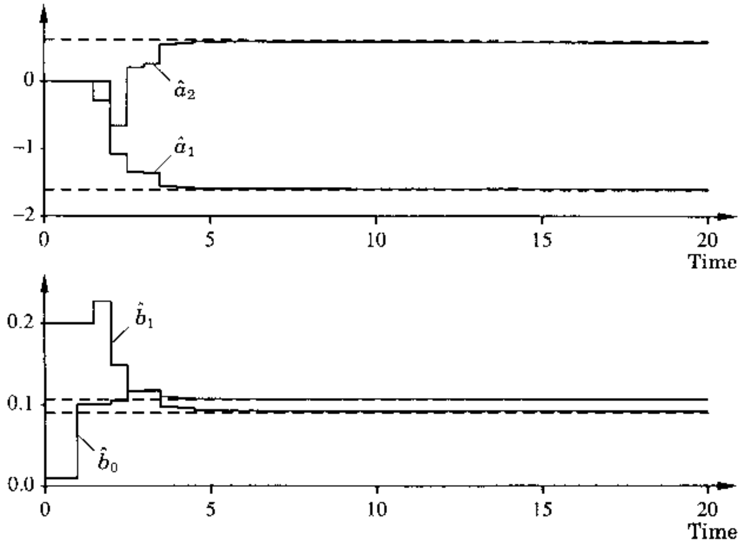

Figure 3.5 Parameter estimates corresponding to the simulation in Fig. 3.4. The true parameters are shown by dashed lines.

which has the same structure as Eq. (3.17), are estimated by using the least-squares algorithm. Algorithm 3.2 is used for the self-tuning regulator. The calculations, which were done in Example 3.1, give the control law

$$u(t) + r_1 u(t-1) = t_0 u_c(t) - s_0 y(t) - s_1 y(t-1)$$

The controller parameters were expressed as functions of the model parameters and the specifications. Figure 3.4 shows the process output and the control signal in a simulation of the process with the self-tuner when the command signal is a square wave. The output converges to the model output after an initial transient. The control signal has a severe oscillation ("ringing") with a period of two sampling periods. This is due to the cancellation of the process zero at $z = -b_1/b_0 = -0.84$. This oscillation is a consequence of a bad choice of the underlying design methodology. The initial transient depends critically on the initial values of the estimator. In this particular case these values were $\hat{a}_1(0) = \hat{a}_2(0) = 0$, $\hat{b}_0(0) = 0.01$, and $\hat{b}_1(0) = 0.2$. Notice that it is necessary that $\hat{b}_0 \neq 0$. (Compare with Example 3.1.) The initial covariance matrix was diagonal with $P(1,1) = P(2,2) = 100$ and $P(3,3) = P(4,4) = 1$. The reason for using different values for parameters $\hat{a}_i$ and $\hat{b}_i$ is that these parameters differ by an order of magnitude.

The parameter estimates are shown in Fig. 3.5. The behavior of the estimates depends critically on the initial values of the estimator. Notice that the

estimates converge quickly. They are close to their correct values already at time $t = 5$. The estimates obtained at time $t = 100$ are

$$\hat{a}_1(100) = -1.60 \quad (-1.6065) \qquad \hat{b}_0(100) = 0.107 \quad (0.1065)$$
$$\hat{a}_2(100) = 0.60 \quad (0.6065) \qquad \hat{b}_1(100) = 0.092 \quad (0.0902)$$

These values are quite close to the true values, which are given in parentheses. The controller parameters obtained at time $t = 100$ are

$$r_1(100) = 0.85 \quad (0.8467) \qquad t_0(100) = 1.65 \quad (1.6531)$$
$$s_0(100) = 2.64 \quad (2.6852) \qquad s_1(100) = -0.99 \quad (-1.0321)$$

□

The system in Example 3.4 behaves quite well, apart from the "ringing" control signal. This can be avoided by using a design in which the process zero is not canceled. The consequences of this are illustrated in the next example.

EXAMPLE 3.5 Indirect self-tuner without cancellation of process zero

Consider the same process as in Example 3.4, but use a control design in which there is no cancellation of the process zero. The parameters are estimated in the same way as in Example 3.4, but the control law is now computed as in Example 3.2. Polynomial A_o is of first order. As in the previous examples the initial transient depends critically on the initial state of the

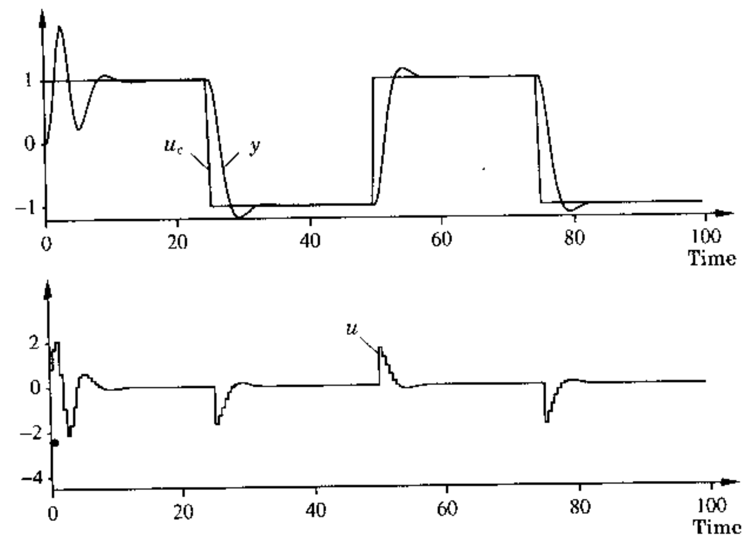

Figure 3.6 Same as in Fig. 3.4 but without cancellation of the process zero.

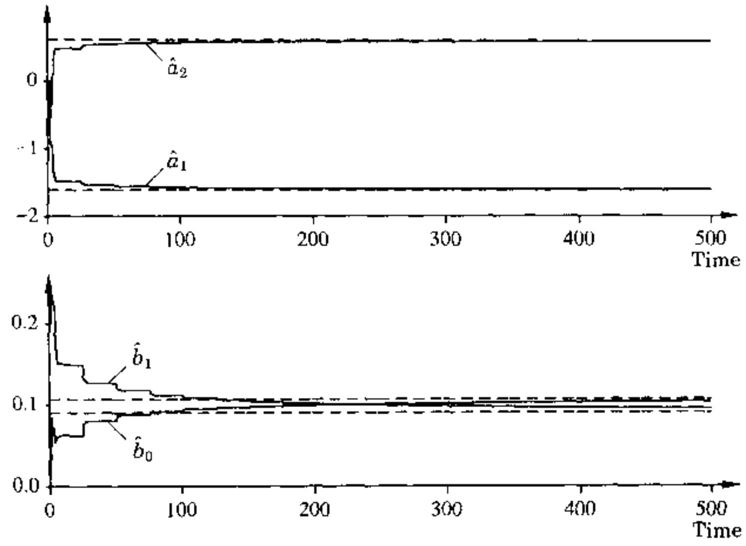

Figure 3.7 Parameter estimates corresponding to the simulation in Fig. 3.6. The true parameter values are indicated by dashed lines.

recursive estimator. For the design calculation it must be required that initial values are chosen so that polynomials A and B do not have a common factor. In this case the initial estimates were chosen to be $\hat{a}_1(0) = \hat{a}_2(0) = 0$, $\hat{b}_0(0) = 0.01$, and $\hat{b}_1(0) = 0.2$. The P-matrix was initialized as a diagonal matrix with $P(1,1) = P(2,2) = 100$ and $P(3,3) = P(4,4) = 1$ as in Example 3.4. Figure 3.6 shows results of a simulation of the direct algorithm with $a_o = 0$. Notice that the behavior of the process output is quite similar to that in Fig. 3.4 but that there is no "ringing" in the control signal. The parameter estimates are shown in Fig. 3.7. The values obtained at time $t = 100$ are

$$\hat{a}_1(100) = -1.57 \quad (-1.6065) \qquad \hat{b}_0(100) = 0.092 \quad (0.1065)$$
$$\hat{a}_2(100) = 0.57 \quad (0.6065) \qquad \hat{b}_1(100) = 0.112 \quad (0.0902)$$

The true values are given in parentheses. The controller parameters at time $t = 100$ are

$$r_1(100) = 0.114 \quad (0.1111) \qquad t_0(100) = 0.86 \quad (0.8951)$$
$$s_0(100) = 1.44 \quad (1.6422) \qquad s_1(100) = -0.58 \quad (-0.7471)$$

A comparison of Fig. 3.5 and Fig. 3.7 shows that it takes significantly longer for the estimates to converge when no zero is canceled. The reason for this is that the excitation is not as good as when there was "ringing" in the control signal.

There is very little excitation of the system in the periods when the output and the control signals are constant. This explains the steplike behavior of the estimates.

It may seem surprising that the controller already gives the correct steady-state value at time $t = 20$ when the parameter estimates differ so much from their correct values. The controller parameters are

$$r_1(20) = 0.090 \quad (0.1111) \qquad t_0(20) = 0.83 \quad (0.8951)$$
$$s_0(20) = 1.13 \quad (1.6422) \qquad s_1(20) = -0.29 \quad (-0.7471)$$

Since the process has integral action, we have $A(1) = 0$. It then follows from Eq. (3.3) that the static gain from command signal to output is

$$\frac{B(1)T(1)}{A(1)R(1) + B(1)S(1)} = \frac{T(1)}{S(1)}$$

To obtain the correct steady-state value, it is thus sufficient that the controller parameters are such that $S(1) = T(1)$, which in the special case is the same as $t_0 = s_0 + s_1$. When no poles are canceled, it follows from Eq. (3.12) that

$$T(1) = A_o(1)B_m'(1) = A_o(1)\frac{A_m(1)}{\hat{B}(1)}$$

where $\hat{B}$ is the estimated B polynomial. Hence

$$\frac{T(1)}{S(1)} = \frac{A_o(1)A_m(1)}{\hat{B}(1)S(1)} = 1$$

where the last equality follows from Eq. (3.11). Notice that we have $A(1) = 0$. We thus obtain the rather surprising conclusion that the adaptive controller in this case will automatically have parameters such that there will be no steady-state error. □

These examples indicate that the indirect self-tuning algorithm behaves as can be expected and that the estimate of convergence time given by Eq. (3.23) is reasonable. The examples also show the importance of using a good underlying control design. With model-following design it is recommended that cancellation of process zeros is avoided.

Summary

The indirect self-tuning regulator based on model-following given by Algorithm 3.1 is a straightforward application of the idea of self-tuning. The adaptive controller has states that correspond to the parameter estimate φ, the covariance matrix P, the regression vector φ, and the states required for the implementation of the control law. The controller in Example 3.4 has 20 state variables; updating of the covariance matrix P alone requires ten states. The complete codes for the controllers in the examples are listed in the problems at the end of this chapter.

The algorithm can be generalized in many different ways by choosing other recursive estimation methods and other control design techniques. The idea is easy to apply. A detailed discussion of practical implementation is given in Chapter 11.

3.4 CONTINUOUS-TIME SELF-TUNERS

Continuous-time self-tuners can be derived in the same way as discrete-time self-tuners. To show this, consider a system that can be described by the model (3.1) with $v = 0$, that is,

$$A(p)y(t) = B(p)u(t)$$

where $A(p)$ and $B(p)$ are polynomials in the differential operator, $p = d/dt$:

$$A(p) = p^n + a_1 p^{n-1} + \cdots + a_n$$
$$B(p) = b_1 p^{n-1} + \cdots + b_n$$

A self-tuning regulator can be obtained by applying Algorithm 3.1. The only complication is that we now must apply recursive least-squares estimation to the continuous-time model. This was discussed in Section 2.3. Let us recall the key idea. Since it is undesirable to take derivatives, a stable filtering transfer function H_f with a pole excess of n or more is introduced.

If we introduce the filtered signals

$$y_f(t) = H_f y(t) \qquad u_f(t) = H_f u(t)$$

the model (3.1) can be written as

$$p^n y_f(t) = \varphi^T(t)\theta$$

where

$$\varphi(t) = \begin{pmatrix} -p^{n-1}y_f & \cdots & -y_f & p^{n-1}u_f & \cdots & u_f \end{pmatrix}^T$$
$$\theta = \begin{pmatrix} a_1 & \cdots & a_n & b_1 & \cdots & b_n \end{pmatrix}^T$$

By using least squares with exponential forgetting the parameter estimate is then obtained from Theorem 2.5:

$$\frac{d\hat{\theta}(t)}{dt} = P(t)\varphi(t)\left(p^n y_f(t) - \varphi^T(t)\hat{\theta}(t)\right)$$
$$\frac{dP(t)}{dt} = \alpha P(t) - P(t)\varphi(t)\varphi^T(t)P(t)$$

We illustrate the procedure given by Algorithm 3.1 by an example.

EXAMPLE 3.6 Continuous-time self-tuner

Consider the system in Example 3.3, in which the process has the transfer function

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

with $a = 1$ and $b = 1$. Notice that the process has only two unknown parameters, a and b. The regressor filters in the estimator are chosen to be

$$H_f(s) = \frac{1}{A_m(s)}$$

Furthermore, we use an estimator without forgetting, that is, $\alpha = 0$. Assume that it is desired to obtain a closed-loop system with the transfer function

$$G_m(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

The observer polynomial is chosen to be $A_o(s) = s + a_o$ with $a_o = 2$. The specifications are the same as in Example 3.4, that is, $\zeta = 0.7$ and $\omega = 1$. In Example 3.3 we solved the design problem when the parameters a and b are known. We found that the controller has the form

$$u(t) = -\frac{s_0 p + s_1}{p + r_1}\, y(t) + \frac{t_0(p + a_o)}{p + r_1}\, u_c(t)$$

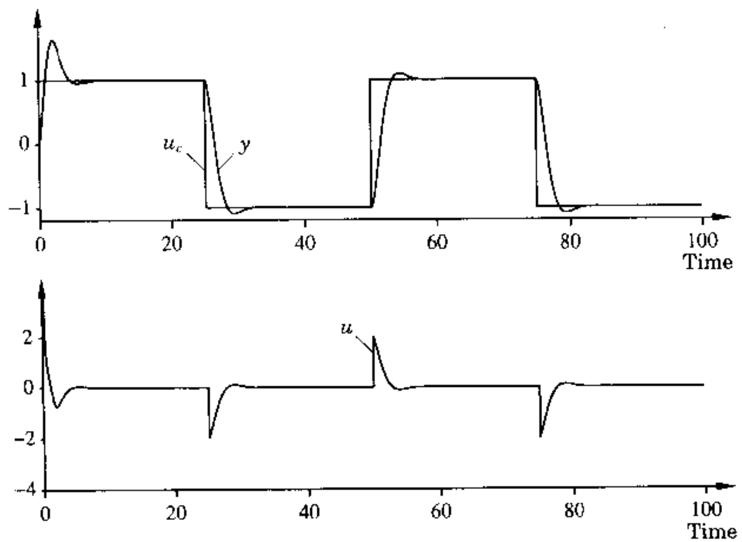

Figure 3.8 Output and input when using a continuous-time indirect self-tuning regulator to control the process in Example 3.6.

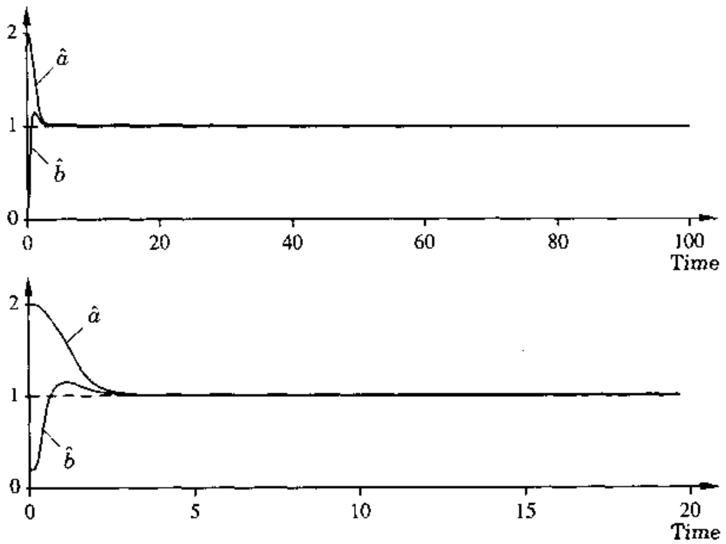

Figure 3.9 Continuous-time parameter estimates corresponding to the simulation in Fig. 3.8. The lower part shows the estimates in an extended time scale.

where the controller parameters are given by

$$
\begin{aligned}
r_1 &= 2\zeta\omega + a_o - a \\
s_0 &= \frac{a_o 2\zeta\omega + \omega^2 - a r_1}{b} \\
s_1 &= \frac{\omega^2 a_o}{b} \\
t_0 &= \frac{\omega^2}{b}
\end{aligned}
$$

Figure 3.8 shows the process output and the control signal in a simulation. The initial transient depends critically on the initial values of the estimator. In this case we have chosen $\hat{a}(0) = 2$ and $\hat{b}(0) = 0.2$. The initial covariance is diagonal with $P(1,1) = P(2,2) = 100$. The parameter estimates are shown in Fig. 3.9. The estimates obtained at $t = 100$ are

$$\hat{a}(100) = 1.004 \quad (1.0000) \qquad \hat{b}(100) = 1.001 \quad (1.0000)$$

where the true values are given in parentheses. Notice that only two parameters are estimated in this case, whereas four parameters were estimated in Examples 3.4 and 3.5. □

3.5 DIRECT SELF-TUNING REGULATORS

The design calculations in the indirect self-tuners may be time-consuming and poorly conditioned for some parameter values. It is possible to derive other algorithms in which the design calculations are simplified or even eliminated. The idea is to use the design equations to reparameterize the model in terms of the parameters of the controller. This reparameterization is also the key to understanding the relations between model-reference adaptive systems and self-tuning regulators.

Consider a process described by Eq. (3.1) with $v = 0$, that is,

$$Ay(t) = Bu(t)$$

and let the desired response be given by Eq. (3.5):

$$A_m y_m(t) = B_m u_c(t)$$

The process model will now be reparameterized in terms of the controller parameters. To do this, consider the Diophantine equation (3.11),

$$A_o A_m = AR' + B^- S$$

as an operator identity, and let it operate on $y(t)$. This gives

$$A_o A_m y(t) = R' A y(t) + B^- S y(t) = R' B u(t) + B^- S y(t)$$

It follows from Eq. (3.10) that

$$R'B = R'B^+B^- = RB^-$$

Hence

$$A_o A_m y(t) = B^- \left(Ru(t) + Sy(t)\right) \tag{3.24}$$

Notice that this equation can be considered a process model that is parameterized in the coefficients of the polynomials B^-, R, and S. If the parameters in the model given by Eq. (3.24) are estimated, the control law is thus obtained directly without any design calculations. Notice that the model Eq. (3.24) is nonlinear in the parameters because the right-hand side is multiplied by B^-. The difficulties caused by this can be avoided in the special case of minimum-phase systems in which $B^- = b_0$, which is a constant.

Minimum-Phase Systems

If the process dynamics is minimum phase, we have $\deg A_o = \deg A - \deg B - 1$, B^- is simply a constant, and Eq. (3.24) becomes

$$A_m A_o y(t) = b_0 \left(Ru(t) + Sy(t)\right) = \bar{R}u(t) + \bar{S}y(t) \tag{3.25}$$

where R is monic, $\tilde{R} = b_0 R$, and $\tilde{S} = b_0 S$. Since R and $\tilde{R}$ differ only by R being monic, we will *not* use a separate notation in the following discussion. When it is necessary, we will simply note whether or not R is monic.

When all process zeros are canceled, it is also natural to choose specifications so that

$$B_m = q^{d_0} A_m(1)$$

where $d_0 = \deg A - \deg B$. This gives response with minimal delay and unit static gain.

By introducing the parameter vector

$$\theta = \begin{pmatrix} r_0 & \dots & r_\ell & s_0 & \dots & s_\ell \end{pmatrix}$$

and the regression vector

$$\varphi(t) = \begin{pmatrix} u(t) & \dots & u(t-\ell) & y(t) & \dots & y(t-\ell) \end{pmatrix}$$

the model given by Eq. (3.25) can be written as

$$\eta(t) = A_o^*(q^{-1})\, A_m^*(q^{-1})\, y(t) = \varphi^T(t-d_0)\theta \tag{3.26}$$

Since $\eta(t)$ can be computed from $y(t)$, it can be regarded as an auxiliary output, and a recursive estimate of the parameters can now be obtained as described in Chapter 2.

This estimation method works very well if there is little noise, but the operation $A_o^*(q^{-1})A_m^*(q^{-1})y(t)$ may amplify noise significantly. The following method can be used to overcome this. Rewrite Eq. (3.25) as

$$y(t) = \frac{1}{A_o A_m}\left(Ru(t) + Sy(t)\right) = R^* u_f(t-d_0) + S^* y_f(t-d_0) \tag{3.27}$$

where

$$\begin{aligned} u_f(t) &= \frac{1}{A_o^*(q^{-1})A_m^*(q^{-1})}\, u(t) \\ y_f(t) &= \frac{1}{A_o^*(q^{-1})A_m^*(q^{-1})}\, y(t) \end{aligned} \tag{3.28}$$

and $d_0 = \deg A - \deg B$. We have further assumed that $\deg R = \deg S = \deg(A_o A_m) - d_0 = \ell$. Equation (3.27) can be used for least-squares estimation. If we introduce

$$\theta = \begin{pmatrix} r_0 & \dots & r_\ell & s_0 & \dots & s_\ell \end{pmatrix}$$

and

$$\varphi(t) = \begin{pmatrix} u_f(t) & \dots & u_f(t-\ell) & y_f(t) & \dots & y_f(t-\ell) \end{pmatrix}$$

it can be written as

$$y(t) = \varphi^T(t-d_0)\theta$$

The estimates are then obtained recursively from Eqs. (3.22). The following adaptive control algorithm is then obtained.

ALGORITHM 3.3 Simple direct self-tuner

Data: Given specifications in terms of A_m, B_m, and A_o and the relative degree d_0 of the system.

Step 1: Estimate the coefficients of the polynomials R and S in the model (3.27), that is,

$$y(t) = R^* u_f(t-d_0) + S^* y_f(t-d_0)$$

by recursive least squares, Eqs. (3.22).

Step 2: Compute the control signal from

$$R^* u(t) = T^* u_c(t) - S^* y(t)$$

where R and S are obtained from the estimates in Step 1 and

$$T^* = A_o^* A_m(1) \tag{3.29}$$

with $\deg A_o = d_0 - 1$. Repeat Steps 1 and 2 at each sampling period. □

Equation (3.29) is obtained from the observation that the closed-loop transfer operator from command signal u_c to process output is

$$\frac{TB}{AR + BS} = \frac{Tb_0 B^+}{b_0 A_o A_m B^+} = \frac{T}{A_o A_m}$$

Requiring that this be equal to $q^{d_0} A_m(1)/A_m$ gives Eq. (3.29).

Remark 1. A comparison with Algorithm 3.2 shows that the step corresponding to control design is missing in Algorithm 3.3. This motivates the name "direct algorithm."

Remark 2. Notice that it is necessary to know the relative degree d_0 of the plant *a priori*.

Remark 3. The polynomials R and S contain the factor b_0. Notice that the polynomial R is not monic and that the parameter r_0 must be different from zero. Otherwise, the control law given by Eq. (3.2) is not causal. Since d_0 is the relative degree of the plant, the true value of $r_0 = b_0$ is different from zero. Any consistent estimate of the parameter will thus be different from zero. The estimate obtained for finite time may, however, be zero. In practice it is therefore essential to take some precautions.

Remark 4. Notice that the assumption $B^- = b_0$ implies that all process zeros are canceled. This is the reason why the algorithm requires the plant to be minimum phase.

Examples

Properties of direct self-tuners will now be illustrated by some examples.

EXAMPLE 3.7 Direct self-tuner with $d_0 = 1$

Consider the system in Example 3.1. Since $\deg A = 2$ and $\deg B = 1$, we have $\deg A_m = 2$ and $\deg A_o = 0$. Hence $A_o = 1$, and we will choose $B_m = qA_m(1)$. Equation (3.29) in Algorithm 3.3 then gives $T = qA_m(1)$. The controller structure is given by $\deg R = \deg S = \deg T = \deg A - 1 = 1$. The model given by Eq. (3.27) therefore becomes

$$y(t) = r_0 u_f(t-1) + r_1 u_f(t-2) + s_0 y_f(t-1) + s_1 y_f(t-2) \qquad (3.30)$$

where

$$u_f(t) + a_{m1} u_f(t-1) + a_{m1} u_f(t-2) = u(t)$$

$$y_f(t) + a_{m1} y_f(t-1) + a_{m1} y_f(t-2) = y(t)$$

It is now straightforward to obtain a direct self-tuner by applying Algorithm 3.3. The parameters of the model given by Eq. (3.30) are thus estimated, and the control signal is then computed from

$$\hat{r}_0 u(t) + \hat{r}_1 u(t-1) = \hat{t}_0 u_c(t) - \hat{s}_0 y(t) - \hat{s}_1 y(t-1)$$

where $\hat{r}_0$, $\hat{r}_1$, $\hat{s}_0$, and $\hat{s}_1$ are the estimates obtained and $\hat{t}_0$ is given by Eq. (3.29), that is,

$$\hat{t}_0 = 1 + a_{m1} + a_{m2}$$

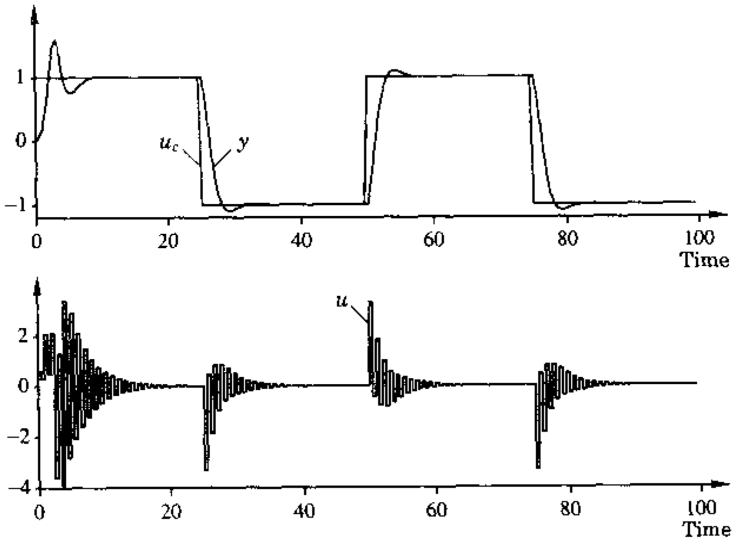

Figure 3.10 Command signal u_c, process output y, and control signal u when the process given by Eq. (3.16) is controlled by using a direct self-tuner with $d_0 = 1$. Compare with Fig. 3.4.

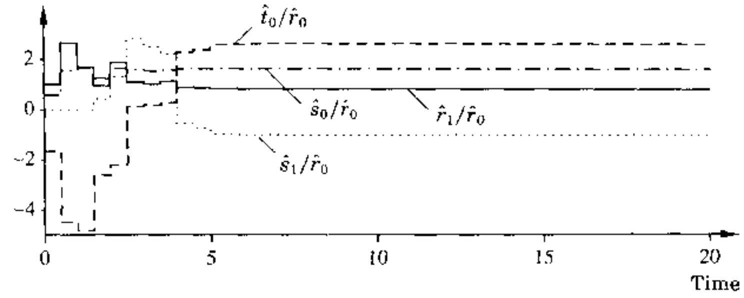

Figure 3.11 Parameter estimates corresponding to the simulation shown in Fig. 3.10: $\hat{r}_1/\hat{r}_0$ (solid line), $\hat{t}_0/\hat{r}_0$ (dashed line), $\hat{s}_0/\hat{r}_0$ (dash-dot line), $\hat{s}_1/\hat{r}_0$ (dotted line).

Notice that the estimate of r_0 must be different from zero for the controller to be causal.

Figure 3.10 shows the process inputs and outputs in a simulation of the direct algorithm, and Fig. 3.11 shows the parameter estimates. The initial transient depends strongly on the initial conditions. At $t = 100$ the controller parameters are

$$\frac{\hat{r}_1(100)}{\hat{r}_0(100)} = 0.850 \quad (0.8467) \qquad \frac{\hat{t}_0(100)}{\hat{r}_0(100)} = 1.65 \quad (1.6531)$$

$$\frac{\hat{s}_0(100)}{\hat{r}_0(100)} = 2.68 \quad (2.6852) \qquad \frac{\hat{s}_1(100)}{\hat{r}_0(100)} = -1.03 \quad (-1.0321)$$

The controller parameters are divided by $\hat{r}_0$ to make a direct comparison with Examples 3.1 and 3.3. The correct values are given in parentheses. A comparison of Fig. 3.4 and Fig. 3.10 shows that the direct and indirect algorithms have very similar behavior. The limiting control law is the same in both cases. There is "ringing" in the control signal because of the cancellation of the process zero. □

In a practical case the time delay and the order of the process that we would like to control are not known. It is therefore natural to consider these variables as design parameters that are chosen by the user. The parameter d_0 is of particular importance for a direct algorithm. In the next example we show that "ringing" can be avoided simply by increasing the value of d_0.

EXAMPLE 3.8 Direct self-tuner with $d_0 = 2$

In the derivation of the direct algorithm the parameter d_0 was the pole excess of the plant. Assume for a moment that we do not know the value of d_0 and that we treat it as a design parameter instead. Figure 3.12 shows a simulation of the direct algorithm used in Example 3.7 but with $d_0 = 2$ instead of $d_0 = 1$. All

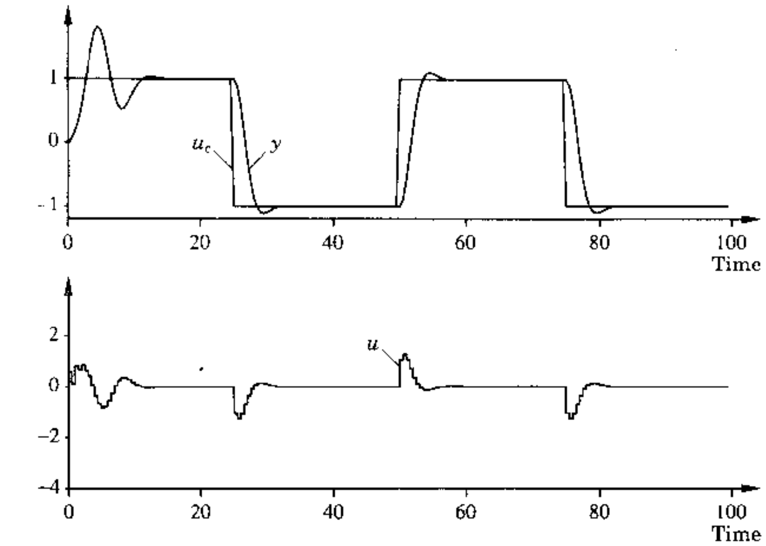

Figure 3.12 Command signal u_c, process output y, and control signal u when the process described by Eq. (3.16) is controlled with a direct self-tuner with $d_0 = 2$.

the other parameters are the same. Notice that the behavior of the system is quite reasonable without any "ringing" in the control signal. Figure 3.13 shows the parameter estimates. The estimates obtained at time $t = 100$ correspond to the controller parameters

$$\frac{\hat{r}_1(100)}{\hat{r}_0(100)} = -0.337 \qquad \frac{\hat{s}_0(100)}{\hat{r}_0(100)} = 1.20 \qquad \frac{\hat{s}_1(100)}{\hat{r}_0(100)} = -0.67 \qquad \frac{\hat{t}_0(100)}{\hat{r}_0(100)} = 0.52$$

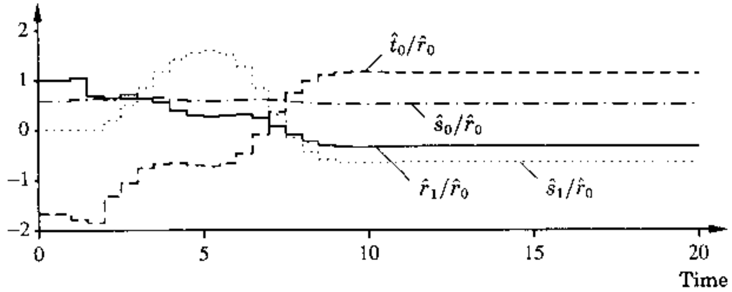

Figure 3.13 Parameter estimates corresponding to Fig. 3.12: $\hat{r}_1/\hat{r}_0$ (solid line), $\hat{t}_0/\hat{r}_0$ (dashed line), $\hat{s}_0/\hat{r}_0$ (dash-dot line), $\hat{s}_1/\hat{r}_0$ (dotted line).

We thus find the interesting and surprising result that cancellation of the process zero can be avoided by increasing the parameter d_0. This observation will be explained later when we will be analyzing the algorithms. □

Feedforward Control

A nice feature of the direct self-tuner is that it is easy to include feedforward. Let v be a disturbance that can be measured. By estimating parameters in the model

$$y(t) = \frac{1}{A_o A_m}\left(Ru(t) + Sy(t) - Uv(t)\right) \tag{3.31}$$

and using the control law

$$Ru(t) = Tu_c(t) - Sy(t) - Uv(t)$$

we obtain a self-tuning controller that combines feedback and feedforward. The term Tu_c in the control law can also be viewed as a feedforward term.

In Algorithm 3.3, polynomials R and S are estimated and the polynomial T is computed. This means that the different terms of the control law are treated differently. It is possible to obtain an algorithm in which all coefficients of the control law are estimated by treating Tu_c as a feedforward term that is adapted. To do this, we first notice that the desired response is given by

$$y_m(t) = \frac{B_m}{A_m} u_c(t) = \frac{T}{A_o A_m} u_c(t)$$

It follows from Eq. (3.27) that error $e(t) = y(t) - y_m(t)$ is given by

$$\begin{aligned} e(t) &= \frac{1}{A_o A_m}\Big(Ru(t) + Sy(t) - Tu_c(t)\Big) \\ &= R^* u_f(t - d_0) + S^* y_f(t - d_0) - T^* u_{cf}(t - d_0) \end{aligned} \tag{3.32}$$

where u_f, y_f, and u_{cf} are the filtered signals defined by Eqs. (3.28) and

$$u_{cf}(t) = \frac{1}{A_o^*(q^{-1})A_m^*(q^{-1})} u_c(t)$$

Furthermore, $\deg T = \deg R = \deg S = \deg(A_o A_m) - d_0$ and $\deg A_m - \deg B_m = d_0$. An algorithm that is analogous to Algorithm 3.3, in which the parameters of the feedforward polynomial T are also estimated is now easily obtained by estimating the parameters in Eq. (3.32).

Non-minimum-Phase (NMP) Systems

The case in which process zeros cannot be canceled will now be discussed. Consider the transformed process model Eq. (3.24), that is,

$$A_o A_m y(t) = B^-(Ru(t) + Sy(t))$$

where $\deg R = \deg S = \deg(A_o A_m) - \deg B^-$. If we introduce

$$\mathcal{R} = B^- R \quad \text{and} \quad \mathcal{S} = B^- S$$

the equation can be written as

$$y(t) = \frac{1}{A_o A_m}(\mathcal{R}u(t) + \mathcal{S}y(t)) = \mathcal{R}^* u_f(t - d_0) + \mathcal{S}^* y_f(t - d_0) \qquad (3.33)$$

where u_f and y_f are the filtered inputs and outputs given by Eqs. (3.28). Notice that the polynomial $\mathcal{R}$ is not monic. The polynomials $\mathcal{R}$ and $\mathcal{S}$ have a common factor, which represents poorly damped zeros. This factor should be canceled before the control law is calculated. The following direct adaptive control algorithm is then obtained.

ALGORITHM 3.4 Direct self-tuning regulator for NMP systems

Data: Given specifications in terms of A_m, B_m, and A_o and the relative degree d_0 of the system.

Step 1: Estimate the coefficients of the polynomials $\mathcal{R}$ and $\mathcal{S}$ in the model of Eq. (3.33) by recursive least squares.

Step 2: Cancel possible common factors in $\mathcal{R}$ and $\mathcal{S}$ to obtain R and S.

Step 3: Calculate the control signal from Eq. (3.2) where R and S are those obtained in Step 2 and T is given by Eq. (3.12).

Repeat Steps 1, 2, and 3 at each sampling period. □

This algorithm avoids the nonlinear estimation problem, but more parameters have to be estimated than when Eq. (3.24) is used because the parameters of the polynomial B^- are estimated twice. The estimation is straightforward, however, because the model is linear in the parameters. The Euclidean algorithm in Chapter 11 can be used in Step 2 to eliminate common factors of polynomials $\mathcal{R}$ and $\mathcal{S}$. This step is crucial because an unstable common factor may cause instabilities.

Calculation of polynomial T should be avoided. To do this, notice that

$$y_m = \frac{B^- B_m'}{A_m} u_c$$

The error $e = y - y_m$ can then be written as

$$\begin{aligned} e(t) &= \frac{B^-}{A_o A_m}(Ru(t) + Sy(t) - Tu_c(t)) \\ &= \mathcal{R}^* u_f(t - d_0) + \mathcal{S}^* y_f(t - d_0) - \mathcal{T}^* u_{cf}(t - d_0) \end{aligned} \qquad (3.34)$$

By basing parameter estimation on this equation, estimates of polynomials $\mathcal{R}$, $\mathcal{S}$, and $\mathcal{T}$ can be determined. Notice that to estimate coefficients of $\mathcal{T}$, it is necessary that the command signal be persistently exciting.

Mixed Direct and Indirect Algorithms

Another direct algorithm can be derived in the particular case in which no process zeros are canceled. In this case we have $B^- = B$, and the model Eq. (3.24) becomes

$$A_o A_m y(t) = B\big(Ru(t) + Sy(t)\big)$$

which can also be written as

$$y(t) = \frac{B}{A_o A_m}(Ru(t) + Sy(t)) = B^*\left(R^* u_f(t - d_0) + S^* y_f(t - d_0)\right) \qquad (3.35)$$

The following algorithm is a hybrid algorithm that combines features of direct and indirect schemes.

ALGORITHM 3.5 A hybrid self-tuner

Data: Given polynomials A_o and A_m.

Step 1: Estimate parameters of polynomials A and B in the model

$$Ay = Bu$$

Step 2: Estimate parameters of polynomials R and S in Eq. (3.35) where B is the estimate obtained in Step 1.

Step 3: Use the control law

$$Ru = Tu_c - Sy$$

where R and S are obtained from Step 2 and $T = t_0 A_o$ where

$$t_0 = \frac{A_m(1)}{B(1)}$$

□

Remark 1. Instead of being computed, polynomial T can also be estimated by replacing Step 2 by the following step:

Step 2': Estimate parameters of polynomials R and S and t_0 from the model

$$\begin{aligned} e(t) = y(t) - y_m(t) &= \frac{B}{A_o A_m}(Ru(t) + Sy(t) - t_0 A_o u_c(t)) \\ &= B^*\left(R^* u_f(t - d_0) + S^* y_f(t - d_0) - t_0 A_o^* u_{cf}(t - d_0)\right) \end{aligned} \qquad (3.36)$$

where B is the polynomial obtained in Step 1. It is then assumed that $\deg A_m = \deg A$.

Remark 2. Instead of the Diophantine equation being solved at each step, two process models are estimated. This implies that an additional iteration of the least-squares estimator has to be done at each sampling time.

3.6 DISTURBANCES WITH KNOWN CHARACTERISTICS

So far, we have concentrated on servo problems that are common in aerospace and mechatronics. In process control, regulation problems are more common. It is then important to consider attenuation of disturbances that act on the process. The disturbance may enter the process in many different ways. For simplicity we will assume that it enters at the process input as shown in Figure 3.2. This assumption is not very restrictive. If the disturbance is denoted by v, the system is then described by Eq. (3.1). We will first use an example to illustrate that load disturbances will cause problems.

EXAMPLE 3.9 Effect of Load Disturbances

Consider the system in Example 3.5, that is, an indirect self-tuning regulator with no zero cancellation. We will now make a simulation that is identical to the one shown in Fig. 3.6 except that the load disturbance will be $v(t) = 0.5$ for $t \geq 40$. A forgetting factor $\lambda = 0.98$ has also been introduced; otherwise, the conditions are identical to those in Example 3.5. The behavior of the system is shown in Fig. 3.14. Compare Fig. 3.14 with Fig. 3.6. Figure 3.14 shows that a load disturbance may be disastrous. It follows from the discussion in Example 3.5 that the correct steady-state value will always be reached provided that

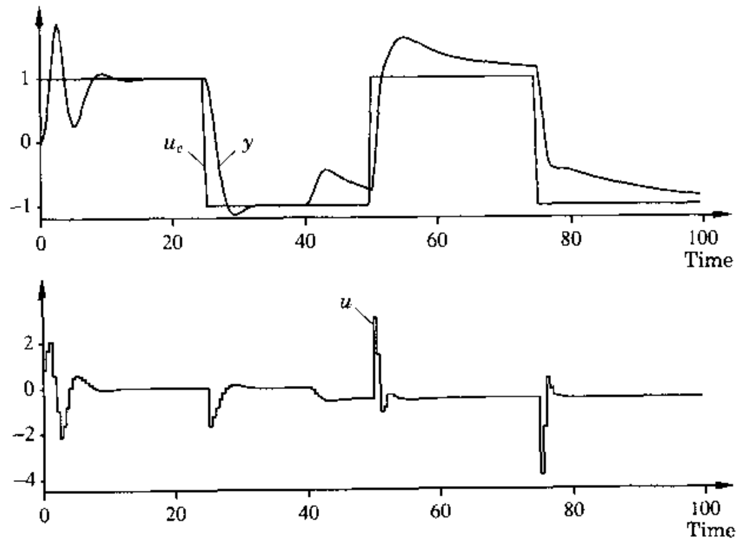

Figure 3.14 Output and control signal when for a system with an indirect self-tuner without zero canceling when there is a load disturbance in the form of a step at the process input at time $t = 40$.

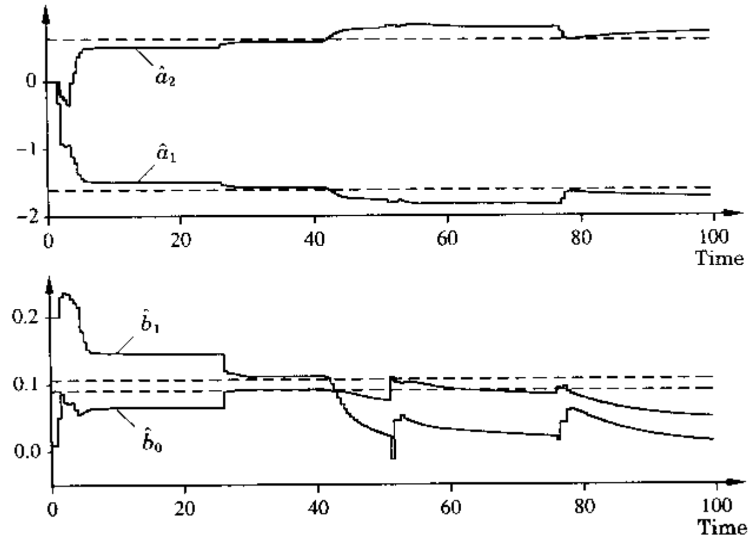

Figure 3.15 Parameter estimates corresponding to Fig. 3.14.

the steps are sufficiently long. Notice that the response is strongly asymmetric. The reason for this is that the controller parameters change rapidly when the control signal changes; see Fig. 3.15, which shows the parameter estimates. Rapid changes of the estimates in response to command signals indicates that the model structure is not correct. The parameter estimates also change significantly at the step in the load disturbance. When the command signal is constant, the parameters appear to settle at constant values that are far from the true parameters. □

There are many ways to deal with disturbances. The *internal model principle* is used in this section. An alternative is to estimate the disturbance and compensate for it in a feedforward fashion. An in-depth discussion of different methods and their advantages and disadvantages is found in Chapter 11.

A Modified Design Procedure

The pole placement procedure can be modified to take disturbances into account. In many cases the important disturbances have known characteristics. This can be captured by assuming that the disturbance v in the model (3.1) is generated by the dynamical system

$$A_d v = e \tag{3.37}$$

where e is a pulse, a set of widely spread pulses, white noise, or the equivalent continuous-time concepts. For example, a step disturbance is generated in discrete-time systems by

$$A_d(q) = q - 1$$

and in continuous-time systems by

$$A_d(p) = p$$

With the controller of Eq. (3.2) we find

$$\begin{aligned} y &= \frac{BT}{AR+BS}u_c + \frac{BR}{A_d(AR+BS)}e \\ u &= \frac{AT}{AR+BS}u_c - \frac{BS}{A_d(AR+BS)}e \end{aligned} \tag{3.38}$$

The closed-loop characteristic polynomial thus contains the disturbance dynamics as a factor. This polynomial typically has roots on the stability boundary or in the unstable region. It follows from Eqs. (3.38) that to maintain a finite output in case of these disturbances, A_d must be a factor of R. This would make y finite, but the controlled input u may be infinite. This is, of course, necessary to compensate for an infinite disturbance.

It has already been mentioned that the Diophantine equation has many solutions. Compare with Eqs. (3.14). If R^0 and S^0 are solutions to the Diophantine equation

$$AR^0 + BS^0 = A_c^0$$

it follows that

$$\begin{aligned} R &= XR^0 + YB \\ S &= XS^0 - YA \end{aligned} \tag{3.39}$$

satisfies the equation

$$AR + BS = XA_c^0$$

If a controller R^0 S^0 that gives the characteristic polynomial A_c^0 has been obtained, we can thus obtain a controller with characteristic polynomial XA_c^0 by using the controller (3.39). Suppose that we have designed a controller R^0 and S^0 and that we would like to have a new controller in which $R = R'A_d$. We then choose a stable polynomial X that represents the additional closed-loop poles, and we determine R' and Y such that

$$R = A_dR' = XR^0 + YB \tag{3.40}$$

The new controller is then given by Eqs. (3.39).

Integral Action

In the special case in which the disturbance is a constant, that is, $A_d = q - 1$, we have to add an additional closed-loop pole. Hence

$$X = q + x_0$$

and Eq. (3.40) becomes

$$(q-1)R' = (q + x_0)R^0 + y_0B$$

Putting $q = 1$ gives one equation to solve for y_0. Hence

$$y_0 = -\frac{(1+x_0)R^0(1)}{B(1)} \tag{3.41}$$

Inserting X and $Y = y_0$ into Eqs. (3.39) gives the new controller.

Modifications of the Estimator

Disturbances will change the relations between the inputs and the outputs in the model. Load disturbances such as steps will have a particularly bad effect on the low-frequency properties of the model. Several ways to deal with this problem are discussed in Section 11.5. One possibility is to include the disturbance in the model and estimate it; another, which we will use here, is to filter the signal so that the effect of the disturbance is not so large. In the model given by Eq. (3.1) the equation error is $B(q)v$. This could be a very large quantity if $B(1) \neq 0$ and v is a large step. If the disturbance v in Eq. (3.1) can be described by Eq. (3.37) we find that Eq. (3.1) can be written as

$$A_dAy(t) = A_dB(u(t) + v(t)) = A_dBy(t) + e(t)$$

Hence

$$Ay_f(t) = Bu_f(t) + e(t) \tag{3.42}$$

By introducing the filtered signals $y_f = A_dy$ and $u_f = A_du$ we thus obtain a model in which the equation error is e instead of v, where e is significantly smaller than v. For example, if v is a step and $A_d = q - 1$ as in Example 3.9, we find that e is zero except at the time where the step in v occurs.

The next example shows that the difficulties encountered in Example 3.9 can be avoided by using a self-tuner with a modified estimator and a modified control design.

EXAMPLE 3.10 Load disturbances: Modified estimator and controller

We now show that the difficulties found in Example 3.9 can be avoided by modifying the estimator and the controller. We first introduce a controller that has integral action by applying the design procedure that we have just

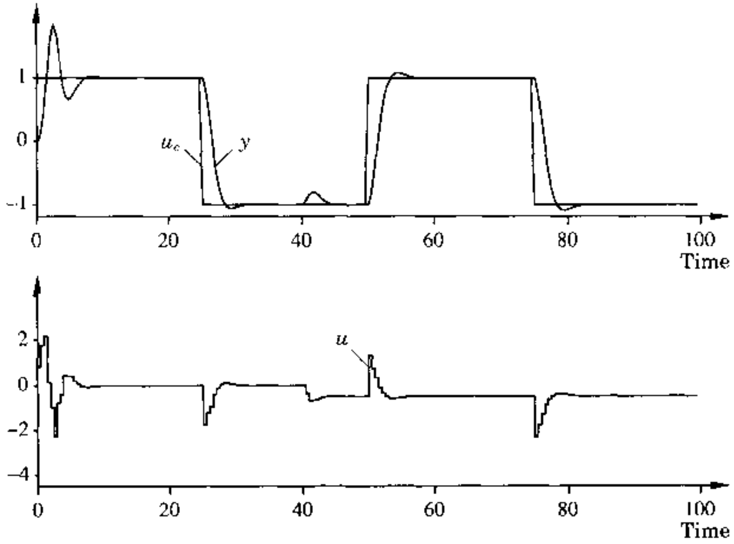

Figure 3.16 Output and control signal with an indirect self-tuner with integral action and a modified estimator.

described. To do this, we consider the same system as in Example 3.5 where the controller was defined by

$$R^0 = q + r_1 \qquad S^0 = s_0 q + s_1$$

The closed-loop characteristic polynomial A_c has degree three. To obtain a controller with integral action, the order of the closed-loop system is increased by introducing an extra closed-loop pole at $q = -x_0 = 0$. It then follows from Eq. (3.41) that

$$y_0 = -\frac{1 + r_1}{b_0 + b_1}$$

Hence $X = q$ and $Y = y_0$, and Eqs. (3.39) now give

$$R = q(q + r_1) + y_0(b_0 q + b_1) = (q - 1)(q - b_1 y_0)$$
$$S = q(s_0 q + s_1) - y_0(q^2 + a_1 q + a_2) = (s_0 - y_0)q^2 + (s_1 - a_1 y_0)q - a_2 y_0$$

The estimates are based on the model (3.42) with $A_d = q - 1$ to reduce the effects of the disturbances. Figure 3.16 shows a simulation corresponding to Fig. 3.14 with the modified self-tuning regulator. A comparison with Fig. 3.14 shows a significant improvement. The load disturbance is reduced quickly. Because of the integral action the control will decrease with a magnitude corresponding to the load disturbance shortly after $t = 40$. The parameter

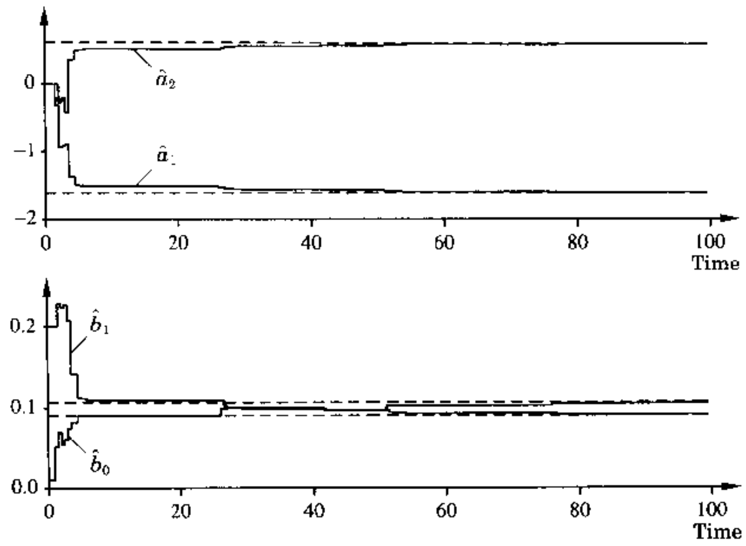

Figure 3.17 Parameter estimates corresponding to Fig. 3.16.

estimates are shown in Fig. 3.17, which indicates the advantages in using the modified estimator. Notice in particular that there is a very small change in the estimates when the load disturbance occurs. □

A Direct Self-tuner with Integral Action

It is also straightforward to introduce integrators in the direct self-tuners. Consider a process model given by

$$A(q)y(t) = B(q)(u(t) + v(t)) \tag{3.43}$$

where $d = \deg A(q) - \deg B(q)$. It is assumed that v is constant or changes infrequently. Let the desired response to command signals be given by

$$A_m(q)y(t) = A_m(1)u_c(t - d) \tag{3.44}$$

where $\deg A_m \geq d$. Let the observer polynomial be $A_o(q)$. The design equation is

$$AR + BS = B^+ A_o A_m \tag{3.45}$$

where $B = b_0 B^+$. If we require that the regulator has integral action, we find that the polynomial R has the form

$$R = R'B^+ = R_1' B^+(q - 1) = R_1' B^+ \Delta \tag{3.46}$$

Equation (3.45) then becomes

$$A\Delta R_1' + b_0 S = A_o A_m \tag{3.47}$$

Hence

$$\begin{aligned} A_o A_m y &= AR_1'\Delta y + b_0 Sy \\ &= BR_1'\Delta u + b_0 R'\Delta v + b_0 Sy \\ &= b_0(R'\Delta u + Sy) + b_0 R'\Delta v \end{aligned} \tag{3.48}$$

where Eq. (3.43) was used to obtain the second equality. Notice that the last term will vanish after a transient if v is constant. If we rewrite Eq. (3.48) in the backwards operator, ignoring v, we get

$$A_o^*(q^{-1})A_m^*(q^{-1})y(t+d) = b_0\Big(R'^*(q^{-1})\Delta^*(q^{-1})u(t) + S^*(q^{-1})y(t)\Big) \tag{3.49}$$

This equation can be used as a basis for parameter estimation, but there are several drawbacks in doing so. First, the operation $A_o^* A_m^*$ is a high-pass filter that is very sensitive to noise. Furthermore, it follows from Eq. (3.47) that

$$b_0 S^*(1) = A_o^*(1)A_m^*(1) = A_o(1)A_m(1) \tag{3.50}$$

All the parameters in the S polynomial are thus not free. If all parameters are estimated, there is, of course, no guarantee that Eq. (3.50) holds. However, it is easy to find a remedy. A polynomial S^* with the property given by Eq. (3.50) can be written as

$$\begin{aligned} b_0 S^* &= A_o(1)A_m(1) + (1-q^{-1})S'^*(q^{-1}) \\ &= A_o(1)A_m(1) + S'^*(q^{-1})\Delta^* \end{aligned}$$

Equation (3.49) then becomes

$$\begin{aligned} A_o^*(q^{-1})&A_m^*(q^{-1})y(t+d) - A_o(1)A_m(1)y(t) \\ &= b_0\Big(R'^*(q^{-1})\Delta^* u(t) + S'^*(q^{-1})\Delta^* y(t)\Big) \\ &= \mathcal{R}^*(q^{-1})\Delta^* u(t) + \mathcal{S}^*(q^{-1})\Delta^* y(t) \end{aligned} \tag{3.51}$$

Division by $A_o^* A_m^*$ now gives

$$y(t+d) - \frac{A_o(1)A_m(1)}{A_o^*(q^{-1})A_m^*(q^{-1})}y(t) = \mathcal{R}^*(q^{-1})u_f(t) + \mathcal{S}^*(q^{-1})y_f(t) \tag{3.52}$$

where

$$u_f(t) = \frac{1-q^{-1}}{A_o^*(q^{-1})A_m^*(q^{-1})}u(t)$$

$$y_f(t) = \frac{1-q^{-1}}{A_o^*(q^{-1})A_m^*(q^{-1})}y(t)$$

Notice that the difference operation eliminates levels and that division by $A_o^* A_m^*$ corresponds to low-pass filtering. Thus the net effect is that the signals

are band-pass filtered with filters that are matched to the desired closed-loop dynamics and the specified observer polynomial.

To complete the algorithm, it now remains to specify how the control law is obtained from the estimated parameters. To obtain the response to command signals given by Eq. (3.44), it follows from Eq. (3.51) that

$$\mathcal{R}^*(q^{-1})\Delta^* u(t) + \mathcal{S}^*(q^{-1})\Delta^* y(t) + A_o(1)A_m(1)y(t) = A_o^*(q^{-1})A_m(1)u_c(t)$$

A controller with integral action may perform poorly if there are actuators that saturate. The feedback loop is broken during saturation, and the integrator may drift to undesirable values. This phenomenon, which is called windup, can be avoided if the control algorithm is modified to

$$\begin{aligned} A_o^*(q^{-1})&\Big(\bar{u}(t) - A_m(1)u_c(t)\Big) \\ &= -A_o(1)A_m(1)y(t) - \mathcal{S}^*(q^{-1})\Delta^* y(t) \\ &\quad - \Big(\mathcal{R}^*(q^{-1})\Delta^* - A_o^*(q^{-1})\Big)u(t) \\ u(t) &= \operatorname{sat}\bar{u}(t) \end{aligned} \tag{3.53}$$

The windup phenomenon is discussed in detail in Section 11.2. In summary, Algorithm 3.6 is obtained.

ALGORITHM 3.6 A direct self-tuning algorithm

Step 1: Estimate the parameters in Eq. (3.52) by recursive least squares.

Step 2: Compute the control signal from Eqs. (3.53) by using the estimates from Step 1. □

This algorithm may be viewed as a practical version of Algorithm 3.3.

3.7 CONCLUSIONS

Deterministic self-tuning regulators have been developed in this chapter. The controllers may be viewed as an attempt to automate the steps of modeling and control design that are normally done by a control system designer. By specifying a model structure, modeling reduces to recursive parameter estimation. Control design results in a map from process parameters to controller parameters. Simple estimation methods (least squares) and simple control design techniques (pole placement) have been used in this chapter. The control design was based on the certainty equivalence principle, which means that the uncertainties in the estimates are neglected in computing the control law. Two classes of algorithms have been discussed: indirect and direct algorithms. The indirect algorithms are a straightforward implementation in which process parameters are estimated and the controller parameters are computed by using some design equations. In the direct algorithms the controller parameters are estimated directly. To do this, design equations are used to reparameterize the process model in the controller parameters. This makes it possible to establish relations between MRAS and STR, as is discussed in Chapter 5.

PROBLEMS

3.1 In sampling a continuous-time process model with $h = 1$ the following pulse transfer function is obtained:

$$H(z) = \frac{z + 1.2}{z^2 - z + 0.25}$$

The design specification states that the discrete-time closed-loop poles should correspond to the continuous-time characteristic polynomial

$$s^2 + 2s + 1$$

(a) Design a minimal-order discrete-time indirect self-tuning regulator. The controller should have integral action and give a closed-loop system having unit gain in stationary. Determine the Diophantine equation that solves the design problem.

(b) Suggest a design that includes direct estimation of the controller parameters. Discuss why a well-working direct self-tuning regulator is more difficult to design for this process than is an indirect self-tuning regulator.

3.2 Consider the process

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

where a is an unknown parameter. Assume that the desired closed-loop system is

$$G_m(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

Construct continuous- and discrete-time indirect self-tuning algorithms for the system.

3.3 Consider the system

$$G(s) = G_1(s)G_2(s)$$

where

$$G_1(s) = \frac{b}{s + a}$$

$$G_2(s) = \frac{c}{s + d}$$

where a and b are unknown parameters and c and d are known. Construct discrete-time direct and indirect self-tuning algorithms for the partially known system.

3.4 A process has the transfer function

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

where b is a time-varying parameter. The system is controlled by a proportional controller

$$u(t) = k\,(u_c(t) - y(t))$$

It is desirable to choose the feedback gain so that the closed-loop system has the transfer function

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

Construct a continuous-time indirect self-tuning algorithm for the system.

3.5 The code for simulating Examples 3.4 and 3.5 is listed below. Study the code and try to understand the details.

```
DISCRETE SYSTEM reg

"Indirect Self-Tuning Regulator based on the model
"   H(q)=(b0*q+b1)/(q^2+a1*q+a2)
"using standard RLS estimation and pole placement design
"Polynomial B is canceled if cancel>0.5

INPUT ysp y                    "set point and process output
OUTPUT u                       "control variable
STATE ysp1 y1 u1 v1            "controller states
STATE th1 th2 th3 th4          "parameter estimates
STATE f1   f2  f3  f4          "regression variables
STATE p11 p12 p13 p14          "covariance matrix
STATE     p22 p23 p24
STATE         p33 p34
STATE             p44
NEW  nysp1 ny1 nu1 nv1
NEW nth1 nth2 nth3 nth4
NEW nf1 nf2 nf3 nf4
NEW n11 n12 n13 n14 n22 n23 n24 n33 n34 n44
TIME t
TSAMP ts

INITIAL
"Compute sampled Am and Ao
a=exp(-z*w*h)
am1=-2*a*cos(w*h*sqrt(1-z*z))
```

```
am2=a*a
aop=IF w*To>100 THEN 0 ELSE -exp(-h/To)
ao=IF cancel>0.5 THEN 0 ELSE -aop

SORT
"1.0 Parameter Estimation
"1.1 Computation of P*f and estimator gain k
pf1=p11*f1+p12*f2+p13*f3+p14*f4
pf2=p12*f1+p22*f2+p23*f3+p24*f4
pf3=p13*f1+p23*f2+p33*f3+p34*f4
pf4=p14*f1+p24*f2+p34*f3+p44*f4
denom=lambda+f1*pf1+f2*pf2+f3*pf3+f4*pf4
k1=pf1/denom
k2=pf2/denom
k3=pf3/denom
k4=pf4/denom

"1.2 Update estimates and covariances
eps=y-f1*th1-f2*th2-f3*th3-f4*th4
nth1=th1+k1*eps
nth2=th2+k2*eps
nth3=th3+k3*eps
nth4=th4+k4*eps
n11=(p11-pf1*k1)/lambda
n12=(p12-pf1*k2)/lambda
n13=(p13-pf1*k3)/lambda
n14=(p14-pf1*k4)/lambda
n22=(p22-pf2*k2)/lambda
n23=(p23-pf2*k3)/lambda
n24=(p24-pf2*k4)/lambda
n33=(p33-pf3*k3)/lambda
n34=(p34-pf3*k4)/lambda
n44=(p44-pf4*k4)/lambda

"1.3 Update and filter regression vector
nf1=-y
nf2=f1
nf3=u
nf4=f3


"2.0 Control design
"2.1 Rename parameters
a1=nth1
a2=nth2
```



```
b0=nth3
b1=nth4

"2.2 Solve the polynomial identity AR+BS=AoAm
n=b1*b1-a1*b0*b1+a2*b0*b0
r10=(ao*am2*b0^2+(a2-am2-ao*am1)*b0*b1+(ao+am1-a1)*b1^2)/n
w1=(a2*am1+a2*ao-a1*a2-am2*ao)*b0
s00=(w1+(-a1*am1-a1*ao-a2+a1^2+am2+am1*ao)*b1)/n
w2=(-a1*am2*ao+a2*am2+a2*am1*ao-a2^2)*b0
s10=(w2+(-a2*am1-a2*ao+a1*a2+am2*ao)*b1)/n

"2.3 Compute polynomial T=Ao*Am(1)/B(1)
bs=b0+b1
as=1+am1+am2
bm0=as/bs

"2.4 Choose control algorithm
r1=IF cancel>0.5 THEN b1/b0 ELSE r10
s0=IF cancel>0.5 THEN (am1-a1)/b0 ELSE s00
s1=IF cancel>0.5 THEN (am2-a2)/b0 ELSE s10
t0=IF cancel>0.5 THEN as/b0 ELSE bm0
t1=IF cancel>0.5 THEN 0 ELSE bm0*ao

"3.0 Control law with anti-windup
v=-ao*v1+t0*ysp+t1*ysp1-s0*y-s1*y1+(ao-r1)*u1
u=IF v<-ulim THEN -ulim ELSE IF v<ulim THEN v ELSE ulim

"3.1 Update controller state
ny1=y
nu1=u
nv1=v
nysp1=ysp

"4.0  Update sampling time
ts=t+h

"Parameters
lambda:1        "forgetting factor
To:200          "observer time constant
z:0.7           "desired closed loop damping
w:1             "desired closed loop natural frequency
h:1             "sampling period
ulim:1          "limit of control signal
cancel:1        "switch for cancellation
th1:-2          "initial estimates
```

```
th2:1
th3:0.01
th4:0.01
p11:100          "initial covariances
p22:100
p33:100
p44:100

END
```

3.6 Consider the simulation of the indirect self-tuning regulator in Example 3.5. Investigate how the transient behavior of the algorithm depends on the initial values of θ and P and the forgetting factor.

3.7 Consider the indirect self-tuning regulator in Example 3.5. Make a simulation over longer time periods, and investigate how the parameters approach their true values. Also explore how the convergence rate depends on the forgetting factor λ.

3.8 Consider the indirect self-tuning regulator in Example 3.5. Show that no steady-state error is obtained if

$$\hat{a}_1 + \hat{a}_2 = 1$$

Modify the simulation used to generate Figs. 3.6 and 3.7, plot the parameter combination $\hat{a}_1 + \hat{a}_2$, and check how well the above condition is satisfied.

3.9 Consider the indirect self-tuning regulator in Example 3.5. Change the specifications on the closed-loop system, and investigate how the behavior of the system changes.

3.10 Consider the indirect self-tuning regulator in Example 3.5. Modify the simulation program so that the parameters of the process can be changed. Investigate experimentally how well the adaptive system can follow reasonable parameter variations.

3.11 Apply the indirect self-tuning regulator in Example 3.5 to a process with the transfer function

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

Study and explain the behavior of the error when the reference signal is a square wave.

3.12 The code for simulating Example 3.6 is listed below. Study the code and try to understand all the details.

```
CONTINUOUS SYSTEM reg
"Continuous time STR for the system b/[s(s+a)]
```



```
"Desired response given by am2/(s^2+am1*s+am2)
"Observer polynomial s+ao

INPUT y ysp
OUTPUT u
STATE yf yf1 uf uf1 xu
STATE th1 th2
STATE p11 p12 p22
DER dyf dyf1 duf duf1 dxu
DER dth1 dth2
DER dp11 dp12 dp22

"Filter input and output
dyf=yf1
dyf1=-am1*yf1+am2*(y-yf)
duf=uf1
duf1=-am1*uf1+am2*(u-uf)

"Update parameter estimate
f1=-yf1
f2=uf
e=dyf1-f1*th1-f2*th2
pf1=p11*f1+p12*f2
pf2=p12*f1+p22*f2
dth1=pf1*e
dth2=pf2*e

"Update covariance matrix
dp11=alpha*p11-pf1*pf1
dp12=alpha*p12-pf1*pf2
dp22=alpha*p22-pf2*pf2
det=p11*p22-p12*p12


"Control design
a=th1
b=th2
r1=am1+ao-a
s0=(am2+am1*ao-a*r1)/b
s1=am2*ao/b
t0=am2/b

"Control signal computation
dxu=-ao*xu-(s1-ao*s0)*y+(ao-r1)*u
v=t0*ysp-s0*y+xu
```

```
u=if v<-ulim then -ulim else if v>ulim then ulim else v

"Parameters
am1:1.4
am2:1
alpha:0
ao:2
ulim:4
END
```

3.13 Consider the simulation of the continuous-time indirect self-tuning regulator in Example 3.6. Investigate how the transient behavior of the algorithm depends on the initial values of θ and P.

3.14 Consider the indirect self-tuning regulator in Example 3.6. Make a simulation, and investigate how the convergence rate depends on the forgetting factor α.

3.15 Consider the system in Problem 1.9.

(a) Sample the system, and determine a discrete-time controller for the known nominal system such that the specifications are satisfied.

(b) Use a direct self-tuning controller, and study the transient for different initial conditions and different values of the variable parameters of the system.

(c) Assume that $e = 0$ and that u_c is a square wave. Simulate a self-tuning controller for different prediction horizons.

(d) Investigate the behavior when the disturbance d is a step. What happens when the controller does not have an integrator?

REFERENCES

The pole placement design is extensively discussed in:

Åström, K. J., and B. Wittenmark, 1990. *Computer Controlled Systems—Theory and Design*, 2nd edition. Englewood Cliffs, N.J.: Prentice-Hall.

It is possible to solve the Diophantine equation using polynomial calculations. Solution of the Diophantine equation is discussed in:

Blankinship, W. A., 1963. "A new version of the Euclidean algorithm." *American Mathematics Monthly* **70**: 742–745.

Kučera, V., 1979. *Discrete Linear Control—The Polynomial Equation Approach*. New York: John Wiley.

Ježek, J., 1982. "New algorithm for minimal solution of linear polynomial equations." *Kybernetica* **18**: 505–516.

There are many papers, reports, and books about self-tuning algorithms. Some fundamental references are given in this section. The first publication of the self-tuning idea is probably:

Kalman, R. E., 1958. "Design of a self-optimizing control system." *Trans. ASME* **80**: 468–478.

In this paper, least-squares estimation combined with deadbeat control is discussed. No analysis is given of the properties of the closed-loop system. A prototype special-purpose computer was built to implement the controller, but the development was hampered by hardware problems. The main development of the theory for self-tuning controllers was first done for discrete-time systems with stochastic noise. This type of self-tuning controllers is discussed in the next chapter. Two similar algorithms based on least-squares estimation and minimum-variance control were presented at an IFAC symposium in Prague 1970:

Peterka, V., 1970. "Adaptive digital regulation of noisy systems." *Preprints 2nd IFAC Symposium on Identification and Process Parameter Estimation*. Prague.

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

The first thorough presentation and analysis of a self-tuning regulator was given in:

Åström, K. J., and B. Wittenmark, 1972. "On the control of constant but unknown systems." *5th IFAC World Congress*. Paris.

A revised version of this paper, in which the phrase "self-tuning regulator" was coined, is:

Åström, K. J., and B. Wittenmark, 1973. "On self-tuning regulators." *Automatica* **9**: 185–199.

The preceding papers inspired intensive research activity in adaptive control based on the self-tuning idea. A comprehensive treatment of the fundamental theory of adaptive control, especially self-tuning algorithms, is 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.

Pole placement and model-reference-type self-tuners are treated in:

Wellstead, P. E., J. M. Edmunds, D. Prager, and P. Zanker, 1979. "Self-tuning pole/zero assignment regulators." *Int. J. Control* **30**: 1–26.

Åström, K. J., and B. Wittenmark, 1980. "Self-tuning regulators based on pole-zero placement." *IEE Proceedings Part D* **127**: 120–130.

Continuous-time self-tuning regulators are discussed in:

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

Gawthrop, P. J., 1987. *Continuous-Time Self-Tuning Control I*. Letchworth, U.K.: Research Studies Press.

The book by Egardt also gives a unification of MRAS and STR.

CHAPTER 4

STOCHASTIC AND PREDICTIVE SELF-TUNING REGULATORS

4.1 INTRODUCTION

In Chapter 3 the key issue was to find self-tuning controllers that give desired responses to command signals. In this chapter we discuss self-tuners for the regulation problem. The key issue is now to design a controller that reduces disturbances as well as possible. Stochastic models are useful to describe disturbances. For this reason we start in Section 4.2 by describing a simple stochastic control problem. This leads to a minimum-variance controller and its generalization, the moving-average controller. In Section 4.3 we present a direct adaptive controller that has the surprising property that the moving-average controller is an equilibrium solution. This surprising property was one of the motivating factors in the original work on the self-tuning regulator. The minimum-variance controller has the drawback that its properties are critically dependent on the sampling period. In Section 4.4 some extensions are therefore presented. Linear quadratic Gaussian self-tuners are discussed in Section 4.5, and adaptive predictive control is discussed in Section 4.6.

4.2 DESIGN OF MINIMUM-VARIANCE AND MOVING-AVERAGE CONTROLLERS

In this section we derive controllers for linear stochastic systems. It is assumed that the process can be described by a pulse transfer function and that the disturbances acting on the system are filtered white noise. A steady-state

regulation problem is considered. The criterion is based on the mean square deviations of the output and the control signal.

Process Model

Assume that the process dynamics are characterized by

$$x(t) = \frac{B_1(q)}{A_1(q)}u(t)$$

where $A_1(q)$ and $B_1(q)$ are polynomials in the forward shift operator without any common factors.

It is assumed that the action of the disturbances on the system can be described as filtered white noise. Since the system is linear, we can reduce all disturbances to an equivalent disturbance v at the system output. The output is thus given by

$$y(t) = x(t) + v(t)$$

where

$$v(t) = \frac{C_1(q)}{A_2(q)}e(t)$$

$C_1(q)$ and $A_2(q)$ are polynomials in the forward shift operator without any common factors, and $\{e(t)\}$ is a sequence of independent random variables (white noise) with zero mean and standard deviation σ.

The process can now be reduced to the standard form

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \tag{4.1}$$

where

$$\begin{aligned} A &= A_1A_2 \\ B &= B_1A_2 \\ C &= C_1A_1 \end{aligned} \tag{4.2}$$

Because of the assumptions, the three polynomials have no common factor. The model (4.1) is thus a minimal representation. The polynomials are normalized such that both the A and C polynomials are monic, that is, the leading coefficients are unity. Finally, the C polynomial can be multiplied by an arbitrary power of q without changing the correlation structure of $C(q)e(t)$. This is used to normalize C such that

$$\deg C = \deg A = n$$

The A and B polynomials may have zeros inside or outside the unit disc. It is assumed that the zeros of the C polynomial are inside the unit disc. By spectral factorization the polynomial $C(q)$ can be changed so that all its zeros are inside the unit disc or on the unit circle. An example shows how this is done.

EXAMPLE 4.1 Modification of the polynomial C

Consider the polynomial

$$C(z) = z + 2$$

which has the zero $z = -2$ outside the unit disc. Consider the signal

$$v(t) = C(q)e(t)$$

where $\{e(t)\}$ is a sequence of uncorrelated random variables with zero mean and unit variance. The spectral density of v is given by

$$\Phi(e^{i\omega h}) = \frac{1}{2\pi} C(e^{i\omega h})C(e^{-i\omega h})$$

Because

$$\begin{aligned} C(z)C(z^{-1}) &= (z+2)(z^{-1}+2) = (1+2z^{-1})(1+2z) \\ &= (2z+1)(2z^{-1}+1) \\ &= 4(z+0.5)(z^{-1}+0.5) \end{aligned}$$

the signal v may also be represented as

$$v(t) = C^*(q)e(t)$$

where

$$C^*(z) = 2z + 1$$

is the reciprocal of the polynomial $C(z)$ (see Section 3.2). □

If the calculations (4.2) give a polynomial C that has zeros outside the unit disc, the polynomial is factored as

$$C = C^+C^-$$

where C^- contains all factors with zeros outside the unit disc. The C polynomial is then replaced by C^+C^{-*}. The model (4.1) is an *innovations representation*. It will be shown later that $e(t)$ is the innovation or the error in predicting the signal $y(t)$ over one sampling period. The C polynomial can be interpreted as the characteristic polynomial of the estimator or predictor.

Criteria

In steady-state regulation it makes sense to express the criteria in terms of the steady-state variances of the output and the control signals. This leads to the performance criterion

$$J = E\{y^2(t) + \rho u^2(t)\} \tag{4.3}$$

where E denotes mathematical expectation with respect to the noise process acting on the system. The control law minimizing (4.3) is the *linear quadratic*

Gaussian (LQG) controller. If $\rho = 0$, then the resulting controller is called the *minimum-variance (MV) controller*.

The properties of the control signal when the minimum-variance controller is used depend critically on the sampling interval. A short sampling interval gives large variance in the control signal, and a long sampling interval gives a low variance. Notice that the loss function (4.3) is defined in discrete time, that is, only the behavior at the sampling instances is considered.

To define the design problem, it is also necessary to define the admissible controllers. It will be assumed that $u(t)$ is allowed to be a function of $y(t)$, $y(t-1), \ldots, u(t-1), u(t-2), \ldots$.

Minimum-Variance Control

It is now assumed that $\rho = 0$ and that the process is minimum-phase, that is, that the B polynomial has all zeros inside the unit disc. Before we solve the general problem, we consider a simple example.

EXAMPLE 4.2 Minimum-variance control of a first-order system

Consider the first-order system

$$y(t+1) + ay(t) = bu(t) + e(t+1) + ce(t) \tag{4.4}$$

where $|c| < 1$ and $\{e(t)\}$ is a sequence of independent random variables with unit variance.

Consider the output at time $t+1$. From (4.4) it follows that by using $u(t)$ it is possible to change $y(t+1)$ arbitrarily. Further, $e(t+1)$ is independent of $y(t)$ and $u(t)$; thus

$$\operatorname{var} y(t+1) \geq \operatorname{var} e(t+1) = 1$$

Given measurements up to time t, we can use Eq. (4.4) to compute $e(t)$. The controller

$$u(t) = \frac{ay(t) - ce(t)}{b} \tag{4.5}$$

gives

$$y(t+1) = e(t+1) \tag{4.6}$$

which gives the lower bound of the variance of y. If Eq. (4.5) is used all the time, then from Eq. (4.6), it follows that $y(t) = e(t)$, and we get the controller

$$u(t) = \frac{a-c}{b} y(t) \tag{4.7}$$

□

The minimum-variance controller can in the general case be derived by using similar ideas as Example 4.2. Define

$$d_0 = \deg A - \deg B$$

as the pole excess of the system. This is the same as the time delay in the system. The input at time t will influence the output first at time $t + d_0$. Now consider

$$y(t + d_0) = \frac{B}{A} u(t + d_0) + \frac{C}{A} e(t + d_0) \tag{4.8}$$

Let the polynomial F of degree $d_0 - 1$ be the quotient, and let the polynomial G of degree $n - 1$ be the remainder when $q^{d_0-1}C$ is divided by A. Hence

$$\frac{q^{d_0-1}C(q)}{A(q)} = F(q) + \frac{G(q)}{A(q)}$$

This can be interpreted as a Diophantine equation,

$$q^{d_0-1}C(q) = A(q)F(q) + G(q) \tag{4.9}$$

Hence the output at $t + d_0$ can be written as

$$y(t + d_0) = \frac{B}{A}u(t + d_0) + Fe(t + 1) + \frac{qG}{A} e(t)$$

where

$$F(q) = q^{d_0-1} + f_1q^{d_0-2} + \dots + f_{d_0-1} \tag{4.10}$$
$$G(q) = g_0q^{n-1} + g_1q^{n-2} + \dots + g_{n-1} \tag{4.11}$$

From Eq. (4.1) we can determine $e(t)$:

$$e(t) = \frac{A}{C} y(t) - \frac{B}{C} u(t)$$

From the measurement of $y(t)$ and $u(t)$ it is thus possible to compute the noise sequence, the innovations. This equation is an observer in which the dynamics are given by the C polynomial. It now follows that

$$\begin{aligned} y(t + d_0) &= Fe(t + 1) + \left(\frac{B}{A}q^{d_0} - \frac{qGB}{AC}\right) u(t) + \frac{qGA}{AC} y(t) \\ &= Fe(t + 1) + \frac{Bq}{AC}\left(q^{d_0-1}C - G\right) u(t) + \frac{qG}{C} y(t) \\ &= Fe(t + 1) + \frac{qBF}{C} u(t) + \frac{qG}{C} y(t) \end{aligned} \tag{4.12}$$

where Eq. (4.9) has been used to obtain the last equality. The polynomials qG, qBF, and C are all of degree n. This implies that we have divided $y(t + d_0)$ in two parts. The first part, $F(q)e(t + 1)$, depends on the noise acting on the system from $t + 1, \dots, t + d_0$. The second part,

$$\hat{y}(t + d_0|t) = \frac{qBF}{C} u(t) + \frac{qG}{C} y(t) \tag{4.13}$$

depends on measured outputs and applied inputs, including the $u(t)$ that we want to determine. From Eqs. (4.12) it follows that $\hat{y}(t + d_0|t)$ is the mean

square prediction of $y(t + d_0)$ given data up to and including time t. The prediction error is given by

$$\tilde{y}(t + d_0|t) = y(t + d_0) - \hat{y}(t + d_0|t) = F(q)e(t + 1)$$

and the variance of the prediction error is

$$\operatorname{var} \tilde{y}(t + d_0|t) = \sigma^2(1 + f_1^2 + f_2^2 + \dots + f_{d_0-1}^2)$$

Minimum variance of the output is now obtained by the control law

$$u(t) = -\frac{G(q)}{B(q)F(q)} y(t) \tag{4.14}$$

Using this controller gives

$$\begin{aligned} y(t + d_0) &= F(q)e(t + 1) \\ &= e(t + d_0) + f_1e(t + d_0 - 1) + \dots + f_{d_0-1}e(t + 1) \end{aligned} \tag{4.15}$$

and the minimum output variance is

$$\operatorname{var} y(t) = \sigma^2(1 + f_1^2 + f_2^2 + \dots + f_{d_0-1}^2)$$

which is the same as the variance of the prediction error. Using the controller (4.14) gives the closed-loop characteristic equation

$$q^{d_0-1}C(q)B(q) = 0$$

This implies that there are $d_0 - 1$ poles at the origin, n poles at the zeros of the C polynomial, which are inside the unit disc, and $n - d_0$ poles at the zeros of the B polynomial. Since the system was assumed to be minimum-phase, these poles are also inside the unit disc. Observe that minimum-variance control is the same as predicting the output d_0 steps ahead and then choosing the control signal such that the predicted value is equal to the desired reference value. See Fig. 4.1.

The minimum-variance controller can be interpreted as a pole placement controller, which was discussed in Section 3.2. This is seen by multiplying Eq. (4.9) by B, that is,

$$q^{d_0-1}CB = AR + BS \tag{4.16}$$

where

$$R = BF$$
$$S = G$$

The pole placement design leads to the controller

$$u(t) = -\frac{S}{R} y(t) = -\frac{G}{BF} y(t)$$

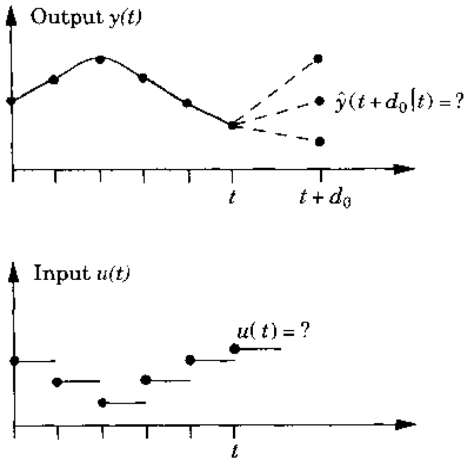

Figure 4.1 Minimum variance control is based on prediction d_0 steps ahead.

Nonminimum-Phase Systems

When the system is nonminimum phase, it is not possible to place some of the closed-loop poles at the zeros of the B polynomial. It can be shown that the optimal controller minimizing Eq. (4.3) with $\rho = 0$ gives the following closed-loop characteristic equation:

$$q^{d_0-1}B^+(q)B^{-*}(q)C(q) = 0$$

that is, the process zeros outside the unit disc, $B^-(q)$, are replaced by the zeros defined by the reciprocal polynomial, $B^{-*}(q)$. See Åström and Wittenmark (1990) in the references at the end of the chapter. The controller

$$u(t) = -\frac{S}{R}y(t)$$

is now obtained from the Diophantine equation

$$q^{d_0-1}B^+B^{-*}C = AR + BS \tag{4.17}$$

Compare Eq. (4.16).

Moving-Average Controller

The minimum-variance controller leads to a closed-loop system in which the output is a moving average of order $d_0 - 1$ (see Eq. (4.15)). It is possible to design controllers such that the output is a moving average of higher order.

Instead of placing $d_0 - 1$ closed-loop poles at the origin, we may place $d - 1$ poles, where $d \geq d_0$.

The moving-average controller can be derived as follows. Factor the B polynomial as

$$B(q) = B^+(q)B^-(q)$$

where B^+ corresponds to well-damped zeros. To obtain a unique factorization, it is assumed that B^- is monic. Determine R and S from

$$q^{d-1}B^+C = AR + BS \tag{4.18}$$

It follows that B^+ must be a factor of R, that is, $R = R_1B^+$. With the feedback law

$$u(t) = -\frac{S}{R}y(t)$$

we get

$$Ay(t) = B\left(-\frac{S}{R}\right)y(t) + Ce(t)$$

or

$$\begin{aligned} y(t) &= \frac{CR}{AR+BS}e(t) = \frac{CB^+R_1}{q^{d-1}B^+C}e(t) \\ &= \frac{R_1}{q^{d-1}}e(t) = \left(1 + r_1q^{-1} + \dots + r_{d-1}q^{-d+1}\right)e(t) \end{aligned}$$

where $\deg R_1 = d - 1$ with

$$d = \deg A - \deg B^+$$

Since the controlled output is a moving-average process of order $d - 1$, we call the strategy *moving-average (MA) control*. Notice that no zeros are canceled if

$$B^+ = 1$$

which means that

$$d = \deg A = n$$

The minimum-variance controller and the moving-average controller are similar. The only difference is the value of the integer d, which controls the number of process zeros that are canceled. With $d = d_0$, all process zeros are canceled; with $d = \deg A = n$, no process zeros are canceled.

EXAMPLE 4.3 Moving-average controller

Consider the system (4.1) with

$$\begin{aligned} A(q) &= q^2 + a_1q + a_2 \\ B(q) &= b_0q + b_1 \\ C(q) &= q^2 + c_1q + c_2 \end{aligned}$$

In this case, $d_0 = 1$. The minimum-variance controller is obtained from Eq. (4.9), giving the controller

$$u(t) = -\frac{(c_1 - a_1) + (c_2 - a_2)q^{-1}}{b_0 + b_1 q^{-1}} y(t)$$

and the closed-loop system is

$$y(t) = e(t)$$

The minimum-variance controller can be used only if $|b_1/b_0| < 1$, that is, for the minimum-phase case.

The moving-average controller is obtained by solving Eq. (4.18). In this case, $d = 2$ and $B^+(q) = 1$. This gives the Diophantine equation

$$q(q^2 + c_1 q + c_2) = (q^2 + a_1 q + a_2)(q + r_1) + (b_0 q + b_1)(s_0 q + s_1)$$

Notice that this is the same as Eq. (3.19) with $A_o(q) = q$ and $A_m(q) = C(q)$. The solution is thus given by Eqs. (3.20) and (3.21):

$$r_1 = \frac{(a_2 - c_2)b_0 b_1 + (c_1 - a_1)b_1^2}{b_1^2 + a_1 b_0 b_1 + a_2 b_0^2}$$

$$s_0 = \frac{b_1(a_1^2 - a_2 - c_1 a_1 + c_2) + b_0(c_1 a_2 - a_1 a_2)}{b_1^2 + a_1 b_0 b_1 + a_2 b_0^2}$$

$$s_1 = \frac{b_1(a_1 a_2 - c_1 a_2) + b_0(a_2 c_2 - a_2^2)}{b_1^2 + a_1 b_0 b_1 + a_2 b_0^2}$$

The closed-loop system is

$$y(t) = (1 + r_1 q^{-1})e(t)$$

□

LQG Control

The pole placement and LQG problems are closely related. In the LQG formulation a loss function is specified. Minimization of the loss function leads to a fixed-gain controller that can be interpreted in terms of pole placement. The details are given in Section 4.5. To obtain the LQG solution, it is first necessary to solve the spectral factorization problem, that is, to find the nth-order monic, stable polynomial $P(q)$ that satisfies

$$rP(q)P(q^{-1}) = \rho A(q)A(q^{-1}) + B(q)B(q^{-1}) \tag{4.19}$$

The LQG-controller is then obtained as the solution to the Diophantine equation

$$C(q)P(q) = A(q)R(q) + B(q)S(q) \tag{4.20}$$

To get a unique solution with $\deg R = \deg S = n$, it is necessary to make some further restrictions to the solution given by Eq. (4.20). See Theorem 4.3 in Section 4.5. The interpretation of Eq. (4.20) is that the LQG-controller places the closed-loop poles in $P(q)$, given by the spectral factorization, and in $C(q)$, which characterizes the disturbances.

Summary

The minimum-variance controller, the moving-average controller, and the LQG-controller can all be interpreted as pole placement design as discussed in Section 3.2. The minimum-variance controller is obtained by solving the Diophantine equation (4.16) for the minimum-variance case or Eq. (4.17) for the nonminimum-variance case. The moving-average controller is given by Eq. (4.18) and the LQG-controller by Eq. (4.20). The closed-loop characteristic polynomial is chosen differently for each of the design methods.

4.3 STOCHASTIC SELF-TUNING REGULATORS

Indirect Self-tuning Regulator

A straightforward way to make a self-tuning regulator for the process (4.1) is to estimate the parameters in the A, B, and C polynomials by using, for instance, the extended least squares (ELS) algorithm or the recursive maximum-likelihood (RML) algorithm. (See Section 2.2.) The estimated parameters are then used in the design equation (4.9) if minimum-variance control is desired or in Eq. (4.20) if LQG control is desired.

EXAMPLE 4.4 Stochastic indirect self-tuning regulator

Consider the process (4.4) in Example 4.2 with $a = -0.9, b = 3$, and $c = -0.3$. The minimum-variance controller is given by the proportional controller

$$u(t) = \frac{a - c}{b} y(t) = -s_0 y(t) = -0.2y(t)$$

This gives the closed-loop system

$$y(t) = e(t)$$

The ELS method is used to estimate the unknown parameters a, b, and c. The estimates are obtained from Eq. (2.21) with

$$\theta^T = \begin{pmatrix} a & b & c \end{pmatrix}$$

$$\varphi^T(t-1) = \begin{pmatrix} -y(t-1) & u(t-1) & \varepsilon(t-1) \end{pmatrix}$$

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

The controller is

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

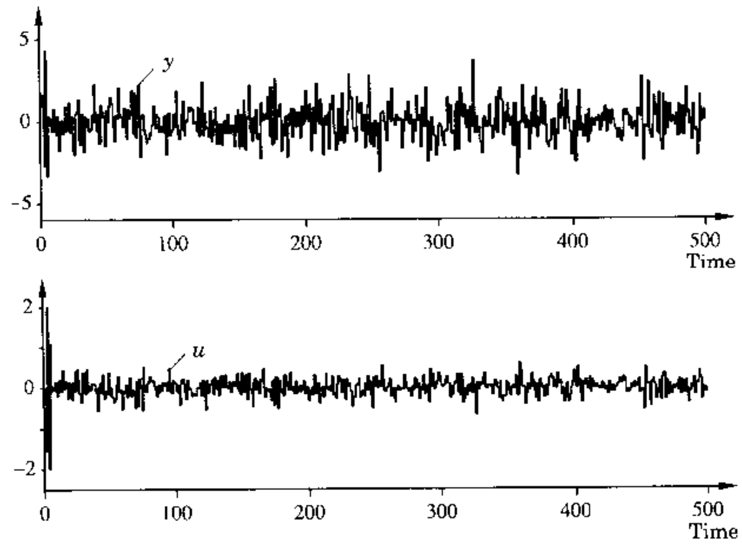

Figure 4.2 Output and input when an indirect self-tuning regulator based on minimum-variance control is used to control the system in Example 4.4.

Figure 4.2 shows the result of a simulation of the algorithm. The initial values in the simulation are

$$\begin{aligned} \hat{a}(0) &= 0 \\ \hat{b}(0) &= 1 \\ \hat{c}(0) &= 0 \\ P(0) &= 100I \end{aligned}$$

Figure 4.3 shows the accumulated loss

$$V(t) = \sum_{i=1}^{t} y^2(i)$$

when the optimal minimum-variance controller and the indirect self-tuning regulator are used. The curve of the accumulated loss of the STR is almost parallel to the optimal curve. This means that the performance of the self-tuning regulator is almost optimal except for a short startup transient. Figure 4.4 shows the estimated process parameters. The parameter estimates have not converged to the true values during the simulated period. However, the controller parameter $\hat{s}_0(t) = (\hat{a}(t) - \hat{c}(t))/\hat{b}(t)$ converges faster, as can be seen in Fig. 4.5. For a fixed controller the closed-loop system is stable when $-0.03 < s_0 < 0.63$. Notice that during some of the first steps the controller

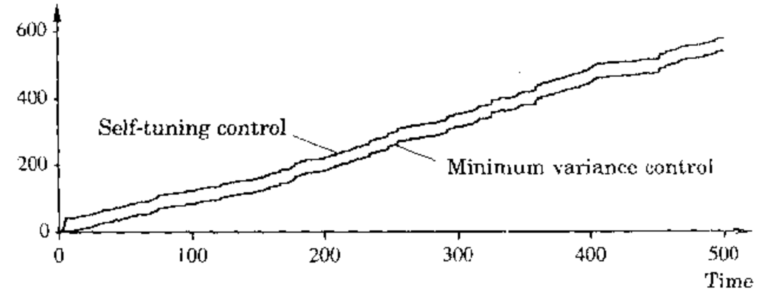

Figure 4.3 The accumulated loss when a self-tuning regulator and the optimal minimum-variance controller are used on the system in Example 4.4.

parameter $\hat{s}_0(t)$ is such that the closed-loop system would be unstable if the controller were frozen to those values.

The reason for the poor convergence of the three estimated process parameters is that the controller converges rapidly to a minimum-variance controller. After that there is poor excitation of the process. The example shows that the self-tuning controller compares well with the optimal controller for the known system. From the control law it can be seen that there may be numerical problems when $\hat{b}(t)$ is small. □

Direct Minimum-Variance and Moving-Average STR

The design calculations for the indirect self-tuning regulators include the solution of a system of equations such as the Diophantine equation (4.18) or (4.20). The time to solve the Diophantine equation may be long in comparison with the sampling period. A self-tuning regulator that directly estimates the controller parameters eliminates the design calculations. It is thus desirable to

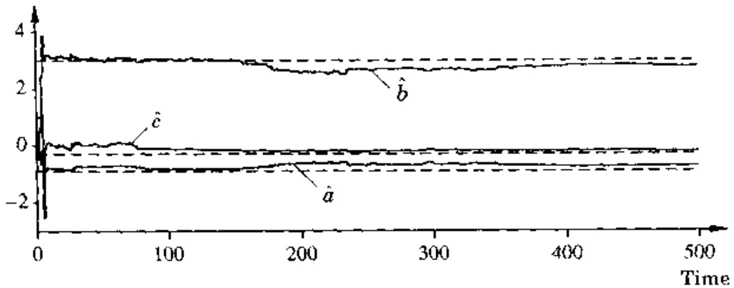

Figure 4.4 The estimated parameters $\hat{a}(t)$, $\hat{b}(t)$, and $\hat{c}(t)$ when the system in Example 4.4 is controlled. The dashed lines correspond to the true parameter values.

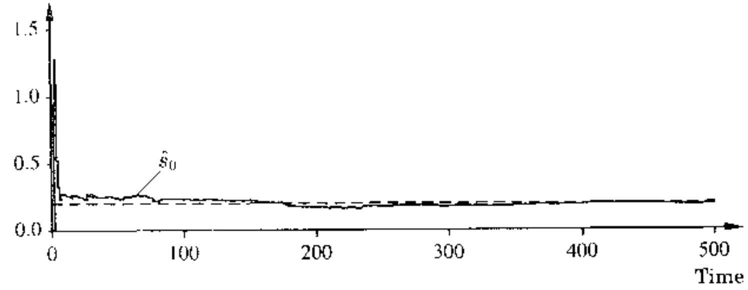

Figure 4.5 The controller parameter $\hat{s}_0(t)$ when the system in Example 4.4 is controlled. The dashed line is the optimal parameter for the minimum-variance controller.

construct direct self-tuning algorithms. Deterministic direct self-tuners were discussed in Section 3.5. The idea is to use the specification and the process model to make a reparameterization of the system. The same idea will now be used for stochastic systems of the form (4.1). In Section 4.2 it was shown that minimum-variance control is the same as predicting the output d_0 steps ahead and then determining the control signal $u(t)$ such that the predicted value is equal to the desired output. Consider the reparameterization (4.12), and rewrite the model in the backward shift operator. This gives

$$y(t+d_0) = \frac{1}{C^*}\left(R^*u(t) + S^*y(t)\right) + R_1^*e(t+d_0) \tag{4.21}$$

where $R_1^* = F^*$ and $\deg R_1 = d_0 - 1$. Using Eq. (4.18), we get, in the same way,

$$y(t+d) = \frac{B^{-*}}{C^*}\left(R^*u(t) + S^*y(t)\right) + R_1^*e(t+d) \tag{4.22}$$

where $\deg R_1 = d - 1$.

The factors $1/C^*$ and B^{-*}/C^* in Eqs. (4.21) and (4.22), respectively, can be interpreted as filters for the regressors. (Compare Section 3.5.) Both equations are now written in predictor form, where the controller polynomials R and S appear directly in the model. These equations can be used as a motivation for the following algorithm.

ALGORITHM 4.1 Basic direct self-tuning algorithm

Data: Given the prediction horizon d, let k and l be the degrees of the R^* and S^* polynomials, respectively. Let Q^*/P^* be a stable filter.

Step 1: Estimate the coefficients of the polynomials R^* and S^* of the model

$$y(t+d) = R^*(q^{-1})u_f(t) + S^*(q^{-1})y_f(t) + \varepsilon(t+d) \tag{4.23}$$

where

$$R^*(q^{-1}) = r_0 + r_1q^{-1} + \cdots + r_kq^{-k}$$
$$S^*(q^{-1}) = s_0 + s_1q^{-1} + \cdots + s_lq^{-l}$$

and

$$u_f(t) = \frac{Q^*(q^{-1})}{P^*(q^{-1})}u(t)$$
$$y_f(t) = \frac{Q^*(q^{-1})}{P^*(q^{-1})}y(t)$$

using Eq. (2.21) with

$$\varepsilon(t) = y(t) - R^*u_f(t-d) - S^*y_f(t-d) = y(t) - \varphi^T(t-d)\hat{\theta}(t-1)$$
$$\varphi^T(t) = \frac{Q^*(q^{-1})}{P^*(q^{-1})}\begin{pmatrix} u(t) & \ldots & u(t-k) & y(t) & \ldots & y(t-l)\end{pmatrix}$$
$$\theta^T = \begin{pmatrix} r_0 & \ldots & r_k & s_0 & \ldots & s_l\end{pmatrix}$$

Step 2: Calculate the control signal from

$$R^*(q^{-1})u(t) = -S^*(q^{-1})y(t) \tag{4.24}$$

with R^* and S^* given by the estimates obtained in Step 1.

Repeat Steps 1 and 2 at each sampling period. □

Remark 1. Notice that this algorithm is the same as Algorithm 3.3 when $u_c = 0$, but with different filters.

Remark 2. The parameter r_0 can either be estimated or be assumed to be known. In the latter case it is convenient to write R^* as

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

and use

$$\varepsilon(t) = y(t) - r_0u_f(t-d) - \varphi^T(t-d)\hat{\theta}(t-1)$$
$$\varphi^T(t) = \frac{Q^*(q^{-1})}{P^*(q^{-1})}\begin{pmatrix} r_0u(t-1) & \ldots & r_0u(t-k) & y(t) & \ldots & y(t-l)\end{pmatrix}$$
$$\theta^T = \begin{pmatrix} r_1' & \ldots & r_k' & s_0 & \ldots & s_l\end{pmatrix}$$

Asymptotic Properties

The models of Eqs. (4.21) and (4.22) can be interpreted as reparameterizations of the process model of Eq. (4.1) in terms of the controller parameters. They are identical to the model of Eq. (4.23) in Algorithm 4.1 if the filter Q^*/P^* is chosen to be $1/C^*$ and B^{*-}/C^*, respectively. The regression vector is then uncorrelated with the errors, and the least-squares estimate can be expected to converge to the true parameters. The C^* and B^{-*} polynomials are not known, however. The surprising result is that the algorithm also self-tunes to the correct controller even when the filter is not correct. This property inspired the authors of this book to introduce the term "self-tuning." The following result shows that the correct controller parameters are equilibrium values for Algorithm 4.1 for an incorrect choice of Q^*/P^* also. A more detailed analysis of stability and convergence is found in Chapter 6.

THEOREM 4.1 Asymptotic properties 1

Let Algorithm 4.1 with $Q^*/P^* = 1$ be used with a least-squares estimator. The parameter $r_0 = b_0$ can be either fixed or estimated. Assume that the regression vectors are bounded, and assume that the parameter estimates converge. The closed-loop system obtained in the limit is then characterized by

$$\begin{aligned} \overline{y(t+\tau)y(t)} &= 0 \qquad \tau = d, d+1, \ldots, d+l \\ \overline{y(t+\tau)u(t)} &= 0 \qquad \tau = d, d+1, \ldots, d+k \end{aligned} \tag{4.25}$$

where the overbar indicates a time average. Also, k and l are the degrees of the polynomials R^* and S^*, respectively.

Proof: The model of Eq. (4.23) can be written as

$$y(t+d) = \varphi^T(t)\theta + \varepsilon(t+d)$$

and the control law becomes

$$\varphi^T(t)\hat{\theta}(t+d) = 0 \tag{4.26}$$

At an equilibrium the estimated parameters $\hat{\theta}$ are constant. Furthermore, they satisfy the normal equations (2.5), which in this case are written as

$$\frac{1}{t}\sum_{k=1}^{t} \varphi(k)y(k+d) = \frac{1}{t}\sum_{k=1}^{t} \varphi(k)\varphi^T(k)\hat{\theta}(t+d)$$

By using the control law it follows from Eq. (4.26) that

$$\lim_{t\to\infty} \frac{1}{t}\sum_{k=1}^{t} \varphi(k)y(k+d) = \lim_{t\to\infty} \frac{1}{t}\sum_{k=1}^{t} \varphi(k)\varphi^T(k)\left(\hat{\theta}(t+d) - \hat{\theta}(k+d)\right)$$

If the estimate $\hat{\theta}(t)$ converges as $t \to \infty$ and the regression vector $\varphi(k)$ is bounded, the right-hand side goes to zero. Equation (4.25) now follows from $Q^*/P^* = 1$ and the definition of the regression vector in Algorithm 4.1. □

Stronger statements can be made if more is assumed about the system to be controlled.

THEOREM 4.2 Asymptotic properties 2

Assume that Algorithm 4.1 with least-squares estimation is applied to Eq. (4.1) and that

$$\min(k, l) \geq n - 1 \tag{4.27}$$

If the asymptotic estimates of R^* and S^* are relatively prime, the equilibrium solution is such that

$$\overline{y(t+\tau)y(t)} = 0 \qquad \tau = d, d+1, \ldots \tag{4.28}$$

that is, the output is a moving-average process of order $d-1$.

Proof: The closed-loop system is described by

$$\begin{aligned} R^*u(t) &= -S^*y(t) \\ A^*y(t) &= B^*u(t-d_0) + C^*e(t) \end{aligned}$$

Hence

$$\begin{aligned} (A^*R^* + q^{-d_0}B^*S^*)y &= R^*C^*e \\ (A^*R^* + q^{-d_0}B^*S^*)u &= -S^*C^*e \end{aligned}$$

Introduce the signal w defined by

$$(A^*R^* + q^{-d_0}B^*S^*)w = C^*e \tag{4.29}$$

Hence

$$y = R^*w \qquad \text{and} \qquad u = -S^*w \tag{4.30}$$

The condition of Eq. (4.25) then implies that

$$\begin{aligned} \overline{R^*w(t)y(t+\tau)} &= 0 \qquad \tau = d, d+1, \ldots, d+l \\ \overline{S^*w(t)y(t+\tau)} &= 0 \qquad \tau = d, d+1, \ldots, d+k \end{aligned}$$

If we introduce

$$C_{wy}(\tau) = \overline{w(t)y(t+\tau)}$$

the preceding equations can be written as

$$\begin{pmatrix} r_0 & r_1 & r_2 & \ldots & r_k & 0 & \ldots & 0 \\ 0 & r_0 & r_1 & r_2 & \ldots & r_k & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & & \ddots & \\ 0 & \ldots & 0 & r_0 & r_1 & r_2 & \ldots & r_k \\ s_0 & s_1 & s_2 & \ldots & s_l & 0 & \ldots & 0 \\ 0 & s_0 & s_1 & s_2 & \ldots & s_l & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & & \ddots & \\ 0 & \ldots & 0 & s_0 & s_1 & s_2 & \ldots & s_l \end{pmatrix} \begin{pmatrix} C_{wy}(d) \\ \vdots \\ C_{wy}(d+k-l) \end{pmatrix} = 0$$

Since the Sylvester matrix on the left is nonsingular when R^* and S^* are relatively prime (compare Section 11.4), it follows that

$$C_{wy}(\tau) = 0 \qquad \tau = d, d+1, \dots, d+k+l$$

The covariance function satisfies the equation

$$F^*(q^{-1})C_{wy}(\tau) = 0 \qquad \tau \ge 0$$

The system of Eq. (4.29) has the order

$$n + k = n + \max(k, l)$$

If

$$k + l + 1 \ge n + \max(k, l)$$

or, equivalently,

$$\min(k, l) \ge n - 1$$

it follows that

$$C_{wy}(\tau) = 0 \qquad \tau = d, d+1, \dots$$

It also follows from Eq. (4.30) that

$$C_y(\tau) = 0 \qquad \tau = d, d+1, \dots$$

which completes the proof. □

Remark 1. The algorithm thus drives the correlation of the output to zero starting at lag $\tau = d$. It follows from Theorem 4.1 that the correlations at lags d, $d+1, \dots, d+l$ will always be zero at equilibrium. If there are enough parameters in the controller, the covariance of the output will be zero for all higher lags. Notice that the condition of Eq. (4.28) is easily checked by monitoring the covariances of the output.

Remark 2. It is possible to influence cancellation of the process zeros simply by choosing the integer d. With $d = d_0$ a controller that cancels all zeros is obtained. With $d = n$ the controller will not cancel any process zeros. □

Theorems 4.1 and 4.2 imply that if the estimates converge, and if there are sufficiently many parameters in the controller, then Algorithm 4.1 will converge to the moving-average controller.

Examples

The properties of the minimum-variance and moving-average self-tuners are illustrated with two examples.

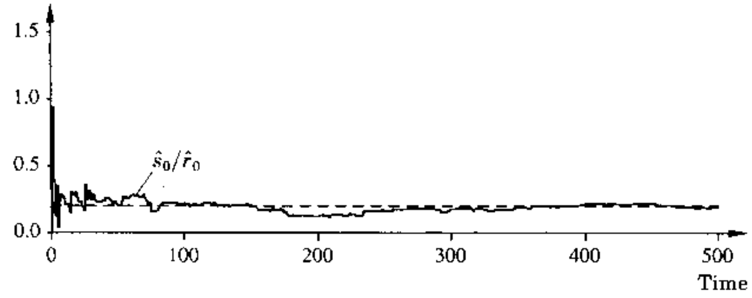

Figure 4.6 The parameter $\hat{s}_0/\hat{r}_0$ in the controller, when the process in Example 4.5 is controlled by using the direct minimum-variance self-tuning controller.

EXAMPLE 4.5 Direct minimum-variance self-tuning regulator

Consider the same process as in Example 4.4. The process model of Eq. (4.23) is now

$$y(t+1) = r_0 u(t) + s_0 y(t) + \varepsilon(t+1)$$

It is assumed that r_0 is fixed to the value $\hat{r}_0 = 1$. Notice that this is different from the true value, which is 3. The parameter s_0 is estimated by using the least-squares method. The control law becomes

$$u(t) = -\frac{\hat{s}_0}{\hat{r}_0} y(t)$$

Figure 4.6 shows $\hat{s}_0/\hat{r}_0$, which is seen to converge rapidly to a value corresponding to the value of the optimal minimum-variance controller, even if $\hat{r}_0$ is not equal to its true value. This is also seen in Fig. 4.7, which shows the loss function when the self-tuner and the optimal minimum-variance controller are used. Compare Figs. 4.3 and 4.5. □

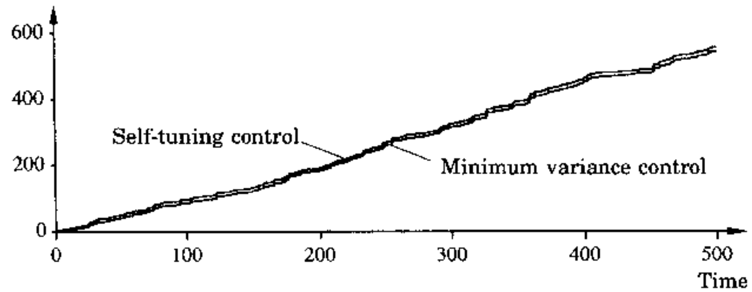

Figure 4.7 The loss function when the direct self-tuning regulator and the optimal minimum-variance controller are used on the system in Example 4.5.

EXAMPLE 4.6 MA control of a nonminimum-phase system

Consider an integrator with a time delay τ. For the sampling period $h > \tau$ the system is described by

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

where

$$b = \frac{\tau}{h-\tau} \quad \text{and} \quad d_0 = 1$$

The noise is assumed to be characterized by

$$C(q) = q(q+c) \qquad |c| < 1$$

The sampled-data system is nonminimum-phase if $\tau > h/2$. This implies that the basic minimum-variance self-tuner can be used only if $\tau < h/2$. Let the controller have the structure

$$u(t) = -\hat{s}_0(t)y(t) - \hat{r}_1(t)u(t-1)$$

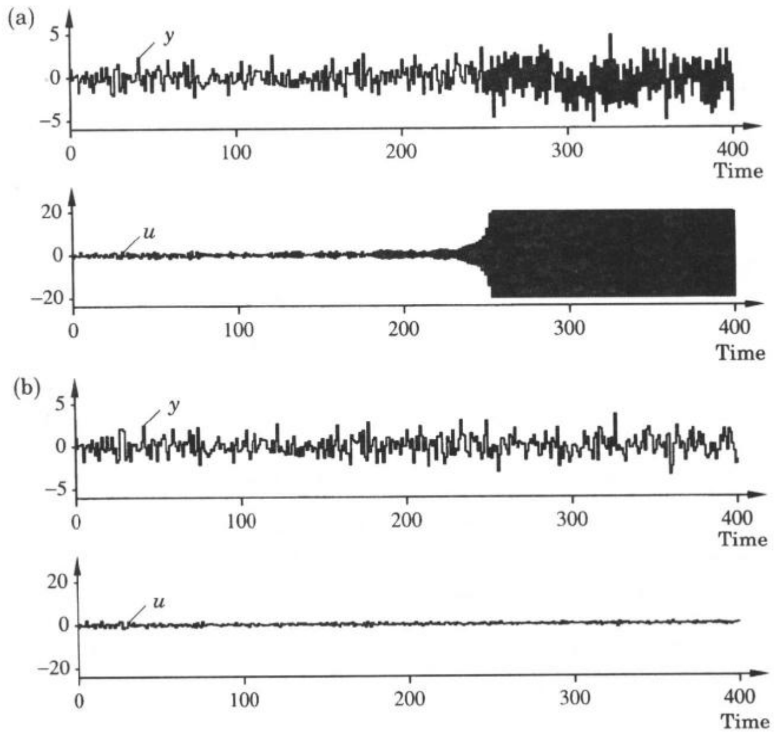

Figure 4.8 Simulation of the self-tuning algorithm on the integrator with time delay in Example 4.6. At $t = 100$ the delay is changed from 0.4 to 0.6. (a) $d = 1$; (b) $d = 2$.

Simulations of the system are shown in Fig. 4.8 for $h = 1$ and $c = -0.8$. The time delay is initially 0.4 and is increased to 0.6 at time $t = 100$, at which time the sampled-data system gets a zero outside the unit circle. Figure 4.8(a) shows the results obtained with $d = 1$, the minimum-variance structure. The parameters first converge toward the minimum-variance controller. At $t = 100$ the sampled-data system gets a zero outside the unit circle. The self-tuning regulator then tries to cancel the zero, and the closed-loop system becomes unstable after some time. It does not become unstable exactly at $t = 100$ because it takes a while for the controller parameters to change. The control signal is limited to ± 20, which explains why the signals do not grow exponentially. The forgetting factor is $\lambda = 0.99$. Figure 4.8(b) shows the results for the algorithm with $d = 2$. The moving-average controller is a stable equilibrium for both $\tau = 0.4$ and $\tau = 0.6$. There will be a shift in the parameter values when the delay is changed, but the closed-loop system is stable.

The controller that gives the smallest attainable variance of the output gives the standard deviations 1.000 and 1.004 when $\tau = 0.4$ and 0.6, respectively, while the moving-average controller gives the standard deviations 1.003 and 1.007 when $\tau = 0.4$ and 0.6, respectively. Degradation in the performance when the moving-average controller is used in this example is thus minor. □

4.4 UNIFICATION OF DIRECT SELF-TUNING REGULATORS

The moving-average self-tuner is attractive because of its simplicity. It is easy to explain intuitively how the algorithm works, and the algorithm is easy to implement. This has led to great interest in the algorithm. The algorithm can be explained as follows: Determine the structure of a predictor that can be used to predict the output d steps ahead. The parameters of the predictor are estimated in real time. On the basis of the estimated parameters the control signal is determined such that the predicted output of the process is equal to the reference value. The algorithm has been analyzed extensively. The closed-loop bandwidth depends critically on the sampling period h and the prediction horizon d, so both must be chosen with care. The algorithm may result in a controller in which process zeros are canceled; the cancellations depend on the choice of prediction horizon. Many variants of the algorithm have been suggested. A number of these can be described in a unified framework, as we will demonstrate.

Consider the model of Eq. (4.1), and introduce the filtered output

$$y_f(t) = \frac{Q^*(q^{-1})}{P^*(q^{-1})}\, y(t)$$

where Q^* and P^* are stable polynomials. The filtered output satisfies the equation

$$A^*(q^{-1})P^*(q^{-1})y_f(t) = B^*(q^{-1})Q^*(q^{-1})u(t-d_0) + C^*(q^{-1})Q^*(q^{-1})e(t)$$

Introduce the identity

$$C^*(q^{-1})Q^*(q^{-1}) = A^*(q^{-1})P^*(q^{-1})R_1^*(q^{-1}) + q^{-d_0}S^*(q^{-1})$$

Then

$$y_f(t+d_0) = \frac{1}{C^*Q^*}\left(S^*y_f(t) + B^*Q^*R_1^*u(t)\right) + R_1^*e(t+d_0)$$

Introducing

$$y_f'(t) = \frac{1}{Q^*(q^{-1})}y_f(t) = \frac{1}{P^*(q^{-1})}y(t)$$

gives the model

$$y_f(t+d_0) = \frac{1}{C^*}\left(S^*y_f'(t) + B^*R_1^*u(t)\right) + R_1^*e(t+d_0) \tag{4.31}$$

By analogy with Eq. (4.21) this model structure could be used with Algorithm 4.1 to derive a self-tuning regulator for minimization of the variance of y_f. This reparameterized model now suggests the following generalized self-tuning algorithm.

ALGORITHM 4.2 Generalized direct self-tuning algorithm

Data: Given the prediction horizon, d, the order of the controller, $\deg R^*$ and $\deg S^*$, the stable observer polynomial, A_o^*, and the stable polynomials Q^* and P^*, define the filtered signals

$$y_f(t) = \frac{Q^*}{P^*}y(t) \qquad y_f'(t) = \frac{1}{P^*}y(t)$$

Step 1: Estimate the coefficients of the polynomials R^* and S^* of the model

$$y_f(t+d) = \frac{R^*}{A_o^*}u(t) + \frac{S^*}{A_o^*}y_f'(t) + \varepsilon(t+d) \tag{4.32}$$

using the least-squares method.

Step 2: Calculate the control signal from

$$u(t) = -\frac{S^*}{R^*}y_f'(t)$$

with R^* and S^* given by the estimates obtained in Step 1.

Repeat Steps 1 and 2 at each sampling period. □

From Eq. (4.31) and Theorems 4.1 and 4.2 it follows that if the estimates converge, then the closed-loop system will be

$$y_f(t) = R_1^*e(t)$$

or

$$y(t) = \frac{P^*R_1^*}{Q^*}e(t) \tag{4.33}$$

where R_1^* is given by the identity

$$C^*Q^* = A^*P^*R_1^* + q^{-d}B^{-*}S^* \tag{4.34}$$

and the control signal is given by

$$u(t) = -\frac{S^*}{R^*}y_f'(t) = -\frac{S^*}{R^*P^*}y(t) \tag{4.35}$$

where $R^* = B^{+*}R_1^*$. The closed-loop poles will thus be influenced by Q^*, and additional zeros can be introduced through P^*. The introduction of the filter Q^*/P^* gives what is sometimes called a *detuned minimum-variance* algorithm.

Algorithm 4.2 is essentially the same as Algorithm 4.1 applied to filtered signals. The filter Q^*/P^* and the prediction horizon will determine the pulse transfer operator of the closed-loop system. The optimal observer polynomial is C^*, which is unknown. Instead, an approximation A_o^* is used. The observer polynomial A_o^* will determine the convergence properties. This will not influence the asymptotic properties as long as the filter Q^*/P^* and its inverse are stable.

Minimum-variance control may result in large control signals. One way to decrease the variation of the control signal is to generalize the loss function such that it also contains a penalty on the control signal. Linear quadratic controllers are of this type; a minor drawback with linear quadratic self-tuning regulators is the computational burden. One way to simplify the problems is to use a loss function of the form

$$E\left\{\left(P^*(q^{-1})y(t+d_0)\right)^2 + \left(Q^*(q^{-1})u(t)\right)^2 \,\middle|\, \mathcal{Y}_t\right\}$$

where

$$\mathcal{Y}_t = \{y(t), y(t-1), \ldots, y(0), u(t), u(t-1), \ldots, u(0)\}$$

that is, the data available at time t. The resulting controller is sometimes called a *generalized minimum-variance controller*. This controller can be interpreted in the same framework as above. To illustrate this, assume that $P^* = 1$ and that $Q^* = \sqrt{\rho}$. This gives the loss function

$$E\left\{y^2(t+d_0) + \rho u^2(t) \,\middle|\, \mathcal{Y}_t\right\} \tag{4.36}$$

Notice that the loss function depends only on the output y at time $t+d_0$, that is, at only one time instant. Loss functions of the form (4.36) are sometimes called one-stage loss functions.

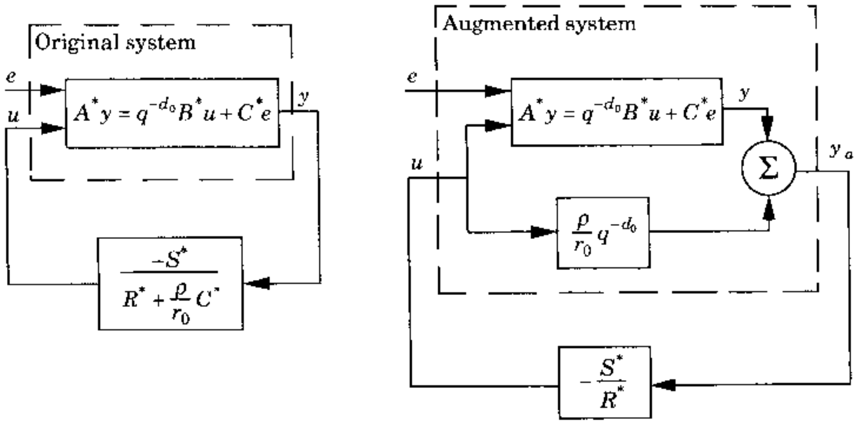

Figure 4.9 Equivalent systems.

Assume that the process is governed by Eq. (4.1). By using the representation of the process dynamics given by Eq. (4.21) it can be shown that the control law that minimizes Eq. (4.36) is

$$\left(R^* + \frac{\rho}{r_0} C^*\right) u(t) = -S^* y(t) \tag{4.37}$$

where

$$R^* = R_1^* B^*$$

and R^* and S^* are given by Eq. (4.16).

By using the same idea it is possible to construct a new system, which has Eq. (4.37) as its minimum-variance controller. Augment the original system with a parallel connection with the pulse transfer operator $\rho q^{-d_0}/r_0$ (see Fig. 4.9). This is in fact a standard technique to obtain an equivalent controller with a bounded gain. The input-output relation of the augmented system is

$$A^* y_a(t) = \left(B^* + \frac{\rho}{r_0} A^*\right) u(t - d_0) + C^* e(t)$$

The minimum-variance control law for this system is given by

$$R_1^* \left(B^* + \frac{\rho}{r_0} A^*\right) u(t) = -S^* y_a(t) \tag{4.38}$$

where R_1^* and S^* satisfy Eq. (4.16). It follows from Fig. 4.9 that

$$y_a(t) = y(t) + \frac{\rho}{r_0} q^{-d_0} u(t)$$

Then Eq. (4.38) can be written as

$$\left(R_1^* B^* + \frac{\rho}{r_0} A^* R_1^*\right) u(t) = -S^* \left(y(t) + \frac{\rho}{r_0} q^{-d_0} u(t)\right)$$

or

$$\left(R_1^* B^* + \frac{\rho}{r_0} (A^* R_1^* + q^{-d_0} S^*)\right) u(t) = -S^* y(t)$$

Equation (4.16) gives

$$\left(R_1^* B^* + \frac{\rho}{r_0} C^*\right) u(t) = -S^* y(t)$$

which is identical to Eq. (4.37). Notice that with the control law of Eq. (4.38) the canceled factor is not B^* but $B^* + \rho A^*/r_0$. This implies that problems can be expected when the system is nonminimum-phase and close to the stability boundary.

In the generalized minimum-variance control algorithm it is assumed that $C^*(q^{-1}) = 1$. The algorithm can thus be obtained simply by adding a parallel path to the original system and applying an ordinary self-tuning regulator based on minimum-variance control to the augmented system. The control gain is adjusted simply by changing the parameter ρ of the parallel path.

The preceding analysis shows that Algorithm 4.2 is very flexible. It can be used for many different types of specifications, not only for minimum-variance control. This is very important for the implementation of self-tuning regulators.

Self-tuning Feedforward Control

Feedforward control is a very useful way to reduce the influence of known disturbances. Examples of measurable disturbances can be temperatures and concentrations in incoming product streams in chemical processes, outdoor temperature in climate control systems, and thickness of the paper in paper machines. Command signals can also be interpreted as a measurable disturbance. The controller in Eq. (3.2) can be interpreted as feedforward from the command signal. To use feedforward, it is necessary to know the dynamics of the process. It is, however, also possible to establish self-tuning feedforward compensation. One way to do this is to postulate a model structure of the form

$$y(t + d) = R^* u(t) + S^* y(t) + T^* v(t) + \varepsilon(t + d)$$

where $v(t)$ is the measurable disturbance acting on the system. The signal v could also be the reference value. The polynomials R^*, S^*, and T^* are estimated in the usual way, and the control law is chosen to be

$$u(t) = -\frac{S^*}{R^*} y(t) - \frac{T^*}{R^*} v(t)$$

Self-tuning feedforward control has been used successfully in many industrial applications.

Examples

The behavior of Algorithm 4.2 is illustrated through two examples.

EXAMPLE 4.7 Effect of filtering

Consider the process

$$y(t) + ay(t-1) = bu(t-1) + e(t) + ce(t-1)$$

where $a = -0.9$, $b = 3$, and $c = -0.3$, which is the same process as in Examples 4.4 and 4.5. Let the filter be

$$\frac{Q^*}{P^*} = \frac{1 + q_1 q^{-1}}{1 + p_1 q^{-1}}$$

The identity of Eq. (4.34) gives the solution

$$s_0 = c + q_1 - a - p_1$$
$$s_1 = cq_1 - ap_1$$

The control law is given by Eq. (4.35), with

$$R_1^* P^* B^{+*} = b(1 + p_1 q^{-1})$$

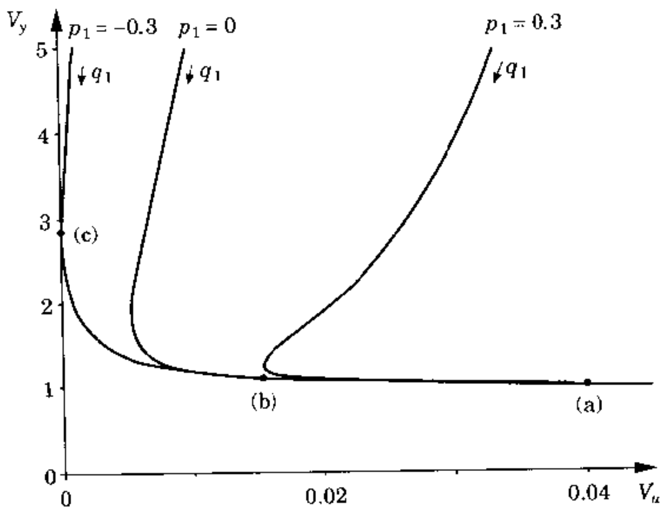

Figure 4.10 The output variance, V_y, and input variance, V_u, as functions of q_1 of the system in Example 4.7 when $p_1 = -0.3$, 0, and 0.3. Three different cases are indicated by dots: (a) $p_1 = q_1 = 0$; (b) $p_1 = 0$, $q_1 = -0.3$; (c) $p_1 = -0.3$, $q_1 = -0.9$.

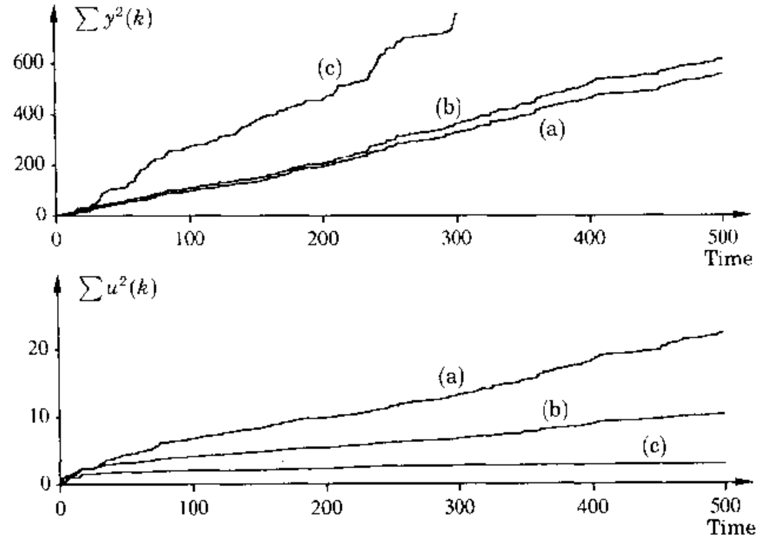

Figure 4.11 Simulation of the generalized self-tuning algorithm on the system in Example 4.7 when (a) $p_1 = q_1 = 0$ (minimum-variance control); (b) $p_1 = 0$, $q_1 = -0.3$; (c) $p_1 = -0.3$, $q_1 = -0.9$ (open-loop system).

The closed-loop system becomes

$$y(t) = \frac{1 + p_1 q^{-1}}{1 + q_1 q^{-1}} e(t)$$

$$u(t) = -\frac{s_0 + s_1 q^{-1}}{b(1 + q_1 q^{-1})} e(t)$$

There are many different ways to choose the filter Q^*/P^*. In principle it should be a phase-advance network. This implies that the closed-loop system given by Eq. (4.33) will be low-pass filtered. Figure 4.10 shows how the output and input variances change with q_1 for some values of p_1. Case (a) in Fig. 4.10 corresponds to minimum-variance control. In case (b) the output variance is increased by 10%, and the input variance is reduced by about 60% compared with the minimum-variance case. In case (c) the input variance is zero; that is, the system is open-loop. Figure 4.11 shows the accumulated losses for the input and the output when the generalized self-tuning algorithm is used. Cases (a), (b), and (c) are the same as in Fig. 4.10. □

EXAMPLE 4.8 Generalized minimum variance self-tuning controller

The self-tuning controller that minimizes Eq. (4.36) will now be used to control the same system as in the previous example. The controller in Eq. (4.37), with

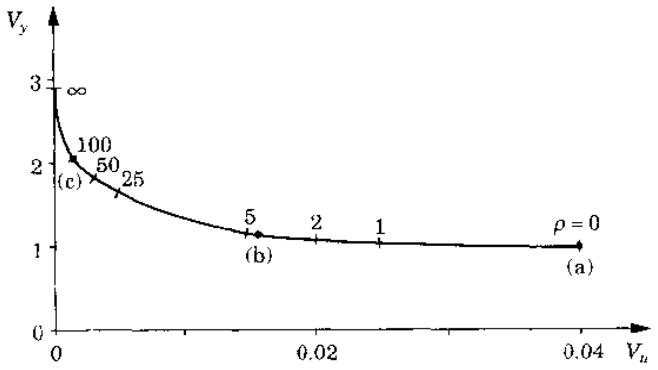

Figure 4.12 The output variance V_y as a function of the input variance V_u in Example 4.8 for different values of ρ: (a) $\rho = 0$; (b) $\rho = 4$; (c) $\rho = 100$.

R^* and S^* given by Eq. (4.16), is

$$u(t) = -\frac{c - a}{b + \rho(1 + aq^{-1})} y_a(t)$$

Figure 4.12 shows the output variance as a function of the input variance for different values of ρ. The curve has the same gross behavior as shown in Fig. 4.10. However, the parameter ρ may be easier to choose than the filter in Example 4.7. Figure 4.13 shows the accumulated losses of the output and the input for different values of ρ when the self-tuner in Algorithm 4.1 is used on the augmented system shown in Fig. 4.9. Compare Fig. 4.11. □

Summary

There are many ways to make direct self-tuning regulators with good properties. The amount of computation is moderate, since the design calculations are eliminated. It has been shown that the generalized direct self-tuning algorithm, Algorithm 4.2, is very flexible. By using the filter Q^*/P^* and the prediction horizon, it is possible to determine the behavior of the closed-loop system. It is possible to choose, for instance, moving-average control, generalized minimum-variance control, or pole-zero placement control.

The observer polynomial does not influence the asymptotic properties. It will instead influence the transient properties and can be used to improve the convergence properties of the algorithm. The robustness and sensitivity of the algorithm are also influenced by the filter Q^*/P^*.

For simplicity, Algorithm 4.2 has been derived for the regulator case, in which the reference value is equal to zero. It is easy to modify the algorithm

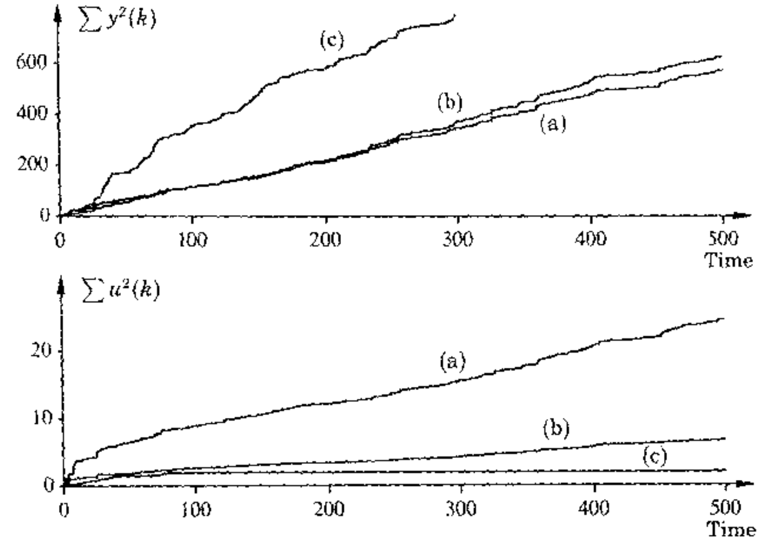

Figure 4.13 Simulation of the generalized minimum variance self-tuning algorithm on the system in Example 4.8 when (a) $\rho = 0$ (minimum-variance control); (b) $\rho = 4$; (c) $\rho = 100$ ("almost" open-loop control).

such that the output follows a reference trajectory; some ideas are suggested in the problems at the end of this chapter and in Section 11.3.

4.5 LINEAR QUADRATIC STR

The linear quadratic design procedure can also be used as the design method in a self-tuning regulator. Consider the process model

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \tag{4.39}$$

and the steady-state loss function

$$J_{yu} = E\left\{(y(t) - y_m(t))^2 + \rho u^2(t)\right\} \tag{4.40}$$

The optimal feedback law that minimizes Eq. (4.40) for the system of Eq. (4.39) is given by the following theorem.

THEOREM 4.3 LQG control

Consider the system in Eq. (4.39). Let the monic polynomials $A(q)$ and $C(q)$ have degree n. Assume that $C(q)$ has all its zeros inside the unit disc, and assume that there is no nontrivial polynomial that divides $A(q)$, $B(q)$, and $C(q)$. Let $A_2(q)$ be the greatest common divisor of $A(q)$ and $B(q)$, let $A_2^+(q)$ of degree l be the factor of $A_2(q)$ with all its zeros inside the unit disc, and let $A_2^-(q)$ of degree m be the factor of $A(q)$ that has all its zeros outside the unit disc or on the unit circle.

The admissible control law that minimizes Eq. (4.40) with $\rho > 0$ is then given by

$$R(q)u(t) = -S(q)y(t) + T(q)y_m(t) \tag{4.41}$$

where R and S are of degree $n + m$

$$\begin{aligned} R(q) &= A_2^-(q)\tilde{R}(q) \\ S(q) &= z^m\tilde{S}(q) \end{aligned} \tag{4.42}$$

and $\tilde{R}(q)$ and $\tilde{S}(q)$ satisfy the Diophantine equation

$$A_1(q)A_2^-(q)\tilde{R}(q) + q^m B_1(q)\tilde{S}(q) = P_1(q)C(q) \tag{4.43}$$

with $\deg \tilde{R}(q) = \deg \tilde{S}(q) = n$ and $\tilde{S}(0) = 0$. Furthermore,

$$\begin{aligned} A(q) &= A_1(q)A_2(q) \\ B(q) &= B_1(q)A_2(q) \\ \tilde{B}(q) &= B_1(q)A_2^+(q) \end{aligned}$$

The polynomial $P(q)$ is given by

$$P(q) = A_2^+(q)P_1(q) \tag{4.44}$$

where $P_1(q)$ is the solution of the spectral factorization problem

$$rP_1(q)P_1(q^{-1}) = \rho A_1(q)A_2^-(q)A_1(q^{-1})A_2^-(q^{-1}) + B_1(q)B_1(q^{-1}) \tag{4.45}$$

with $\deg P_1(q) = \deg A_1(q) + \deg A_2^-(q)$. The polynomial $T(q)$ is given by

$$T(q) = t_0 q^m C(q)$$

where

$$t_0 = P_1(1)/B_1(1)$$ □

A proof of the theorem is found in Åström and Wittenmark (1990).

Remark. By using Eqs. (4.42) the identity (4.43) can be written as

$$A(q)R(q) + B(q)S(q) = A_2(q)P_1(q)C(q)$$

The LQG solution can thus be interpreted as a pole-placement controller, where the poles are positioned at the zeros of A_2, P_1, and C. The controller also has the property that A_2^- divides R. This is an example of the *internal model principle*. Using the internal model principle implies that a model of the disturbance is included in the controller. □

To solve the design problem, it is necessary to solve the spectral factorization problem of Eq. (4.45) and to solve the Diophantine equation Eq. (4.43). The solution to the LQG problem given by Theorem 4.3 is closely related to the pole placement design problem. The solution to the spectral factorization problem gives the desired closed-loop poles. The second part of the algorithm can be interpreted as a pole placement problem.

An alternative solution to the design problem is to use a state space formulation. The process model of Eq. (4.39) can be written in state space form as

$$\begin{aligned} x(t+1) &= \bar{A}x(t) + \bar{B}u(t) + \bar{K}e(t) \\ y(t) &= \bar{C}x(t) + e(t) \end{aligned} \tag{4.46}$$

where the matrices $\bar{A}$, $\bar{B}$, $\bar{C}$, and $\bar{K}$ are given in the canonical form

$$\bar{A} = \begin{pmatrix} -a_1 & 1 & 0 & \dots & 0 \\ \vdots & & \ddots & & \\ -a_{n-1} & 0 & & \dots & 1 \\ -a_n & 0 & & \dots & 0 \end{pmatrix}$$

$$\bar{B} = \begin{pmatrix} 0 & \dots & 0 & b_0 & \dots & b_m \end{pmatrix}^T$$

$$\bar{C} = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}$$

$$\bar{K} = \begin{pmatrix} c_1 - a_1 & \dots & c_n - a_n \end{pmatrix}^T$$

where $m = n - d_0$. The model in Eq. (4.46) is called the *innovation model*, and $\bar{K}$ is the optimal steady-state gain in the Kalman filter, that is, $\hat{x}(t+1|t) = x(t+1)$. It is also possible to derive the filter for $\hat{x}(t|t)$, which is given by

$$\hat{x}(t|t) = (qI - \bar{A} + \bar{K}\bar{C})^{-1}\left(\bar{B}u(t) + \bar{K}y(t)\right)$$

By using the definitions of $\bar{A}$, $\bar{K}$, and $\bar{C}$ it is easily seen that $\det(qI - \bar{A} + \bar{K}\bar{C}) = C(q)$. That is, the optimal observer polynomial is equal to $C(q)$.

Introduce the loss function

$$J_x = E\left\{\sum_{t=1}^{N} x^T(t)Q_1x(t) + \rho u^2(t) + x^T(N)Q_0x(N)\right\} \tag{4.47}$$

The optimal controller is given by

$$u(t) = -L(t)\hat{x}(t|t) \tag{4.48}$$

where $L(t)$ is a time-varying feedback gain given through a Riccati equation

$$\begin{aligned} S(t) &= \left(\bar{A} - \bar{B}L(t)\right)^T S(t+1)\left(\bar{A} - \bar{B}L(t)\right) + Q_1 + \rho L^T(t)L(t) \\ L(t) &= \left(\rho + \bar{B}^T S(t+1)\bar{B}\right)^{-1}\bar{B}^T S(t+1)\bar{A} \end{aligned} \tag{4.49}$$

with $S(N) = Q_0$. The limiting controller

$$\bar{L} = \lim_{t\to\infty} L(t)$$

is such that the closed-loop characteristic equation is

$$P(q) = \det(q - \bar{A} + \bar{B}\bar{L}) = 0$$

where $P(q)$ is the same as in Eq. (4.44).

The two solutions to the LQG control problem suggest two ways to construct indirect linear quadratic self-tuning regulators. In both algorithms it is first necessary to estimate the A, B, and C polynomials in the process model of Eq. (4.39). This can be done by using the recursive maximum-likelihood method or the extended least-squares method. This leads to the following algorithm.

ALGORITHM 4.3 Indirect LQG-STR based on spectral factorization

Data: Given specifications in the form of the parameter ρ in the loss function of Eq. (4.40) and the order of the system.

Step 1: Estimate the coefficients of the polynomials A, B, and C in Eq. (4.39).

Step 2: Replace A, B, and C with the estimates obtained in Step 1 and solve the spectral factorization problem of Eq. (4.45) to obtain $P(q)$.

Step 3: Solve the Diophantine equation of Eq. (4.43).

Step 4: Calculate the control signal from Eq. (4.41).

Repeat Steps 1, 2, 3, and 4 at each sampling period. □

The state space formulation gives the following algorithm.

ALGORITHM 4.4 Indirect LQG-STR based on the Riccati equation

Data: Given specifications in the form of the parameters Q_0, Q_1, and ρ in the loss function of Eq. (4.47) and the order of the system.

Step 1: Estimate the coefficients of the polynomials A, B, and C in Eq. (4.39).

Step 2: Replace A, B, and C with the estimates obtained in Step 1 and solve the algebraic Riccati equation or iterate Eqs. (4.49) to obtain $\bar{L}$.

Step 3: Calculate the control signal from Eq. (4.48).

Repeat Steps 1, 2, and 3 at each sampling period. □

Notice that if $Q_1 = \bar{C}^T\bar{C}$, the steady-state solution to Eqs. (4.49) will give the same result as the minimization of Eq. (4.40). Algorithms 4.3 and 4.4 are indirect algorithms that are able to handle nonminimum-phase systems and varying time delays. The computations are more extensive for these algorithms than for the simple self-tuning regulators discussed above.

Solution of the spectral factorization or the Riccati equation is the major computation in an LQG self-tuner. These calculations can be made in many different ways. The Riccati equation can be solved by using an eigenvalue method or by some iterative method. The iterative methods will in general lead to shorter code. In general the Riccati equation is iterated several steps. To guarantee that the calculations can be done in a prescribed sampling interval, it is necessary to truncate the iterations; it is important that a reasonable result be obtained when the iteration is truncated. For instance, the polynomial P in the spectral factorization must be stable. This is guaranteed for some algorithms. In some algorithms it is suggested that the Riccati equation be iterated only one step at each sampling.

4.6 ADAPTIVE PREDICTIVE CONTROL

Algorithm 4.1 is one way to make a controller with a variable prediction horizon. The underlying control problem is the moving-average controller. The moving-average controller may also be used for nonminimum-phase systems, as was illustrated in Section 4.3.

In using the minimum-variance controller or the moving-average controller the output is predicted only at *one* future time. The prediction horizon d is then a design parameter. The predicted output can also be computed for different prediction horizons and then used in a loss function. Several ways to achieve predictive control have been suggested in the literature; we now discuss and analyze some of these. The case with known parameters is first analyzed before the adaptive versions are discussed.

Predictive control algorithms are based on an assumed model of the process and on an assumed scenario for the future control signals. This gives a sequence of control signals. Only the first one is applied to the process, and a new sequence of control signals is calculated when a new measurement is obtained. This is called a *receding-horizon controller*. There are many variants of predictive control, for instance, *model predictive control*, *dynamic matrix control*, *generalized predictive control*, and *extended horizon control*. The methodology has been used extensively in chemical process control.

Output Prediction

One basic idea in the predictive control algorithms is to rewrite the process model to get an explicit expression for the output at a future time. Compare Eq. (4.22). Consider the deterministic process

$$A^*(q^{-1})y(t) = B^*(q^{-1})u(t - d_0) \tag{4.50}$$

and introduce the identity

$$1 = A^*(q^{-1})F_d^*(q^{-1}) + q^{-d}G_d^*(q^{-1}) \tag{4.51}$$

where

$$\deg F_d^* = d - 1$$
$$\deg G_d^* = n - 1$$

The subscript d is used to indicate that the prediction horizon is d steps. It is assumed that $d \geq d_0$. The polynomial identity of Eq. (4.51) can be used to predict the output d steps ahead. Hence

$$y(t + d) = A^*F_d^*y(t + d) + G_d^*y(t) = B^*F_d^*u(t + d - d_0) + G_d^*y(t) \tag{4.52}$$

Compare Eq. (4.12). Introduce

$$B^*(q^{-1})F_d^*(q^{-1}) = R_d^*(q^{-1}) + q^{-(d-d_0+1)}\bar{R}_d^*(q^{-1})$$

where

$$\deg R_d^* = d - d_0$$
$$\deg \bar{R}_d^* = n - 2$$

The coefficients of R_d^* are the first $d-d_0+1$ terms of the pulse response of the open-loop system. This can be seen as follows:

$$\begin{aligned}\frac{q^{-d_0}B^*}{A^*} &= q^{-d_0}B^*\left(F_d^* + q^{-d}\frac{G_d^*}{A^*}\right)\\ &= q^{-d_0}R_d^*(q^{-1}) + q^{-(d+1)}\bar{R}_d^*(q^{-1}) + \frac{B^*(q^{-1})G_d^*(q^{-1})}{A^*(q^{-1})}q^{-(d+d_0)}\end{aligned} \tag{4.53}$$

The powers of the last two terms are at least $-(d+1)$. It then follows that R_d^* is the first part of the pulse response, since $\deg R_d^* = d - d_0$.

Equation (4.52) can be written as

$$\begin{aligned}y(t+d) &= R_d^*(q^{-1})u(t+d-d_0) + \bar{R}_d^*(q^{-1})u(t-1) + G_d^*(q^{-1})y(t)\\ &= R_d^*(q^{-1})u(t+d-d_0) + \bar{y}_d(t)\end{aligned} \tag{4.54}$$

$R_d^*(q^{-1})u(t+d-d_0)$ depends on $u(t), \ldots, u(t+d-d_0)$, and $\bar{y}_d(t)$ is a function of $u(t-1)$, $u(t-2)$, …, and $y(t)$, $y(t-1)$, …. The variable $\bar{y}_d(t)$ can be interpreted as the constrained prediction of $y(t + d)$ under the assumption that $u(t)$ and future control signals are zero. The output at time $t+d$ thus depends on future control signals (if $d > d_0$), the control signal to be chosen, and old inputs and outputs. If $d > d_0$, it is necessary to make some assumptions about the future control signals. One possibility is to assume that the control signal will remain constant, that is, that

$$u(t) = u(t + 1) = \cdots = u(t + d - d_0) \tag{4.55}$$

Another way is to determine the control law that brings $y(t + d)$ to a desired value while minimizing the control effort over the prediction horizon, that is, to minimize

$$\sum_{k=t}^{t+d} u(k)^2 \tag{4.56}$$

A third way is to assume that the increment of the control signal will be zero after some time. This is used, for instance, in generalized predictive control (GPC), which is discussed below.

Constant Future Control

Consider Eq. (4.54) and assume that the predicted output is equal to the desired output, that is, $y(t + d) = y_m(t + d)$. If we assume that Eq. (4.55) holds, then $u(t)$ should be chosen such that

$$y_m(t + d) = \left(R_d^*(1) + q^{-1}\bar{R}_d^*(q^{-1})\right) u(t) + G_d^*(q^{-1})y(t)$$

This gives the control law

$$u(t) = \frac{y_m(t+d) - G_d^*(q^{-1})y(t)}{R_d^*(1) + \bar{R}_d^*(q^{-1})q^{-1}} \tag{4.57}$$

This control signal is then applied to the process. At the next sampling instant a new measurement is obtained, and the control law of Eq. (4.57) is used again. Note that the value of the control signal is changed rather than kept constant, as was assumed when Eq. (4.57) was derived. The receding-horizon control principle is thus used. Note that the control law is time-invariant, in contrast to a fixed-horizon linear quadratic controller.

We now analyze the closed-loop system when Eq. (4.57) is used to control the process of Eq. (4.50). It is now necessary to make the calculations in the forward shift operator, since poles at the origin may otherwise be overlooked. The identity of Eq. (4.51) can be written in the forward shift operator as

$$q^{n+d-1} = A(q)F_d(q) + G_d(q) \tag{4.58}$$

The characteristic polynomial of the closed-loop system is

$$P(q) = A(q)\left(q^{n-1}R_d(1) + \bar{R}_d(q)\right) + G_d(q)B(q) \tag{4.59}$$

where

$$\deg P = \deg A + n - 1 = 2n - 1$$

The design equation (Eq. 4.58) can now be used to rewrite $P(q)$:

$$\begin{aligned}B(q)q^{n+d-1} &= A(q)B(q)F_d(q) + G_d(q)B(q)\\ &= A(q)\left(q^{n-1}R_d(q) + \bar{R}_d(q)\right) + G_d(q)B(q)\end{aligned}$$

Hence

$$A(q)\bar{R}_d(q) + G_d(q)B(q) = B(q)q^{n+d-1} - A(q)q^{n-1}R_d(q)$$

which gives

$$P(q) = q^{n-1}A(q)R_d(1) + q^{n-1}\left(q^d B(q) - A(q)R_d(q)\right)$$

If the process is stable, it follows from Eq. (4.53) that the last term vanishes as $d \to \infty$. Thus

$$\lim_{d\to\infty} P(q) = q^{n-1}A(q)R_d(1) \qquad \text{if } A(z) \text{ is a stable polynomial}$$

The properties of the predictive control law are illustrated by an example.

EXAMPLE 4.9 Predictive control

Consider the process model

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

The identity of Eq. (4.58) gives

$$q^d = (q+a)(q^{d-1} + f_1 q^{d-2} + \cdots + f_{d-1}) + g_0$$

Hence

$$\begin{aligned} F(q) &= q^{d-1} - aq^{d-2} + a^2q^{d-3} + \cdots + (-a)^{d-1} \\ G(q) &= (-a)^d \\ R_d(q) &= bF(q) \\ \bar{R}_d(q) &= 0 \end{aligned}$$

and the control law becomes, when $y_m = 0$,

$$u(t) = -\frac{(-a)^d}{b(1 - a + \ldots + (-a)^{d-1})}\,y(t) = -\frac{(-a)^d(1+a)}{b(1-(-a)^d)}\,y(t)$$

The characteristic polynomial of the closed-loop system is

$$P(q) = q + a + \frac{(-a)^d(1+a)}{1-(-a)^d}$$

which has the pole

$$p_d = -\frac{a + (-a)^d}{1-(-a)^d}$$

If $a \le 0$ the location of the pole is given by

$$\begin{aligned} &0 \le p_d < -a && |a| \le 1 && \text{(stable open-loop system)} \\ &0 \le p_d < 1 && |a| > 1 && \text{(unstable open-loop system)} \end{aligned}$$

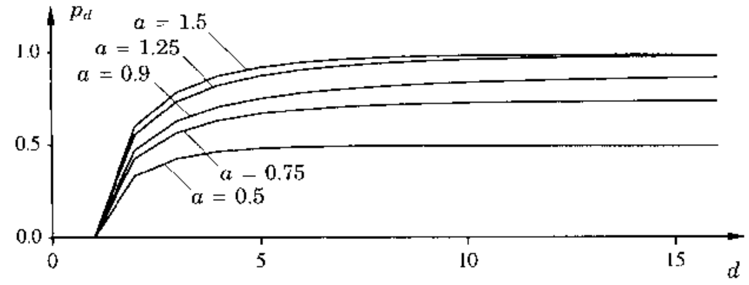

Figure 4.14 The closed-loop pole $p_d = (a^d - a)/(a^d - 1)$ as function of d for different values of a.

The closed-loop pole for different values of a and d is shown in Fig. 4.14. The example indicates that it can be sufficient to use a prediction horizon of five to ten samples. □

It is possible to generalize the result of Example 4.9 to higher-order systems. The conclusion is that the closed-loop response will be slow for slow or unstable systems when the prediction horizon increases. The restriction of Eq. (4.55) is then not very useful.

Minimum Control Effort

The control strategy that brings $y(t+d)$ to $y_m(t+d)$ while minimizing Eq. (4.56) will now be derived. Equation (4.54) is

$$\begin{aligned} y(t+d) &= R_d^*(q^{-1})u(t+d-d_0) + \bar{y}_d(t) \\ &= r_{d0}u(t+\nu) + \cdots + r_{d\nu}u(t) + \bar{y}_d(t) \end{aligned}$$

where $\nu = d - d_0$. The condition

$$y(t+d) = y_m(t+d) = \bar{y}_d(t) + R_d^*(q^{-1})u(t+d-d_0)$$

can be regarded as a constraint while minimizing Eq. (4.56). Introducing the Lagrangian multiplier λ gives the loss function

$$2J = u(t)^2 + \cdots + u(t+\nu)^2 + 2\lambda\left(y_m(t+d) - \bar{y}_d(t) - R_d^*(q^{-1})u(t+\nu)\right)$$

Equating the partial derivatives with respect to $u(t), \cdots, u(t+\nu)$ and λ to zero gives

$$\begin{aligned} u(t) &= \lambda r_{d\nu} \\ &\vdots \\ u(t+\nu) &= \lambda r_{d0} \\ y_m(t+d) - \bar{y}_d(t) &= r_{d0}u(t+\nu) + \cdots + r_{d\nu}u(t) \end{aligned}$$

This set of equations gives

$$u(t) = \frac{y_m(t+d) - \bar{y}_d(t)}{\mu}$$

where

$$\mu = \frac{\sum_{i=0}^{v} r_{di}^2}{r_{dv}}$$

Using the definition of $\bar{y}_d(t)$ gives

$$\mu u(t) = y_m(t+d) - \bar{R}_d^* u(t-1) - G_d^* y(t)$$

or

$$u(t) = \frac{y_m(t+d) - G_d^* y(t)}{\mu + q^{-1}\bar{R}_d^*} = \frac{y_m(t+d+n-1) - G_d(q)y(t)}{\mu q^{n-1} + \bar{R}_d(q)} \tag{4.60}$$

Using Eq. (4.60) and the model of Eq. (4.50) gives the closed-loop characteristic polynomial

$$P(q) = A(q)\left(q^{n-1}\mu + \bar{R}_d(q)\right) + G_d(q)B(q)$$

This is of the same form as Eq. (4.56), with $R_d(1)$ replaced by μ. This implies that the closed-loop poles approach the zeros of $q^{n-1}A(q)$ when $A(q)$ is stable and when $d \to \infty$. What will happen when the open-loop system is unstable? Consider the following example.

EXAMPLE 4.10 Minimum-effort control

Consider the same system as in Example 4.9. The minimum-effort controller is in this case given by

$$\mu = b\frac{1 + a^2 + \cdots + a^{2(d-1)}}{(-a)^{d-1}} = \frac{b(a^{2d}-1)}{(-a)^{d-1}(a^2-1)}$$

which gives (when $y_m = 0$)

$$u(t) = -\frac{(-a)^d}{\mu}y(t) = \frac{a^{2d-1}(a^2-1)}{b(a^{2d}-1)}y(t)$$

The pole of the closed-loop system is

$$p_d = -a + \frac{a^{2d-1}(a^2-1)}{a^{2d}-1} = -\frac{a^{2d-1}+a}{a^{2d}-1}$$

which gives

$$\lim_{d\to\infty} p_d = -a \qquad |a| \le 1 \quad \text{(stable open-loop system)}$$

$$\lim_{d\to\infty} p_d = -1/a \qquad |a| > 1 \quad \text{(unstable open-loop system)}$$

For this example the minimum-effort controller gives a better closed-loop system than if the future control is assumed to be constant. □

Generalized Predictive Control (GPC)

The predictive controllers discussed so far have considered the output at only one future instant of time. Different generalizations of predictive control have been suggested, in which different loss functions are minimized. One possibility is to use

$$J(N_1, N_2, N_u) = E\left\{\sum_{k=N_1}^{N_2}\Big(y(t+k) - y_m(t+k)\Big)^2 + \sum_{k=1}^{N_u}\rho\Delta u(t+k-1)^2\right\} \tag{4.61}$$

where

$$\Delta = 1 - q^{-1}$$

is the difference operator. Different choices of N_1, N_2, and N_u give rise to the different schemes suggested in the literature.

The methodology of generalized predictive control is illustrated by using the loss function of Eq. (4.61) and the process model

$$A^*(q^{-1})y(t) = B^*(q^{-1})u(t-d_0) + \frac{e(t)}{\Delta} \tag{4.62}$$

This model is sometimes called the *CARIMA* (controlled auto-regressive integrating moving-average) model. It has the advantage that the controller will automatically contain an integrator. (Compare Section 3.6.) As with Eq. (4.50), the following identity is introduced:

$$1 = A^*(q^{-1})F_d^*(q^{-1})(1-q^{-1}) + q^{-d}G_d^*(q^{-1}) \tag{4.63}$$

This can be used to determine the output d steps ahead:

$$y(t+d) = F_d^* B^* \Delta u(t+d-d_0) + G_d^* y(t) + F_d^* e(t+d)$$

F_d^* is of degree $d-1$. The optimal mean squared error predictor, given measured output up to time t and any given input sequence, is

$$\hat{y}(t+d) = F_d^* B^* \Delta u(t+d-d_0) + G_d^* y(t) \tag{4.64}$$

Suppose that the future desired outputs, $y_m(t+k)$, $k = 1, 2, \ldots$ are available. The loss function of Eq. (4.61) can now be minimized, giving a sequence of future control signals. Notice that the expectation in Eq. (4.61) is made with respect to data obtained up to time t, assuming that no future measurements are available. That is, it is assumed that the computed control sequence is applied to the system. However, only the first element of the control sequence is used. The calculations are repeated when a new measurement is obtained. The resulting controller belongs to the confusingly named class called *open-loop-optimal-feedback* control. As the name suggests, it is assumed that feedback is used, but it is computed only on the basis of the information available at the present time.

In analogy with Eq. (4.54) we get

$$\begin{aligned} y(t+1) &= R_1^*(q^{-1})\Delta u(t+1-d_0) + \bar{y}_1(t) - F_1^* e(t+1) \\ y(t+2) &= R_2^*(q^{-1})\Delta u(t+2-d_0) + y_2(t) + F_2^* e(t-2) \\ &\vdots \\ y(t+N) &= R_N^*(q^{-1})\Delta u(t+N-d_0) + \bar{y}_N(t) + F_N^* e(t+N) \end{aligned}$$

Each output value depends on future control signals (if $d > d_0$), measured inputs, and future noise signals. The equations above can be written as

$$\mathbf{y} = \mathbf{R}\Delta\mathbf{u} + \bar{\mathbf{y}} + \mathbf{e}$$

where

$$\begin{aligned} \mathbf{y} &= \begin{pmatrix} y(t+1) & \dots & y(t+N) \end{pmatrix}^T \\ \Delta\mathbf{u} &= \begin{pmatrix} \Delta u(t+1-d_0) & \dots & \Delta u(t+N-d_0) \end{pmatrix}^T \\ \bar{\mathbf{y}} &= \begin{pmatrix} \bar{y}_1(t) & \dots & \bar{y}_N(t) \end{pmatrix}^T \\ \mathbf{e} &= \begin{pmatrix} F_1^* e(t+1) & \dots & F_N^* e(t+N) \end{pmatrix}^T \end{aligned}$$

From Eq. (4.53) it follows that the coefficients of R_d^* are the first $d - d_0 + 1$ terms of the pulse response of $q^{-d_0}B^*/(A^*\Delta)$, which are the same as the first $d - d_0 + 1$ terms of the step response of $q^{-d_0}B^*/A^*$. The matrix R is thus a lower triangular matrix:

$$\mathbf{R} = \begin{pmatrix} r_0 & 0 & \dots & 0 \\ r_1 & r_0 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ r_{N-1} & r_{N-2} & \dots & r_0 \end{pmatrix}$$

If there is a dead time in the system, $d_0 > 1$, then the first $d_0 - 1$ rows of $\mathbf{R}$ will be zero. Also introduce

$$\mathbf{y_m} = \begin{pmatrix} y_m(t+1) & \dots & y_m(t+N) \end{pmatrix}^T$$

The expected value of the loss function can be written as

$$\begin{aligned} J(1,N,N) &= E\left\{(\mathbf{y}-\mathbf{y_m})^T(\mathbf{y}-\mathbf{y_m}) + \rho\Delta\mathbf{u}^T\Delta\mathbf{u}\right\} \\ &= (\mathbf{R}\Delta\mathbf{u} + \bar{\mathbf{y}} - \mathbf{y_m})^T(\mathbf{R}\Delta\mathbf{u} + \bar{\mathbf{y}} - \mathbf{y_m}) + \rho\Delta\mathbf{u}^T\Delta\mathbf{u} \qquad (4.65) \end{aligned}$$

Minimization of this expression with respect to $\Delta\mathbf{u}$ gives

$$\Delta\mathbf{u} = (\mathbf{R}^T\mathbf{R} + \rho I)^{-1}\mathbf{R}^T(\mathbf{y_m} - \bar{\mathbf{y}}) \qquad (4.66)$$

The first component in $\Delta\mathbf{u}$ is $\Delta u(t)$, which is the control signal applied to the system. Notice that the controller automatically has an integrator. This is necessary to compensate for the drifting noise term in Eq. (4.62).

Notice that $\mathbf{R}$ is independent of the measurements and the old control signals. Only $\mathbf{y_m}$ and $\bar{\mathbf{y}}$ depend on the measurements. The controller (4.66) is thus a time-invariant controller if the process is time-invariant. The predictive controller can thus be interpreted in terms of a pole placement controller. For instance, $N_u = N_1 = n+1$, $N_2 \geq 2(n+1) - 1$, and $\rho = 0$ leads to a deadbeat controller.

The calculation of Eq. (4.66) involves the inversion of an $N \times N$ matrix, where N is the prediction horizon in the loss function. To decrease the computations, it is possible to introduce constraints on the future control signals. For instance, it can be assumed that the control increments are zero after $N_u < N$ steps:

$$\Delta u(t+k-1) = 0 \qquad k > N_u$$

This implies that the control signal is assumed to be constant after N_u steps. Compare the constraint of Eq. (4.55). The control law (Eq. 4.66) then changes to

$$\Delta\mathbf{u} = (\mathbf{R_1}^T\mathbf{R_1} + \rho I)^{-1}\mathbf{R_1}^T(\mathbf{y_m} - \bar{\mathbf{y}}) \qquad (4.67)$$

where $\mathbf{R_1}$ is the $N \times N_u$ matrix

$$\mathbf{R_1} = \begin{pmatrix} r_0 & 0 & \dots & 0 \\ r_1 & r_0 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ & & \dots & r_0 \\ \vdots & & & \vdots \\ r_{N-1} & r_{N-2} & \dots & r_{N-N_u} \end{pmatrix}$$

The matrix to be inverted is now of order $N_u \times N_u$.

One advantage of the receding horizon controllers is that it is possible to include constraints in the states and the control signal. References to this are given at the end of the chapter. One disadvantage with the GPC is that there are many parameters to determine, and it is not obvious how to choose the parameters to get a stable closed-loop system.

The output and control horizons can be chosen as follows: The lower limit N_1 in Eq. (4.61) indicates the first output that will be used in the loss function. The first output that is influenced by $u(t)$ is $y(t+d_0)$. If the time delay is known, then $N_1 = d_0$ is the obvious choice. When the time delay is unknown, $N_1 = 1$ or $N_1 = d_{0min}$ could be used, where d_{0min} is an estimate of the lower limit of the delay. For unknown delays the order of the B polynomial should be increased to make it possible to include all possible values of d_0. This will make the adaptive GPC quite insensitive to variations in the time delay.

The maximum output horizon N_2 can be chosen such that $N_2 h$ is of the same magnitude as the rise time of the plant, where h is the sampling time of the controller. If the system is nonminimum phase, then N_2 should be chosen such that N_2 exceeds the degree of the B polynomial. This will imply that the maximum output horizon is longer than a possible negative-going nonminimum-phase transient.

The control horizon N_u is an important design parameter. As a rule, N_u should be longer the more complex the process is. For processes that are unstable or close to the stability boundary it is necessary to use a N_u that is at least equal to the number of unstable or poorly damped poles. For simpler processes, $N_u = 1$ often gives good results.

To make the generalized predictive controller adaptive, it is necessary at each step of time to estimate the A^* and B^* polynomials. The predicted values for different prediction horizons are computed, and the control signal is calculated from Eq. (4.67). The adaptive generalized predictive controller is thus an indirect control algorithm. The predictions of Eq. (4.64) can be computed recursively, which will simplify the computations. Finally, N_u is usually chosen to be small, which implies that only a low-order matrix needs to be inverted. The adaptive version of GPC has shown good performance and a certain degree of robustness with respect to the choice of model order and poorly known time delays.

To investigate the closed-loop properties of the system in using GPC, we first determine the control signal $\Delta u(t)$ from Eq. (4.67):

$$\begin{aligned}\Delta u(t) &= \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix} \left(\mathbf{R_1}^T\mathbf{R_1} + \rho I\right)^{-1} \mathbf{R_1}^T\left(\mathbf{y_m} - \bar{\mathbf{y}}\right) \\ &= \begin{pmatrix} \alpha_1 & \dots & \alpha_N \end{pmatrix} \left(\mathbf{y_m} - \bar{\mathbf{y}}\right)\end{aligned}$$

Further, from Eq. (4.62), using Eq. (4.54),

$$\bar{\mathbf{y}} = \begin{pmatrix} \bar{R}_1^*\Delta u(t-1) + G_1^* y(t) \\ \vdots \\ \bar{R}_N^*\Delta u(t-1) + G_N^* y(t) \end{pmatrix} = \begin{pmatrix} \dfrac{\bar{R}_1^* A^*\Delta}{B^*} q^{d_0-1} + G_1^* \\ \vdots \\ \dfrac{\bar{R}_N^* A^*\Delta}{B^*} q^{d_0-1} + G_N^* \end{pmatrix} y(t)$$

The closed-loop system has the characteristic equation

$$A^*\Delta + \begin{pmatrix} \alpha_1 & \dots & \alpha_N \end{pmatrix} \begin{pmatrix} \bar{R}_1^* A^*\Delta\, q^{d_0-1} + B^* G_1^* \\ \vdots \\ \bar{R}_N^* A^*\Delta\, q^{d_0-1} + B^* G_N^* \end{pmatrix}$$

The identity of Eq. (4.63) gives

$$\begin{aligned}B^* &= A^*\Delta B^* F_d^* + q^{-d} G_d^* B^* \\ &= A^*\Delta\left(R_d^* + q^{-(d-d_0+1)}\bar{R}_d^*\right) + q^{-d} G_d^* B^*\end{aligned}$$

This gives the characteristic equation

$$\begin{aligned}A^*\Delta + \begin{pmatrix} \alpha_1 & \dots & \alpha_N \end{pmatrix} \begin{pmatrix} (B^* - A^*\Delta R_1^*)q \\ \vdots \\ (B^* - A^*\Delta R_N^*)q^N \end{pmatrix} \\ = A^*\Delta + \sum_{i=1}^{N} \alpha_i q^i (B^* - A^*\Delta R_i^*) \qquad (4.68)\end{aligned}$$

Equation (4.68) gives an expression for the closed-loop characteristic equation, but it is still difficult to draw any general conclusions about the properties of the closed-loop system even when the process is known.

If $N_u = 1$, then

$$\alpha_i = \frac{r_i}{\rho + \sum_{j=1}^{N} r_j^2}$$

If ρ is sufficiently large, the closed-loop system becomes unstable if the open-loop process is unstable. However, if both the control and output horizons are increased, the problem is the same as a finite-horizon linear quadratic control problem and should thus have better stability properties.

The model predictive controllers such as GPC have the drawback that there are many parameters to choose. Even if there are rules of thumb for choosing the parameters, it is sometimes difficult to determine the parameters such that the closed-loop system is stable (see Problem 4.12). This difficulty exists both when the process is known and when an adaptive GPC algorithm is used.

It is easily seen that the GPC control problem can be interpreted as a stationary LQG control problem but with time-varying weighting matrices or as a finite horizon LQG problem. Compare the loss functions (4.47) and (4.61). The stationary LQG problem and the associated Riccati equation have been extensively studied, and there is much knowledge about the properties of the closed-loop system. The drawback of the infinite-horizon LQG formulation is that it cannot handle constraints in the states or the control signal. Different ways to formulate and solve the constrained receding horizon problem are given in the references at the end of the chapter.

4.7 CONCLUSIONS

This chapter has reviewed different self-tuning regulators. The basic idea is to make a separation between the estimation of the unknown parameters of the process and the design of the controller. The estimated parameters are assumed to be equal to the true parameters in making the design of the controller. It is sometimes of interest to include the uncertainties of the parameter estimates in the design. Such controllers are discussed in Chapter 7. By combining different estimation schemes and design methods, it is possible to derive self-tuners with different properties. In this chapter, only the basic ideas and the asymptotic properties are discussed. The convergence of the estimates and the stability of the closed-loop system are discussed in Chapter 6.

The most important aspect of self-tuning regulators is the issue of parameterization. A reparameterization can be achieved by using the process model *and* the desired closed-loop response. The goal of the reparameterization is to make a direct estimation of the controller parameters, which usually implies that the new model should be linear in the controller parameters.

Only a few of the proposed self-tuning algorithms have been treated in this chapter. Different combinations of estimation methods and underlying control problems give algorithms with different properties. One goal of the chapter has been to give a feel for how self-tuning algorithms can be developed and analyzed. It is important that the desired closed-loop specifications are carefully chosen in applying a self-tuner. A design method that is unsuitable when the process is known will not become better when the process is unknown.

It is also possible to derive self-tuning regulators for multi-input, multi-output (MIMO) systems. The MIMO case is more difficult to analyze. One main difficulty is to define what the necessary *a priori* knowledge is in the MIMO case. It is quite straightforward to derive a self-tuning algorithm corresponding to the generalized direct self-tuning regulator for the restricted case when the delays between the different inputs and outputs are known.

PROBLEMS

4.1 Consider the process and controller in Example 4.4. The controller parameter $\hat{s}_0$ may be very large if $\hat{b}$ is small. Discuss alternatives to ensure that the controller parameter stays bounded.

4.2 Consider the basic direct self-tuning controller in Algorithm 4.1. Discuss different ways to incorporate reference values in the controller. What are the properties of the following three ways for taking care of the reference value?

(a) Use the difference $y - u_c$ instead of y in the algorithm, and introduce an integrator in the controller.

(b) Estimate the parameters using the model

$$y(t+d) = R^*u + S^*y - T^*u_c + \varepsilon$$

and let the controller be

$$R^*u = -S^*y + T^*u_c$$

(c) Use the difference $u_c - y$ instead of y in the algorithm, and introduce an integrator in the controller.

4.3 Show that the control equation (4.37) minimizes the loss function (4.36).

4.4 Consider the system in Example 4.6. Assume that the process is known. Compute the optimal minimum-variance controller and the least attainable output variance when (a) $\tau = 0.4$ (the minimum-phase case) and (b) $\tau = 0.6$ (the nonminimum-phase case). (*Hint*: Use Theorem 4.3 for the nonminimum-phase case.)

4.5 Make the same calculations as in Problem 4.4 but for the moving-average controller with $d = 2$.

4.6 Consider the generalized minimum-variance controller of Eq. (4.37). Compute the closed-loop characteristic equation. Discuss when the design method may give an unstable closed-loop system. For instance, is it useful for the process in Example 4.6 when $\tau = 0.6$?

4.7 Consider the process in Example 4.6 when $\tau = 0.6$ and $C = 0$. Use Eq. (4.67) to compute the closed-loop poles for different values of N when $N_u = 1$.

4.8 Show that the moving-average controller with $B^{+*} = 1$ and $d = n$ corresponds to a state deadbeat controller.

4.9 Consider the process in Example 4.3. Assume that

$$A(q) = q^2 - 1.5q + 0.7$$
$$B(q) = q + b_1$$
$$C(q) = q^2 - q + 0.2$$

Determine the variance of the closed-loop system as a function of b_1 when the moving-average controller is used. Compare with the lowest achievable variance.

4.10 Show that the control law (4.66) minimizes the loss function (4.65).

4.11 Consider the process in Example 4.5. Investigate through simulation what values of $\hat{r}_0$ can be used. Make the simulations with and without bounds on the control signal. How sensitive is the choice of initial values in the algorithm?

4.12 Consider the system (4.62) with $e(t) = 0$ and

$$A^*(q^{-1}) = 1 - 4q^{-1} + 4q^{-2} = (1 - 2q^{-1})^2$$
$$B^*(q^{-1}) = q^{-1} - 1.999q^{-2}$$

The open-loop process is unstable, and there is a near pole-zero cancellation. Assume that $\rho = 0.1$ and compute the generalized predictive controller that minimizes Eq. (4.65) for different values of N. How large must N be to get a stable closed-loop system? (The problem is adopted from Bitmead *et al.* (1990).) (*Hint:* Don't give up until $N > 25$.)

4.13 Consider the system in Problem 1.9.

(a) Sample the system and assume that e is discrete-time measurement noise. Determine the minimum-variance controller for the system.

(b) Simulate a self-tuning moving-average controller for different prediction horizons.

4.14 Make the same investigation as in Problem 4.12 but for the process in Problem 1.10.

REFERENCES

There are many papers, reports, and books about self-tuning algorithms. Some fundamental references are given in this section. The first publication of the self-tuning idea is probably:

Kalman, R. E., 1958. "Design of a self-optimizing control system." *Trans. ASME* **80**: 468–478.

In this paper, least-squares estimation combined with deadbeat control is discussed. Two similar algorithms based on least-squares estimation and minimum-variance control were presented at an IFAC symposium in Prague 1970:

Peterka, V., 1970. "Adaptive digital regulation of noisy systems." *Preprints 2nd IFAC Symposium on Identification and Process Parameter Estimation.* Prague.

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

The first thorough presentation and analysis of a self-tuning regulator were given in:

Åström, K. J., and B. Wittenmark, 1972. "On the control of constant but unknown systems." *Proceedings of the 5th IFAC World Congress,* Pt 3, Paper 37.5. Paris.

A revised version of this paper, in which the phrase "self-tuning regulator" was coined, is:

Åström, K. J., and B. Wittenmark, 1973. "On self-tuning regulators." *Automatica* **9**: 185–199.

Different aspects of the basic self-tuning regulator described in Algorithm 4.1 are given in the thesis:

Wittenmark, B., 1973. "A self-tuning regulator." Ph.D. thesis TFRT-1003, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

The generalized minimum-variance self-tuner was presented in:

Clarke, D. W., and P. J. Gawthrop, 1975. "A self-tuning controller." *IEE Proc.* **122**: 929–934.

The papers above inspired intensive research activity in adaptive control based on the self-tuning idea. A comprehensive treatment of the fundamental theory of adaptive control, especially self-tuning algorithms, is 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.

A more recent state-of-the-art article is:

Ren, W., and P. R. Kumar, 1994. "Stochastic adaptive prediction and model reference control." *IEEE Trans. Automat. Contr.* **AC-39**(10).

The problem of controlling nonminimum-phase plants is discussed in:

Åström, K. J., 1980. "Direct methods for nonminimum phase systems." *Proceedings of the 19th Conference on Decision and Control,* pp. 611–615. Albuquerque, N.M.

Clarke, D. W., 1984. "Self-tuning control of nonminimum-phase systems." *Automatica* **20**(5, Special Issue on Adaptive Control): 501–517.

Åström, K. J., and B. Wittenmark, 1985. "The self-tuning regulator revisited." *Preprints 7th IFAC Symposium on Identification and System Parameter Estimation.* York, U.K.

In the latter, the moving-average controller is presented. Algorithm 4.2 can be used to explain the pole-zero assignment controller in:

Wellstead, P. E., J. M. Edmunds, D. Prager, and P. Zanker, 1979. "Self-tuning pole/zero assignment regulators." *Int. J. Control* **30**: 1–26.

In Åström and Wittenmark (1985), the moving-average controller is presented. It also gives a motivation for the more heuristically introduced model-reference self-tuner in Clarke (1984), where a prediction model of the form of Eq. (4.31) is used but with different filtering.

Multivariable self-tuning regulators are treated in:

Borisson, U., 1979. "Self-tuning regulators for a class of multivariable systems." *Automatica* **15**: 209–215.

Goodwin, G. C., and R. S. Long, 1980. "Generalization of results on multivariable adaptive control." *IEEE Trans. Automat. Contr.* **AC-25**: 1241–1245.

Koivo, H., 1980. "A multivariable self-tuning controller." *Automatica* **16**: 351–356.

Johansson, R., 1983. "Multivariable adaptive control." Ph.D. thesis TFRT-1024, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Dugard, L., G. C. Goodwin, and X. Xianya, 1984. "The role of the interactor matrix in multivariable stochastic adaptive control." *Automatica* **20**(5, Special Issue on Adaptive Control): 701–709.

Elliott, H., and W. A. Wolovich, 1984. "Parameterization issues in multivariable adaptive control." *Automatica* **20**(5, Special Issue on Adaptive Control): 533–545.

Johansson, R., 1986. "Parametric models of linear multivariable systems for adaptive control." *IEEE Trans. Automat. Contr.* **AC-32**: 303–313.

Wittenmark, B., R. H. Middleton, and G. C. Goodwin, 1987. "Adaptive decoupling of multivariable systems." *Int. J. Control* **46**: 1993–2009.

Model predictive control is discussed in:

Richalet, J. A., A. Rault, J. L. Testud, and J. Papon, 1978. "Model predictive heuristic control: Applications to industrial processes." *Automatica* **14**: 413–428.

Cutler, C. R., and B. C. Ramaker, 1980. "Dynamic matrix control—A computer control algorithm." Paper WP5-B, *Preprints Joint Automatic Control Conference*. San Francisco, Calif.

Ydstie, B. E., 1982. "Robust adaptive control of chemical processes." Ph.D. thesis, Imperial College, University of London.

Ydstie, B. E., 1984. "Extended horizon adaptive control." Paper 14.4/E-4, *Preprints 9th IFAC World Congress*. Budapest.

De Keyser, R. M. C., and A. R. Van Cauwenberghe, 1985. "Extended prediction self-adaptive control." *Preprints 7th IFAC Symposium on Identification and System Parameter Estimation*, pp. 1255–1260. York, UK.

Clarke, D. W., C. Mohtadi, and P. S. Tuffs, 1987a. "Generalized predictive control. Part I: The basic algorithm." *Automatica* **23**: 137–148.

Clarke, D. W., C. Mohtadi, and P. S. Tuffs, 1987b. "Generalized predictive control. Part II: Extensions and interpretations." *Automatica* **23**: 149–160.

Clarke, D. W., and C. Mohtadi, 1989. "Properties of generalized predictive control." *Automatica* **25**: 859–875.

Garcia, C. E., and M. Morari, 1989. "Model predictive control: theory and practice–A survey." *Automatica* **25**: 335–348.

Bitmead, R. R., M. Gevers, and V. Wertz, 1990. *Adaptive Optimal Control: The Thinking Man's GPC.* Englewood Cliffs, N.J.: Prentice-Hall.

Clarke, D. W., ed., 1994. *Advances in Model-Based Predictive Control.* Oxford, U.K.: Oxford University Press.

Stability of receding horizon controllers with and without constraints are discussed in Bitmead et al. (1990) and in:

Kwon, W. H., and A. E. Pearson, 1977. "A modified quadratic cost problem and feedback stabilization of a linear system." *IEEE Trans. Automat. Contr.* **AC-22**: 838–842.

Clarke, D. W., and R. Scattolini, 1991. "Constrained receding-horizon predictive control." *Proc. IEE Pt. D* **138**: 347–354.

Mosca, E., and J. Zhang, 1992. "Stable redesign of predictive control." *Automatica* **28**: 1229–1233.

Rawlings, J. B., and K. R. Muske, 1993. "The stability of constrained receding horizon control." *IEEE Trans. Automat. Contr.* **AC-38**: 1512–1516.

Michalska, H., and D. Q. Mayne, 1993. "Robust receding horizon control of constrained nonlinear systems." *IEEE Trans. Automat. Contr.* **AC-38**: 1623-1633.

Linear quadratic Gaussian self-tuning regulators are treated in:

Peterka, V., and K. J. Åström, 1973. "Control of multivariable systems with unknown but constant parameters." *Preprints 3rd IFAC Symposium on Identification and System Parameter Estimation*, pp. 535–544. The Hague, Netherlands.

Åström, K. J., and Z. Zhou-Ying, 1982. "A linear quadratic Gaussian self-tuner." *Recerche di Automatica* **13**: 106–122.

Mosca, E., G. Zappa, and C. Manfredi, 1982. "Progress in multistep horizon self-tuners: The MUSMAR approach." *Ricerche di Automatica* **13**(1): 85–105.

Åström, K. J., 1984. "LQG self-tuners." *Proceedings of the IFAC Workshop on Adaptive Systems in Control and Signal Processing, San Francisco 1983.* New York: Pergamon Press.

Greco, C., G. Menga, E. Mosca, and G. Zappa, 1984. "Performance improvements of self-tuning controllers by multistep horizons: The MUSMAR approach." *Automatica* **20**: 681–699.

Grimble, M. J., 1984. "Implicit and explicit LQG self-tuning controllers." *Automatica* **20**: 661–669.

Peterka, V., 1984. "Predictor-based self-tuning control." *Automatica* **20**: 39–50.

Clarke, D. W., P. P. Kanjilal, and C. Mohtadi, 1985a. "A generalized LQG approach to self-tuning control. Part I: Aspects of design." *Int. J. Control* **41**: 1509–1523.

Clarke, D. W., P. P. Kanjilal, and C. Mohtadi, 1985b. "A generalized LQG approach to self-tuning control. Part II: Implementation and simulation." *Int. J. Control* **41**: 1525–1544.

A detailed treatment of LQG self-tuners is given in:

Kárný, M., A. Halousková, J. Böhm, R. Kulhavý, and P. Nedoma, 1985. "Design of linear quadratic adaptive control: Theory and algorithms for practice." Supplement to *Kybernetica* **21**: 3–97.

It contains much information and many useful hints for practical applications.

Design methods for stochastic systems that are useful in self-tuning regulators are given in:

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

Åström, K. J., and B. Wittenmark, 1990. *Computer Controlled Systems: Theory and Design*, 2nd edition. Englewood Cliffs, N.J.: Prentice-Hall.

More about the Sylvester matrix can be found in:

Barnett, S., 1971. *Matrices in Control Theory.* New York: Van Nostrand Reinhold.

Barnett, S., 1983. *Polynomials and Linear Control Systems.* New York: Marcel Dekker.

CHAPTER 5

MODEL-REFERENCE ADAPTIVE SYSTEMS

5.1 INTRODUCTION

The model-reference adaptive system (MRAS) is an important adaptive controller. It may be regarded as an adaptive servo system in which the desired performance is expressed in terms of a reference model, which gives the desired response to a command signal. This is a convenient way to give specifications for a servo problem. A block diagram of the system is shown in Fig. 5.1. The system has an ordinary feedback loop composed of the process and the controller and another feedback loop that changes the controller parameters. The parameters are changed on the basis of feedback from the error, which is the

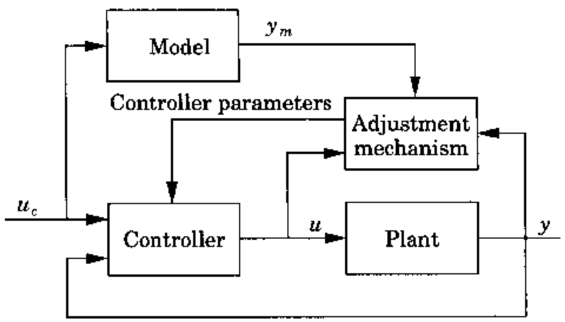

Figure 5.1 Block diagram of a model-reference adaptive system (MRAS).

difference between the output of the system and the output of the reference model. The ordinary feedback loop is called the inner loop, and the parameter adjustment loop is called the outer loop. The mechanism for adjusting the parameters in a model-reference adaptive system can be obtained in two ways: by using a gradient method or by applying stability theory.

In the MRAS the desired behavior of the system is specified by a model, and the parameters of the controller are adjusted based on the error, which is the difference between the outputs of the closed-loop system and the model. Model-reference adaptive systems were originally derived for deterministic continuous-time systems. Extensions to discrete-time systems and systems with stochastic disturbances were given later.

The presentation in this chapter follows the historical development. The MIT rule is derived in Section 5.2. This rule has one parameter, the adaptation gain, that must be chosen by the user. In Section 5.3 we discuss methods to determine the adaptation gain. Section 5.4 presents Lyapunov's stability theory, and Section 5.5 shows how this theory can be used to derive stable adaptation laws. These laws are similar to those obtained by the MIT rule. In Section 5.6 we introduce the theory for input-output stability. This gives another way of viewing adaptive control systems, which is presented in Section 5.7. In Section 5.8 we show how MRASs can be obtained for output feedback of general linear systems. Section 5.9 gives a comparison between self-tuning regulators and MRASs. Adaptive control of nonlinear systems is briefly discussed in Section 5.10. The chapter is summarized in Section 5.11. Further insight into model reference adaptive systems is given in Chapter 6.

5.2 THE MIT RULE

The MIT rule is the original approach to model-reference adaptive control. The name is derived from the fact that it was developed at the Instrumentation Laboratory (now the Draper Laboratory) at MIT.

To present the MIT rule, we will consider a closed-loop system in which the controller has one adjustable parameter θ. The desired closed-loop response is specified by a model whose output is y_m. Let e be the error between the output y of the closed-loop system and the output y_m of the model. One possibility is to adjust parameters in such a way that the loss function

$$J(\theta) = \frac{1}{2}e^2 \tag{5.1}$$

is minimized. To make J small, it is reasonable to change the parameters in the direction of the negative gradient of J, that is,

$$\frac{d\theta}{dt} = -\gamma\frac{\partial J}{\partial\theta} = -\gamma e\frac{\partial e}{\partial\theta} \tag{5.2}$$

This is the celebrated *MIT rule*. The partial derivative $\partial e/\partial\theta$, which is called the *sensitivity derivative* of the system, tells how the error is influenced by the adjustable parameter. If it is assumed that the parameter changes are slower than the other variables in the system, then the derivative $\partial e/\partial\theta$ can be evaluated under the assumption that θ is constant.

There are many alternatives to the loss function given by Eq. (5.1). If it is chosen to be

$$J(\theta) = |e| \tag{5.3}$$

the gradient method gives

$$\frac{d\theta}{dt} = -\gamma \frac{\partial e}{\partial\theta} \operatorname{sign} e \tag{5.4}$$

The first MRAS that was implemented was based on this formula. There are, however, many other possibilities, for example,

$$\frac{d\theta}{dt} = -\gamma \operatorname{sign}\left(\frac{\partial e}{\partial\theta}\right) \operatorname{sign}(e)$$

This is called the *sign-sign algorithm*. A discrete-time version of this algorithm is used in telecommunications, in which simple implementation and fast computations are required. (See Section 13.2.)

Adjusting many parameters Equation (5.2) also applies when there are many parameters to adjust. The symbol θ should then be interpreted as a vector and $\partial e/\partial\theta$ as the gradient of the error with respect to the parameters.

Examples

We now give two examples that illustrate how the MIT rule is used to obtain a simple adaptive controller, and we also show some properties of adaptive systems.

EXAMPLE 5.1 Adaptation of a feedforward gain

Consider the problem of adjusting a feedforward gain. In this problem it is assumed that the process is linear with the transfer function $kG(s)$, where $G(s)$ is known and k is an unknown parameter. The underlying design problem is to find a feedforward controller that gives a system with the transfer function $G_m(s) = k_0G(s)$, where k_0 is a given constant. With the feedforward controller

$$u = \theta u_c$$

where u is the control signal and u_c the command signal, the transfer function from command signal to the output becomes $\theta kG(s)$. This transfer function is equal to $G_m(s)$ if the parameter θ is chosen to be

$$\theta = \frac{k_0}{k}$$

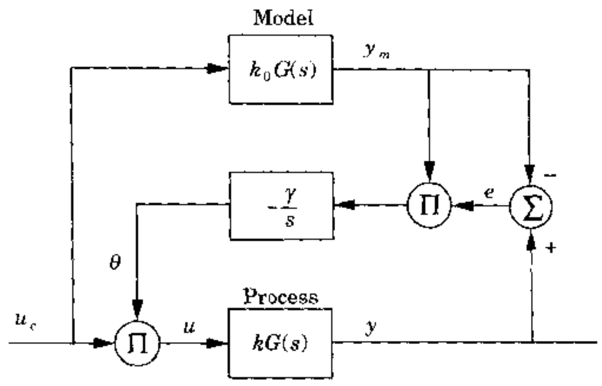

Figure 5.2 Block diagram of an MRAS for adjustment of a feedforward gain based on the MIT rule.

We will now use the MIT rule to obtain a method for adjusting the parameter θ when k is not known. The error is

$$e = y - y_m = kG(p)\theta u_c - k_0G(p)u_c$$

where u_c is the command signal, y_m is the model output, y is the process output, θ is the adjustable parameter, and $p = d/dt$ is the differential operator. The sensitivity derivative is given by

$$\frac{\partial e}{\partial\theta} = kG(p)u_c = \frac{k}{k_0} y_m$$

The MIT rule then gives the following adaptation law:

$$\frac{d\theta}{dt} = -\gamma' \frac{k}{k_0} y_m e = -\gamma y_m e \tag{5.5}$$

where $\gamma = \gamma' k/k_0$ has been introduced instead of γ'. Notice that to have the correct sign of γ, it is necessary to know the sign of k. Equation (5.5) gives the law for adjusting the parameter. A block diagram of the system is shown in Fig. 5.2.

The properties of the system can be illustrated by simulation. Figure 5.3 shows a simulation when the system has the transfer function

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

The input u_c is a sinusoid with frequency 1 rad/s, and the parameter values are $k = 1$ and $k_0 = 2$. Figure 5.3 shows that the parameter converges toward the correct value reasonably fast when the adaptation gain is $\gamma = 1$ and that the process output approaches the model output. Figure 5.3 also shows that

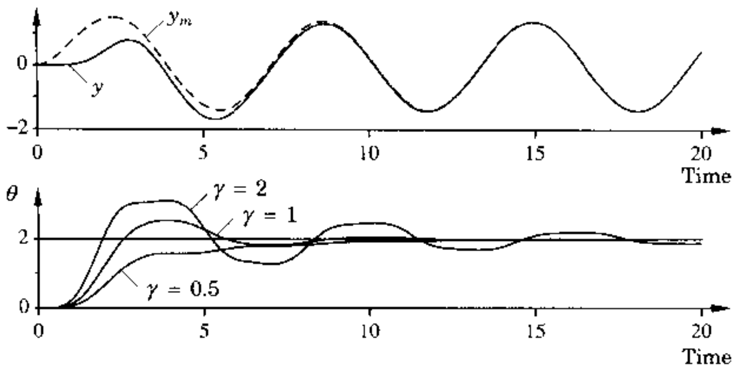

Figure 5.3 Simulation of an MRAS for adjusting a feedforward gain. The process (solid line) and the model (dashed line) outputs are shown in the upper graph for $\gamma = 1$. The controller parameter is shown in the lower graph when the adaptation gain γ has the values 0.5, 1, and 2.

the convergence rate depends on the adaptation gain. It is thus important to know a reasonable value of this parameter. Intuitively, we may expect that parameters converge slowly for small γ and that the convergence rate increases with γ. Simulation experiments indicate that this is true for small values of γ but also that the behavior is quite unpredictable for large γ. □

An example of a practical problem that fits this formulation is control of robots with unknown load, in which the process transfer function from the motor current to the angular velocity is

$$G(s) = \frac{k_I}{Js}$$

where k_I is the current to torque constant and J is the unknown moment of inertia. Another example is the dynamics of a CD player, in which the sensitivity of the laser diode is the unknown process parameter.

A remark on notation In analyzing the MRAS with time-varying parameters it is important to consider the fact that the parameter θ is time-varying. The expression

$$G(p)(\theta u)$$

where $p = d/dt$ is the differential operator should be interpreted as the differential operator $G(p)$ acting on the signal θu. When θ is time-varying, this is different from $\theta G(p)u$. For example, if $G(p) = p$, we have

$$G(p)(\theta u) = p(\theta u) = \theta \frac{du}{dt} + \frac{d\theta}{dt} u = \theta(pu) + u(p\theta)$$

Care must thus be taken in manipulating expressions and block diagrams.

Notice that no approximations were needed in Example 5.1. When the MIT rule is applied to more complicated problems, however, it is necessary to use approximations to obtain the sensitivity derivatives. This is illustrated by another example.

EXAMPLE 5.2 MRAS for a first-order system

Consider a system described by the model

$$\frac{dy}{dt} = -ay + bu \tag{5.6}$$

where u is the control variable and y is the measured output. Assume that we want to obtain a closed-loop system described by

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

Let the controller be given by

$$u(t) = \theta_1 u_c(t) - \theta_2 y(t) \tag{5.7}$$

The controller has two parameters. If they are chosen to be

$$\begin{aligned} \theta_1 &= \theta_1^0 = \frac{b_m}{b} \\ \theta_2 &= \theta_2^0 = \frac{a_m - a}{b} \end{aligned} \tag{5.8}$$

the input-output relations of the system and the model are the same. This is called perfect model-following.

To apply the MIT rule, introduce the error

$$e = y - y_m$$

where y denotes the output of the closed-loop system. It follows from Eqs. (5.6) and (5.7) that

$$y = \frac{b\theta_1}{p + a + b\theta_2} u_c$$

where $p = d/dt$ is the differential operator. The notation used is discussed in Section 1.5. The sensitivity derivatives are obtained by taking partial derivatives with respect to the controller parameters θ_1 and θ_2:

$$\frac{\partial e}{\partial \theta_1} = \frac{b}{p + a + b\theta_2} u_c$$

$$\frac{\partial e}{\partial \theta_2} = -\frac{b^2\theta_1}{(p + a + b\theta_2)^2} u_c = -\frac{b}{p + a + b\theta_2} y$$

These formulas cannot be used directly because the process parameters a and b are not known. Approximations are therefore required. One possible approximation is based on the observation that $p + a + b\theta_2^0 = p + a_m$ when the

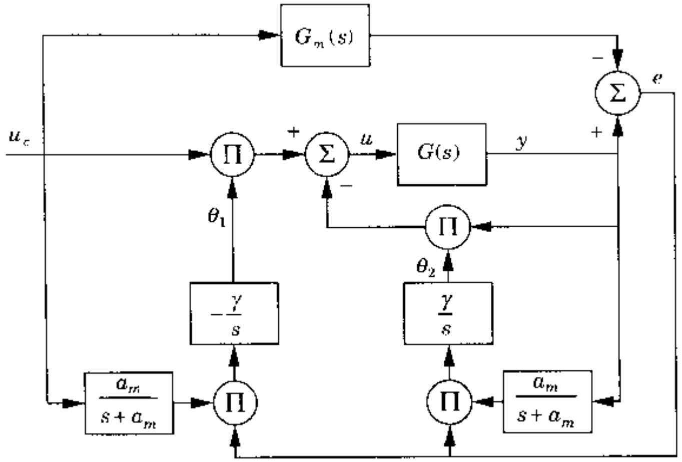

Figure 5.4 Block diagram of a model-reference controller for a first-order process.

parameters give perfect model-following. We will therefore use the approximation

$$p + a + b\theta_2 \approx p + a_m$$

which will be reasonable when parameters are close to their correct values. With this approximation we get the following equations for updating the controller parameters:

$$\begin{aligned} \frac{d\theta_1}{dt} &= -\gamma \left(\frac{a_m}{p + a_m} u_c \right) e \\ \frac{d\theta_2}{dt} &= \gamma \left(\frac{a_m}{p + a_m} y \right) e \end{aligned} \tag{5.9}$$

In these equations we have combined parameters b and a_m with the adaptation gain γ', since they appear as the product $\gamma' b / a_m$. The sign of parameter b must be known to have the correct sign of γ. Notice that the filter has also been normalized so that its steady-state gain is unity.

The adaptive controller is a dynamical system with five state variables that can be chosen to be the model output, the parameters, and the sensitivity derivatives. A block diagram of the system is shown in Fig. 5.4. The behavior of the system is now illustrated by a simulation. The parameters are chosen to be $a = 1$, $b = 0.5$, and $a_m = b_m = 2$, the input signal is a square wave with amplitude 1, and $\gamma = 1$. Figure 5.5 shows the results of a simulation. Figure 5.6 shows the parameter estimates for different values of the adaptation gain γ. Notice that the parameters change most when the command signal changes

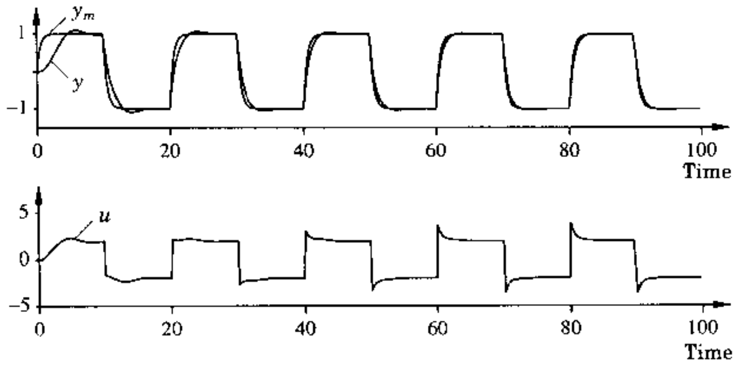

Figure 5.5 Simulation of the system in Example 5.2 using an MRAS. The parameter values are $a = 1$, $b = 0.5$, $a_m = b_m = 2$, and $\gamma = 1$.

and that the parameters converge very slowly. For $\gamma = 1$, the value used in Fig. 5.5, the parameters have the values $\theta_1 = 3.2$ and $\theta_2 = 1.2$ at time $t = 100$. These values are far from the correct values $\theta_1^0 = 4$ and $\theta_2^0 = 2$. However, the parameters will converge to the true values with increasing time. The convergence rate increases with increasing γ, as is shown in Fig. 5.6.

The fact that the control is quite good even at time $t = 10$ is a reflection of the fact that the parameter estimates are related to each other in a very special way, although they are quite far from their true values. This is illustrated in Fig. 5.7, where parameter θ_2 is plotted as a function of θ_1 for a simulation with a duration of 500 time units. Figure 5.7 shows that parameters do indeed

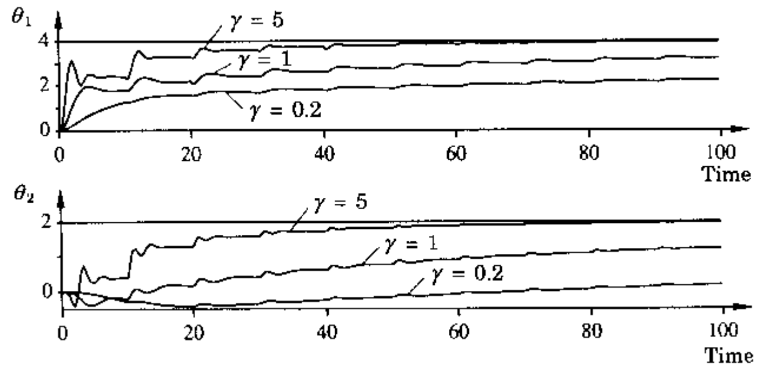

Figure 5.6 Controller parameters θ_1 and θ_2 for the system in Example 5.2 when $\gamma = 0.2$, 1 and 5.

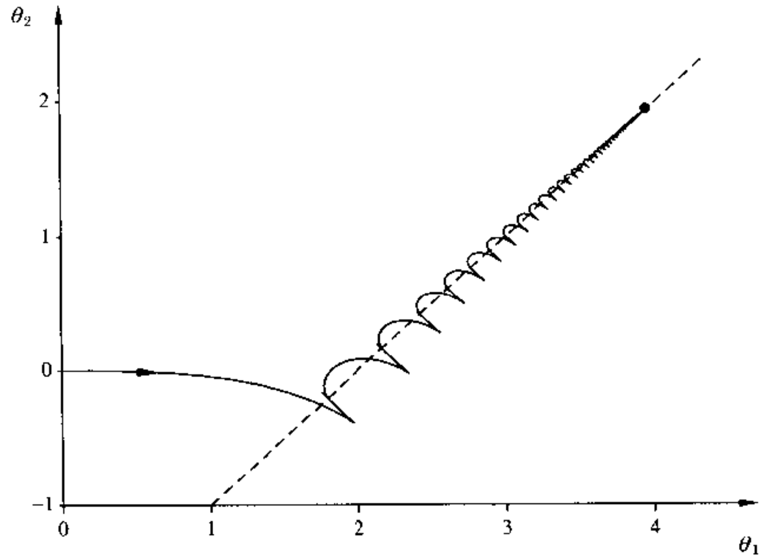

Figure 5.7 Relation between controller parameters θ_1 and θ_2 when the system in Example 5.2 is simulated for 500 time units. The dashed line shows the line $\theta_2 = \theta_1 - a/b$. The dot indicates the convergence point.

approach their correct values as time increases. The parameter estimates quickly approach the line $\theta_2 = \theta_1 - a/b$. This line represents parameter values such that the closed-loop system has correct steady-state gain. □

Error and Parameter Convergence

The goal in model-reference adaptive systems is to drive the error $e = y - y_m$ to zero. This does not necessarily imply that the controller parameters approach their correct values, as is illustrated in the following example.

EXAMPLE 5.3 Lack of parameter convergence

Consider the simple system for updating a feedforward gain, discussed in Example 5.1. Assume that $G(s) = 1$. The process model is $y = ku$, the control law is $u = \theta u_c$, and the desired response is given by $y_m = k_0 u_c$. The error is

$$e = (k\theta - k_0)u_c = k(\theta - \theta^0)u_c$$

where $\theta^0 = k_0/k$. The MIT rule gives the following differential equation for the parameter:

$$\frac{d\theta}{dt} = -\gamma k^2 u_c^2 (\theta - \theta^0)$$

This equation has the solution

$$\theta(t) = \theta^0 + (\theta(0) - \theta^0)e^{-\gamma k^2 I_t} \tag{5.10}$$

where

$$I_t = \int_0^t u_c^2(\tau)\, d\tau$$

and $\theta(0)$ is the initial value of the parameter θ. The estimate converges toward its correct value only if the integral I_t diverges as $t \to \infty$. The convergence is exponential if the input signal is persistently exciting. (Compare with Section 2.4.) The error is given by

$$e(t) = ku_c(t)(\theta(0) - \theta^0)e^{-\gamma k^2 I_t}$$

Notice that the error will always go to zero as $t \to \infty$ because either the integral I_t diverges or else $u_c(t) \to 0$. However, the limiting value of the parameter θ will depend on the properties of the input signal. □

Example 5.3 illustrates the fact that the error e goes to zero but that the parameters do not necessarily converge to their correct values. This is a characteristic feature of all adaptive systems. The input signal must have certain properties for the parameters to converge. The conditions required were discussed in Chapter 2; compare with the notion of persistent excitation, which was introduced in Section 2.4.

5.3 DETERMINATION OF THE ADAPTATION GAIN

In Section 5.2 we found that it was straightforward to obtain an adaptive system by using the MIT rule. The adaptive control laws had one parameter, the adaptation gain γ, which had to be chosen by the user. The simulation experiments indicated that the choice of the adaptation gain could be crucial. In this section we will discuss methods for determining the adaptation gain.

Consider the MRAS for adaptation of a feedforward gain in Example 5.1. We thus have a system with the transfer function $kG(s)$, where $G(s)$ is known and k is an unknown constant. It is assumed that $G(s)$ is stable. We would like to find a feedforward control that gives the transfer function $k_0G(s)$. The system is described by the following equations:

$$\begin{aligned} y &= kG(p)u \\ y_m &= k_0G(p)u_c \\ u &= \theta u_c \\ e &= y - y_m \\ \frac{d\theta}{dt} &= -\gamma y_m e \end{aligned}$$

where u_c is the command signal, y_m is the model output, y is the process output, θ is the adjustable parameter, and $p = d/dt$ is the differential operator. Elimination of the variables u and y in these equations gives

$$\frac{d\theta}{dt} + \gamma y_m \left(kG(p)\theta u_c\right) = \gamma y_m^2 \tag{5.11}$$

This equation is a compact description of the behavior of the parameters that we call the *parameter equation*. Notice that y_m may be considered a known function of time. If $G(s)$ is a rational transfer function, Eq. (5.11) is a linear time-varying ordinary differential equation. Such equations may exhibit very complicated behavior. It is not possible to give a simple analytical characterization of the properties of the system, particularly how they are influenced by the parameter γ.

A Thought Experiment

To get some insight into the behavior of the system given by Eq. (5.11), we consider an experiment with the adaptive system such that the equation simplifies considerably. An understanding of the behavior of the system under such circumstances will give us some insight, but it will of course not give the full picture.

Consider the following experiment: Assume that the value of parameter θ is fixed, that the adaptation mechanism is disconnected, and that a constant input signal u_c is applied. The adaptation mechanism is then connected when all signals have settled to steady-state values. The behavior of the parameter is then given by

$$\frac{d\theta}{dt} + \gamma y_m^o u_c^o \left(kG(p)\theta\right) = \gamma (y_m^o)^2 \tag{5.12}$$

which is a linear time-invariant system. This equation is linear with constant coefficients. Its stability is governed by the algebraic equation

$$s + \gamma y_m^o u_c^o k G(s) = 0 \tag{5.13}$$

We can immediately conclude that the behavior of the parameter is determined by the quantity

$$\mu = \gamma y_m^o u_c^o k \tag{5.14}$$

A picture of how the zeros of Eq. (5.13) vary with parameter μ is easily obtained by plotting the root locus with respect to the parameter. We can conclude that if Eq. (5.13) has zeros in the right half-plane, then the parameters will diverge even in the very special conditions of the experiment. Intuitively, we may also expect the analysis to approximately describe the case in which the command signal is changing slowly with respect to the dynamics of $G(s)$.

Equation (5.13) can also be used to determine a suitable value of the adaptation gain, as is illustrated in Example 5.4.

EXAMPLE 5.4 Choice of adaptation gain

Consider the system in Example 5.1 with $G(s) = 1/(s+1)$, $k = 1$, and $k_0 = 2$. Assume that the reference signal has unit amplitude. Equation (5.13) then becomes

$$s^2 + s + \mu = s^2 + s + \gamma y_m^o u_c^o k = 0$$

A reasonable choice is to make $\gamma y_m^o u_c^o k = 1$. If we disregard the transients, the average value of $y_m u_c$ is 2. This gives $\gamma = 0.5$, which is the value used in one of the simulations in Fig. 5.3. □

Normalized Algorithms

It follows from Eq. (5.13) that the adaptive system will be unstable if the transfer function $G(s)$ has pole excess larger than 1 and parameter μ in Eq. (5.14) is sufficiently large. The parameter μ is large if the signals are large or if the adaptation gain is large. The behavior of the system depends strongly on the signal levels. This will be illustrated by a numerical experiment.

EXAMPLE 5.5 Stability depends on the signal amplitudes

Consider the system in Example 5.1. Let the transfer function G be given by

$$G(s) = \frac{1}{s^2 + a_1 s + a_2}$$

Equation (5.13) then becomes

$$s^3 + a_1 s^2 + a_2 s + \mu = 0$$

where $\mu = \gamma y_m^o u_c^o k$. The equation has all its roots in the left half-plane if

$$\gamma y_m^o u_c^o k < a_1 a_2 \tag{5.15}$$

Since this inequality involves the magnitude of the command signal, it may happen that the equilibrium solution corresponding to one command signal is stable and the solution corresponding to another command signal is unstable. This is illustrated by the simulation results shown in Fig. 5.8, where parameters are chosen so that $k = a_1 = a_2 = 1$. In the simulation the adaptation rate γ was adjusted to give a good response when u_c is a square wave with unit amplitude. In this case we have $u_c^o = y_m^o = 1$, and inequality (5.15) gives the stability condition $\gamma < 1$. A reasonable value of γ is $\gamma = 0.1$, which was used in the simulation. Figure 5.8 shows clearly that the convergence rate depends on the magnitude of the command signal. Notice that the solution is unstable when the amplitude of u_c is 3.5. The approximate model predicts instability for u_c larger than 3.16. Also notice that the response is intolerably slow for low amplitudes of u_c. □

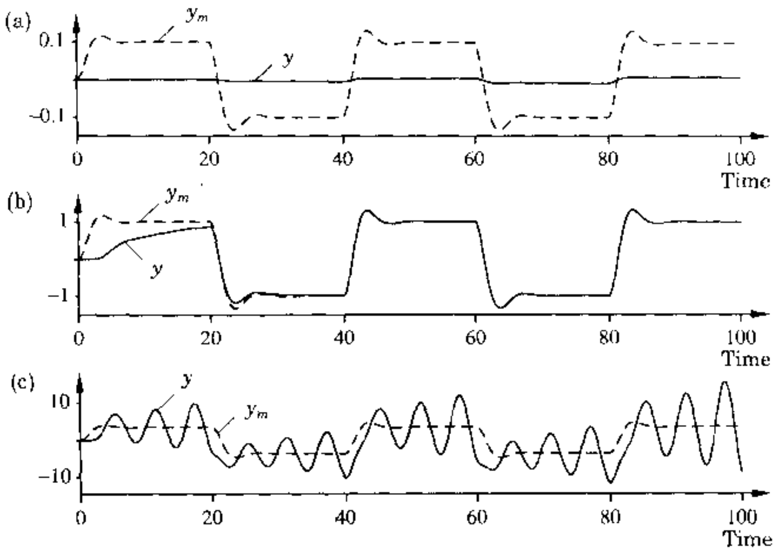

Figure 5.8 Simulation of the MRAS in Example 5.5. The command signal is a square wave with the amplitude (a) 0.1, (b) 1, and (c) 3.5. The model output y_m is a dashed line; the process output is a solid line. The following parameters are used: $k = a_1 = a_2 = \theta^0 = 1$, and $\gamma = 0.1$.

The example indicates clearly that the choice of adaptation gain is crucial and that the value chosen depends on the signal levels. Because of this it seems natural to modify the algorithm so that it does not depend on the signal levels. To do this, we will write the MIT rule as

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

where we have introduced $\varphi = -\partial e/\partial\theta$. Introduce the following modified adjustment rule:

$$\frac{d\theta}{dt} = \frac{\gamma \varphi e}{\alpha + \varphi^T \varphi} \tag{5.16}$$

where parameter $\alpha > 0$ is introduced to avoid difficulties when φ is small. Notice that we have written the equation in such a way that it also holds when θ is a vector; in that case, φ is also a vector of the same dimension.

If we repeat the analysis of the thought experiment, we find that Eq. (5.13) is replaced by

$$s + \gamma \frac{\varphi^o u_c^o}{\alpha + \varphi^{oT} \varphi^o} kG(s) = 0$$

Since φ^o is proportional to u_c^o, the roots of this equation will not change much with the signal levels. The adaptation rule given by Eq. (5.16) is called

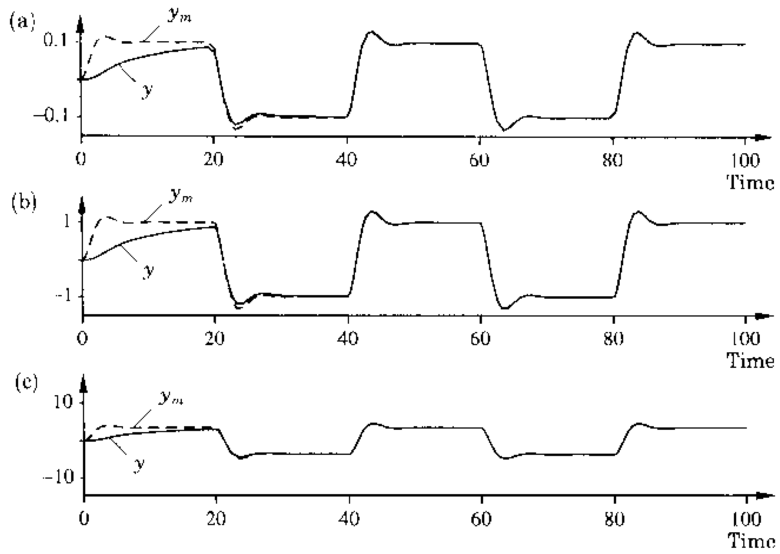

Figure 5.9 Simulation of the MRAS in Example 5.5 with the normalized MIT rule. The command signal is a square wave with the amplitude (a) 0.1, (b) 1, and (c) 3.5. Compare with Fig. 5.8. The model output y_m is a dashed line; the process output is a solid line. The parameters used are $k = a_1 = a_2 = \theta^0 = 1$, $\alpha = 0.001$, and $\gamma = 0.1$.

the *normalized MIT rule*. The improved performance with this algorithm is illustrated in Fig. 5.9. A comparison with Fig. 5.8 shows that normalization is useful.

Notice that the normalized adjustment rule performs very well even in the cases in which difficulties were encountered with the MIT rule. It is in fact possible to make the modified adjustment rule work very well over a wide range of command signal amplitudes. Notice that the normalization is obtained automatically with algorithms based on parameter estimation. (Compare with Example 2.16.)

Summary

Having derived the MIT rule and investigated some of its properties, we can now summarize some of the key issues. The model-reference control problem can be described as follows: Let the desired performance be specified by a reference model having the transfer function $G_m(s)$, and let the closed-loop transfer function of the plant be $G(s, \theta)$, where θ are the adjustable parameters. Furthermore, let u_c be the command signal. The model-reference adaptive system tries to change the controller parameters so that the error

$$e(t) = (G(p,\theta) - G_m(p))u_c(t)$$

goes to zero. The MIT rule given by

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

where $\varphi = \partial e/\partial\theta$ and γ is the adaptation gain, can be interpreted as a gradient method for minimizing the error. The MIT rule can be applied in many different cases; a few examples have been given in this section. The choice of the adaptation gain is critical and depends on the signal levels. The normalized algorithm

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

is less sensitive to signal levels. Notice that a normalization of a similar type is obtained automatically in the self-tuning regulator. Compare with Eq. (3.22).

Preliminary numerical experiments indicate that the systems obtained with the MIT rule work as expected for small adaptation gains. Very complex behavior may be obtained for high adaptation gains. To proceed to develop our understanding of adaptive systems, we will investigate the stability problem.

5.4 LYAPUNOV THEORY

There is no guarantee that an adaptive controller based on the MIT rule will give a stable closed-loop system. It is clearly desirable to see whether there are other methods for designing adaptive controllers that can guarantee the stability of the system. As a first step in this direction we now present the Lyapunov stability theory. For the benefit of students who are encountering Lyapunov theory for the first time, we first prove a stability theory for time-invariant systems. We then state a more powerful theorem for time-varying systems, which can be used to design adaptive controllers.

Lyapunov's Theory for Time-invariant Systems

Fundamental contributions to the stability theory for nonlinear systems were made by the Russian mathematician Lyapunov in the end of the nineteenth century. Lyapunov investigated the nonlinear differential equation

$$\frac{dx}{dt} = f(x) \qquad f(0) = 0 \tag{5.17}$$

Since $f(0) = 0$, the equation has the solution $x(t) = 0$. To guarantee that a solution exists and is unique, it is necessary to make some assumptions about

$f(x)$. A sufficient assumption is that $f(x)$ is locally Lipschitz, that is,

$$\|f(x) - f(y)\| \le L\|x - y\| \qquad L > 0$$

in the neighborhood of the origin. Lyapunov was interested in investigating whether the solution of Eq. (5.17) is stable with respect to perturbations. For this purpose he introduced the following stability concept.

DEFINITION 5.1 Lyapunov stability

The solution $x(t) = 0$ to the differential equation (5.17) is called *stable* if for given $\varepsilon > 0$ there exists a number $\delta(\varepsilon) > 0$ such that all solutions with initial conditions

$$\|x(0)\| < \delta$$

have the property

$$\|x(t)\| < \varepsilon \quad \text{for} \quad 0 \le t < \infty \tag{5.18}$$

The solution is *unstable* if it is not stable. The solution is *asymptotically stable* if it is stable and δ can be found such that all solutions with $\|x(0)\| < \delta$ have the property that $\|x(t)\| \to 0$ as $t \to \infty$. □

Remark 1. If the solution is asymptotically stable for any initial value, then it is said to be *globally asymptotically stable.*

Remark 2. Notice that Lyapunov stability refers to stability of a particular solution and not to the differential equation. □

Lyapunov developed a method for investigating stability that is based on the idea of finding a function with special properties. To describe these, we first introduce the notion of positive definite functions.

DEFINITION 5.2 Positive definite and semidefinite functions

A continuously differentiable function $V : R^n \to R$ is called *positive definite* in a region $U \subset R^n$ containing the origin if

1. $V(0) = 0$
2. $V(x) > 0, \quad x \in U$ and $x \ne 0$

A function is called *positive semidefinite* if Condition 2 is replaced by $V(x) \ge 0$. □

A positive definite function has level curves that enclose the origin. Curves corresponding to larger values of the function enclose curves that correspond to smaller values. The situation in the two-dimensional case is illustrated in Fig. 5.10. If we can find a function so that the velocity vector, $dx/dt = f(x)$, always points toward the interior of the level curves, then it seems intuitively clear that a solution that starts inside a given level curve can never pass to the outside of the same level curve. We have the following theorem.

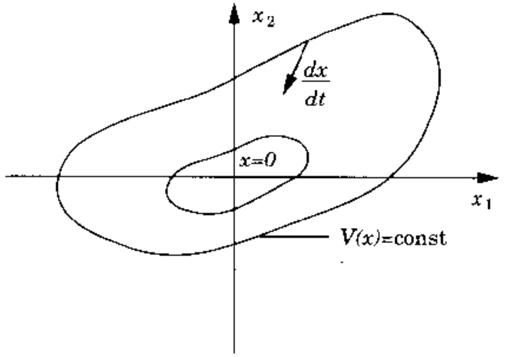

Figure 5.10 Illustration of Lyapunov's method for investigating stability.

THEOREM 5.1 Lyapunov's stability theorem: time-invariant systems
If there exists a function $V : R^n \to R$ that is positive definite such that its derivative along the solution of Eq. (5.17),

$$\frac{dV}{dt} = \frac{\partial V^T}{\partial x}\frac{dx}{dt} = \frac{\partial V^T}{\partial x} f(x) = -W(x) \tag{5.19}$$

is negative semidefinite, then the solution $x(t) = 0$ to Eq. (5.17) is stable. If dV/dt is negative definite, then the solution is also asymptotically stable. The function V is called a *Lyapunov function* for the system (5.17).

Moreover if

$$\frac{dV}{dt} < 0 \quad \text{and} \quad V(x) \to \infty \quad \text{when} \quad \|x\| \to \infty$$

then the solution is globally asymptotically stable.

Proof: Given $\varepsilon > 0$ such that $\{x \mid \|x\| \le \varepsilon\} \in U$, determine ℓ and δ such that

$$\ell = \min_{\|x\|=\varepsilon} V(x) = \max_{\|x\|\le\delta} V(x) \tag{5.20}$$

Consider initial conditions such that

$$\|x(0)\| < \delta$$

Since V is positive definite, it then follows from Definition 5.2 that

$$V(x(0)) < \ell$$

To prove that inequality (5.18) holds, we proceed by contradiction. Assume that t_1 is the smallest value such that $\|x(t_1)\| = \varepsilon$. It follows from Eq. (5.20) that

$$V(x(t_1)) \ge \ell$$

Furthermore,

$$V(x(t_1)) = V(x(0)) + \int_0^{t_1} \frac{dV}{dt}\,dt = V(x(0)) - \int_0^{t_1} W(x(s))\,ds \tag{5.21}$$

Since $W(x)$ is positive semidefinite, it follows that

$$V(x(t_1)) \le V(x(0)) < \ell$$

and we have thus obtained a contradiction and it can be concluded that $\|x(t)\| < \varepsilon$ for all t, which by Definition 5.1 implies that the solution $x(t) = 0$ is stable. To prove asymptotic stability, we notice that it follows from Eq. (5.21) that

$$0 \le \int_0^t W(x(s))\,ds = V(x(0)) - V(x(t)) \le \ell$$

Since $W(x)$ and $x(t)$ are continuous, it then follows that

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

If $W(x)$ is positive definite, this implies that $x(t) \to 0$ as $t \to \infty$. □

Remark. Notice that it follows from the proof that if the derivative of the Lyapunov function is negative semidefinite, the solution converges to the set $\{x \mid W(x) = 0\}$. □

Finding Lyapunov Functions

Lyapunov's theorem is very elegant. However, it is necessary to have methods for constructing Lyapunov functions. There is no universal method for constructing Lyapunov functions for a stable system. To apply the method, we therefore have to resort to trial and error. A good first attempt is to test quadratic functions. However, for linear systems we have the following important result.

THEOREM 5.2 Lyapunov functions for linear systems
Assume that the linear system

$$\frac{dx}{dt} = Ax \tag{5.22}$$

is asymptotically stable. Then for each symmetric positive definite matrix Q there exists a unique symmetric positive definite matrix P such that

$$A^TP + PA = -Q \tag{5.23}$$

Furthermore, the function

$$V(x) = x^T Px \tag{5.24}$$

is a Lyapunov function for Eq. (5.22).

Proof: Let Q be a symmetric positive definite matrix. Define

$$P(t) = \int_0^t e^{A^T(t-s)} Q e^{A(t-s)}\, ds$$

The matrix P is symmetric and positive definite because an integral of positive definite matrices is positive definite. The matrix P also satisfies

$$\frac{dP}{dt} = A^T P + PA + Q$$

Since the matrix A is stable, the limit

$$P_o = \lim_{t\to\infty} P(t)$$

exists. This matrix satisfies Eq. (5.23). It can also be shown that the solution to Eq. (5.23) is unique, which completes the argument. □

For a stable linear system we can thus always find a quadratic Lyapunov function. To use Theorem 5.2 to construct a Lyapunov function, we simply choose a positive matrix Q and solve the linear equation (5.23) for P. The following example shows how it can be done.

EXAMPLE 5.6 Lyapunov functions for a linear system

Consider the linear system (5.22) with

$$A = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$$

where it is assumed that all eigenvalues of A are in the left half-plane. Let the matrix Q be

$$Q = \begin{pmatrix} q_1 & 0 \\ 0 & q_2 \end{pmatrix}$$

where q_1 and q_2 are positive. Assume that the matrix P has the form

$$P = \begin{pmatrix} p_1 & p_2 \\ p_2 & p_3 \end{pmatrix}$$

Equation (5.23) then becomes

$$\begin{pmatrix} 2a_1 & 2a_3 & 0 \\ a_2 & a_1 + a_4 & a_3 \\ 0 & 2a_2 & 2a_4 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} -q_1 \\ 0 \\ -q_2 \end{pmatrix}$$

This is a linear equation. Theorem 5.2 implies that it always has a solution when A is stable and that the solution is a positive definite matrix P. □

Lyapunov Theory for Time-variable Systems

We now consider time-variable differential equations of the type

$$\frac{dx}{dt} = f(x,t) \tag{5.25}$$

The origin is an equilibrium point for Eq. (5.25) if $f(0,t) = 0\ \forall t \ge 0$. It is assumed that f is such that solutions exist for all $t \ge t_0$. To guarantee this, it is assumed that f is piecewise continuous in t and locally Lipschitz in x in a neighborhood of $x(t) = 0$. We now investigate the stability of the solution $x(t) = 0$.

In the time-varying case the solution will depend on t as well as on the starting time t_0. This implies that the bound δ in Definition 5.1 will depend on ε and t_0. The definition on stability can be refined to give uniform stability properties with respect to the initial time. We have the following definition.

DEFINITION 5.3 Uniform Lyapunov stability

The solution $x(t) = 0$ of Eq. (5.25) is *uniformly stable* if for $\varepsilon > 0$ there exists a number $\delta(\varepsilon) > 0$, independent of t_0, such that

$$\|x(t_0)\| < \delta \Rightarrow \|x(t)\| < \varepsilon \qquad \forall t \ge t_0 \ge 0$$

The solution is *uniformly asymptotically stable* if it is uniformly stable and there is $c > 0$, independent of t_0, such that $x(t) \to 0$ as $t \to \infty$, uniformly in t_0, for all $\|x(t_0)\| < c$. □

To state a stability theorem for solutions to Eq. (5.25), we first have to introduce the so-called *class K functions*.

DEFINITION 5.4 Class K function

A continuous function α: $[0,a) \to [0,\infty)$ is said to belong to *class K* if it is strictly increasing and $\alpha(0) = 0$. It is said to belong to class K_∞ if $a = \infty$ and $\alpha(r) \to \infty$ as $r \to \infty$. □

For time-varying systems the following stability theorem can now be stated.

THEOREM 5.3 Lyapunov's stability theorem: Time-varying systems

Let $x = 0$ be an equilibrium point for Eq. (5.25) and $D = \{x \in R^n \mid \|x\| < r\}$. Let V be a continuously differentiable function such that

$$\alpha_1(\|x\|) \le V(x,t) \le \alpha_2(\|x\|) \tag{5.26}$$

$$\frac{dV}{dt} = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(x,t) \le -\alpha_3(\|x\|)$$

for $\forall t \geq 0$, where α_1, α_2, and α_3 are class K functions. Then $x = 0$ is uniformly asymptotically stable.

Proof: A proof can be found in Khalil (1992). □

Remark 1. The derivative of V along the trajectories of Eq. (5.25) is now given by

$$\frac{dV}{dt} = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(x,t)$$

Remark 2. A function $V(x,t)$ satisfying the left inequality of (5.26) is said to be positive definite. A function satisfying the right inequality of (5.26) is said to be *decrescent.*

Remark 3. To show stability for time-variable systems, it is necessary to bound the function $V(x,t)$ by a function that doesn't depend on t. □

When using Lyapunov theory on adaptive control problems, we often find that dV/dt only is negative semidefinite. This implies that additional conditions must be imposed on the system. The following lemma gives a useful result.

LEMMA 5.1 Barbalat's lemma

If g is a real function of a real variable t, defined and uniformly continuous for $t \geq 0$, and if the limit of the integral

$$\int_0^t g(s)\, ds$$

as t tends to infinity exists and is a finite number, then

$$\lim_{t\to\infty} g(t) = 0$$ □

Remark. A consequence of Barbalat's lemma is that if $g \in L_2$ and dg/dt is bounded, then

$$\lim_{t\to\infty} g(t) = 0$$ □

When applying Lyapunov theory to an adaptive control problem, we get a time derivative of the Lyapunov function V, which depends on the control signal and other signals in the system. If these signals are bounded, Lemma 5.1 and the remark that follows can be used on dV/dt to prove stability. We have the following theorem.

THEOREM 5.4 Boundedness and convergence set

Let $D = \{x \in R^n \mid \|x\| < r\}$ and suppose that $f(x,t)$ is locally Lipschitz on $D \times [0,\infty)$. Let V be a continuously differentiable function such that

$$\alpha_1(\|x\|) \leq V(x,t) \leq \alpha_2(\|x\|)$$

and

$$\frac{dV}{dt} = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(x,t) \leq -W(x) \leq 0$$

$\forall t \geq 0$, $\forall x \in D$, where α_1 and α_2 are class K functions defined on $[0,r)$ and $W(x)$ is continuous on D. Further, it is assumed that dV/dt is uniformly continuous in t.

Then all solutions to Eq. (5.25) with $\|x(t_0)\| < \alpha_2^{-1}(\alpha_1(r))$ are bounded and satisfy

$$W(x(t)) \to 0 \quad \text{as} \quad t \to \infty$$

Moreover, if all the assumptions hold globally and α_1 belongs to class K_∞, the statement is true for all $x(t_0) \in R^n$. □

A proof of a slight modification of this theorem can be found in Khalil (1992). The theorem states that the states of the system are bounded and that they approach the set $\{x \in D \mid W(x) = 0\}$. In the theorem it is assumed that dV/dt is uniformly continuous, that is, that the continuity is independent of t. A sufficient condition for this is that $\ddot{V}$ is bounded.

5.5 DESIGN OF MRAS USING LYAPUNOV THEORY

We will now show how Lyapunov's stability theory can be used to construct algorithms for adjusting parameters in adaptive systems. To do this, we first derive a differential equation for the error, $e = y - y_m$. This differential equation contains the adjustable parameters. We then attempt to find a Lyapunov function and an adaptation mechanism such that the error will go to zero. When using the Lyaponov theory for adaptive systems, we find that dV/dt is usually only negative semidefinite. The procedure is to determine the error equation and a Lyapunov function with a bounded second derivative. Theorem 5.4 is then used to show boundedness and that the error goes to zero. To show parameter convergence, it is necessary to impose further conditions, such as persistently excitation and uniform observability, on the reference signal and the system. (See the references in the end of the chapter.) We start with a simple example.

EXAMPLE 5.7 First-order MRAS based on stability theory

Consider the problem in Example 5.2. The desired response is given by

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

where $a_m > 0$ and the reference signal is bounded. The process is described by

$$\frac{dy}{dt} = -ay + bu$$

The controller is

$$u = \theta_1 u_c - \theta_2 y$$

Introduce the error

$$e = y - y_m$$

Since we are trying to make the error small, it is natural to derive a differential equation for the error. We get

$$\frac{de}{dt} = -a_m e - (b\theta_2 + a - a_m)y + (b\theta_1 - b_m)u_c$$

Notice that the error goes to zero if the parameters are equal to the values given by Eqs. (5.8). We will now attempt to construct a parameter adjustment mechanism that will drive the parameters θ_1 and θ_2 to their desired values. For this purpose, assume that $b\gamma > 0$ and introduce the following quadratic function:

$$V(e, \theta_1, \theta_2) = \frac{1}{2}\left(e^2 + \frac{1}{b\gamma}(b\theta_2 + a - a_m)^2 + \frac{1}{b\gamma}(b\theta_1 - b_m)^2\right)$$

This function is zero when e is zero and the controller parameters are equal to the correct values. For the function to qualify as a Lyapunov function the derivative dV/dt must be negative. The derivative is

$$\begin{aligned}\frac{dV}{dt} &= e\frac{de}{dt} + \frac{1}{\gamma}(b\theta_2 + a - a_m)\frac{d\theta_2}{dt} + \frac{1}{\gamma}(b\theta_1 - b_m)\frac{d\theta_1}{dt} \\ &= -a_m e^2 + \frac{1}{\gamma}(b\theta_2 + a - a_m)\left(\frac{d\theta_2}{dt} - \gamma y e\right) \\ &\quad + \frac{1}{\gamma}(b\theta_1 - b_m)\left(\frac{d\theta_1}{dt} + \gamma u_c e\right)\end{aligned}$$

If the parameters are updated as

$$\begin{aligned}\frac{d\theta_1}{dt} &= -\gamma u_c e \\ \frac{d\theta_2}{dt} &= \gamma y e\end{aligned} \tag{5.27}$$

we get

$$\frac{dV}{dt} = -a_m e^2$$

The derivative of V with respect to time is thus negative semidefinite but not negative definite. This implies that $V(t) \le V(0)$ and thus that e, θ_1, and θ_2 must be bounded. This implies that $y = e + y_m$ also is bounded. To use Theorem 5.4, we determine

$$\frac{d^2V}{dt^2} = -2a_m e\frac{de}{dt} = -2a_m e\left(-a_m e - (b\theta_2 + a - a_m)y + (b\theta_1 - b_m)u_c\right)$$

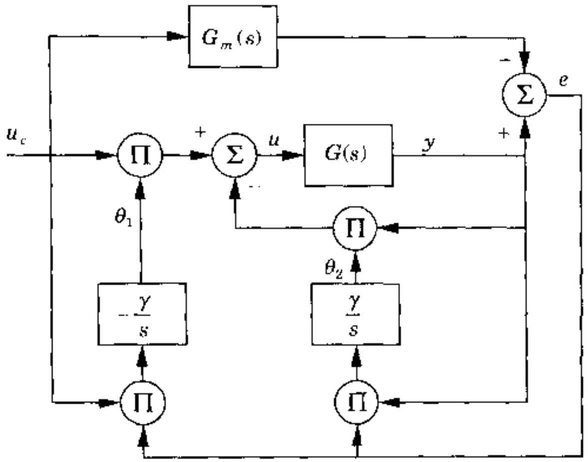

Figure 5.11 Block diagram of an MRAS based on Lyapunov theory for a first-order system. Compare with the controller based on the MIT rule for the same system in Fig. 5.4.

Since u_c, e, and y are bounded, it follows that $\ddot{V}$ is bounded; hence dV/dt is uniformly continuous. From Theorem 5.4 it now follows that the error e will go to zero. However, the parameters will not necessarily converge to their correct values; it is shown only that they are bounded. To have parameter convergence, it is necessary to impose conditions on the excitation of the system. (Compare with Example 5.3.)

The adaptation rule given by Eqs. (5.27) is similar to the MIT rule given by Eqs. (5.9), but the sensitivity derivatives are replaced by other signals. A block diagram of the system is shown in Fig. 5.11. Compare with the corresponding block diagram for the system with the MIT rule in Fig. 5.4. The only difference is that there is no filtering of the signals u_c and y with the Lyapunov rule. In both cases the adjustment law can be written as

$$\frac{d\theta}{dt} = \gamma\varphi e \tag{5.28}$$

where θ is a vector of parameters and

$$\varphi = \begin{pmatrix} -u_c & y \end{pmatrix}^T$$

for the Lyapunov rule and

$$\varphi = \frac{a_m}{p + a_m}\begin{pmatrix} -u_c & y \end{pmatrix}^T$$

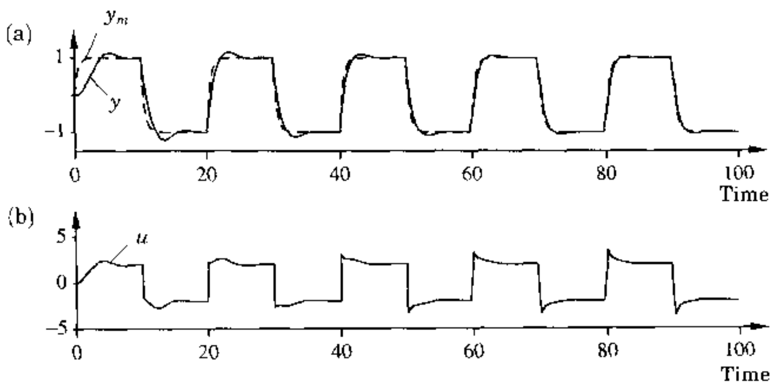

Figure 5.12 Simulation of the system in Example 5.7 using an adaptive controller based on Lyapunov theory. The parameter values are $a = 1$, $b = 0.5$, $a_m = b_m = 2$, and $\gamma = 1$. (a) Process (solid line) and model (dashed line) outputs. (b) Control signal.

for the MIT rule. The adjustment rule obtained from Lyapunov theory is simpler because it does not require filtering of the signals. Figure 5.12 shows a simulation of the system for the case $G(s) = 0.5/(s+1)$ and $G_m(s) = 2/(s+2)$. The behavior is quite similar to that obtained with the MIT rule in Fig. 5.5. Notice, however, that arbitrary large values of the adaptation gain γ can be used with the Lyapunov approach.

Figure 5.13 shows the parameter estimates in the simulation for different values of adaptation gain γ. For comparison we have also shown the parameters obtained with the MIT rule. □

State Space Systems

We will now show how Lyapunov's theory can be used to derive stable MRASs for general linear systems. The idea is the same as used previously. It can be described as follows:

1. Find a controller structure.
2. Derive the error equation.
3. Find a Lyapunov function and use it to derive a parameter updating law such that the error will go to zero.

Consider a linear system described by

$$\frac{dx}{dt} = Ax + Bu \tag{5.29}$$

Assume that it is desired to find a control law so that the response to command

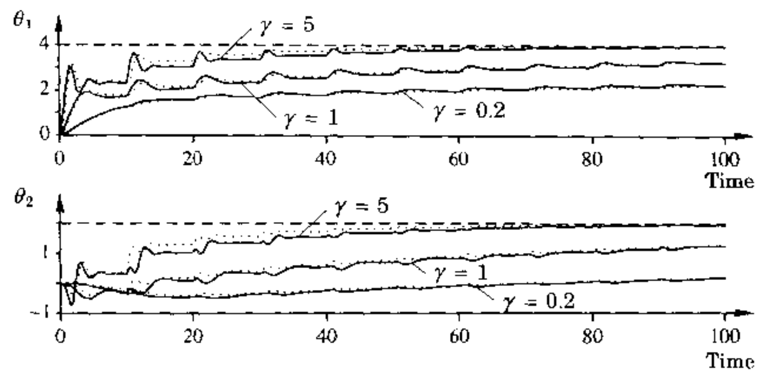

Figure 5.13 Controller parameters θ_1 and θ_2 for the system in Example 5.7 when $\gamma = 0.2$, 1, and 5. The dotted lines are the parameters obtained with the MIT rule. Compare Fig. 5.6.

signals is given by

$$\frac{dx_m}{dt} = A_m x_m + B_m u_c \tag{5.30}$$

A general linear control law for the system given by Eq. (5.29) is

$$u = Mu_c - Lx \tag{5.31}$$

The closed-loop system then becomes

$$\frac{dx}{dt} = (A - BL)x + BMu_c = A_c(\theta)x + B_c(\theta)u_c \tag{5.32}$$

The control law can be parameterized in different ways. All parameters in the matrices L and M may be chosen freely. There may also be constraints among the parameters. The general case can be captured by assuming that the closed-loop system is described by Eq. (5.32), where matrices A_c and B_c depend on a parameter θ.

Compatibility conditions It is not always possible to find parameters θ such that Eq. (5.32) is equivalent to Eq. (5.30). A sufficient condition is that there exists a parameter value θ^0 such that

$$\begin{aligned} A_c(\theta^0) &= A_m \\ B_c(\theta^0) &= B_m \end{aligned} \tag{5.33}$$

This condition for perfect model-following is fairly stringent. When all parameters in the control law can be chosen freely, it implies that

$$\begin{aligned} A - A_m &= BL \\ B_m &= BM \end{aligned}$$

This means that the columns of matrices $A - A_m$ and B_m are linear combinations of the columns of matrix B. If these conditions are satisfied and the columns of B and B_m are linearly independent, then the matrices L and M are given by

$$L = (B^TB)^{-1}B^T(A - A_m) = (B_m^TB)^{-1}B_m^T(A - A_m)$$
$$M = (B^TB)^{-1}B^TB_m = (B_m^TB)^{-1}B_m^TB_m$$

The error equation Introduce the error defined as

$$e = x - x_m$$

Subtracting Eq. (5.30) from Eq. (5.29) gives

$$\frac{de}{dt} = \frac{dx}{dt} - \frac{dx_m}{dt} = Ax + Bu - A_mx_m - B_mu_c$$

Adding and subtracting A_mx from the right-hand side give

$$\begin{aligned}\frac{de}{dt} &= A_me + (A - A_m - BL)x + (BM - B_m)u_c\\ &= A_me + (A_c(\theta) - A_m)x + (B_c(\theta) - B_m)u_c\\ &= A_me + \Psi(\theta - \theta^0)\end{aligned} \tag{5.34}$$

To obtain the last equality, it has been assumed that the conditions for exact model-following are satisfied. This is required for θ^0 to exist. To derive a parameter adjustment law, we introduce the Lyapunov function

$$V(e,\theta) = \frac{1}{2}\left(\gamma e^TPe + (\theta - \theta^0)^T(\theta - \theta^0)\right)$$

where P is a positive definite matrix. The function V is positive definite. To find out whether it can be a Lyapunov function, we calculate its total time derivative

$$\begin{aligned}\frac{dV}{dt} &= -\frac{\gamma}{2}e^TQe + \gamma(\theta - \theta^0)\Psi^TPe + (\theta - \theta^0)^T\frac{d\theta}{dt}\\ &= -\frac{\gamma}{2}e^TQe + (\theta - \theta^0)^T\left(\frac{d\theta}{dt} + \gamma\Psi^TPe\right)\end{aligned}$$

where Q is positive definite and such that

$$A_m^TP + PA_m = -Q$$

Notice that it follows from Theorem 5.2 that a pair of positive definite matrices P and Q with this property always exist if A_m is stable.

If the parameter adjustment law is chosen to be

$$\frac{d\theta}{dt} = -\gamma\Psi^TPe \tag{5.35}$$

we get

$$\frac{dV}{dt} = -\frac{\gamma}{2}e^TQe$$

The time derivative of the Lyapunov function is negative semidefinite. By using Lemma 5.1 in the same way as in Example 5.7 it can be shown that the error goes to zero. Notice that we have assumed that all states x are measurable.

Adaptation of a Feedforward Gain

We now attempt to use Lyapunov theory to derive parameter adjustment laws for the problem of adjusting a feedforward gain. We consider the case in which the plant has transfer function $kG(s)$, where $G(s)$ is known and k is unknown. The desired response is given by the transfer function $k_0G(s)$. This problem was discussed previously in Examples 5.1 and 5.3. The error is given by

$$e = (kG(p)\theta - k_0G(p))u_c = kG(p)(\theta - \theta^0)u_c$$

where $\theta^0 = k_0/k$. To use Lyapunov theory, we first introduce a state space representation of the transfer function G. The relation between the parameter θ and the error e can then be written as

$$\begin{aligned}\frac{dx}{dt} &= Ax + B(\theta - \theta^0)u_c\\ e &= Cx\end{aligned} \tag{5.36}$$

If the homogeneous system $\dot{x} = Ax$ is asymptotically stable, there exist positive definite matrices P and Q such that

$$A^TP + PA = -Q \tag{5.37}$$

Choose the following function as a candidate for a Lyapunov function:

$$V = \frac{1}{2}\left(\gamma x^TPx + (\theta - \theta^0)^2\right)$$

The time derivative of V along the differential equation (Eqs. 5.36) is given by

$$\frac{dV}{dt} = \frac{\gamma}{2}\left(\frac{dx^T}{dt}Px + x^TP\frac{dx}{dt}\right) + (\theta - \theta^0)\frac{d\theta}{dt}$$

Using Eqs. (5.36), we get

$$\begin{aligned}\frac{dV}{dt} &= \frac{\gamma}{2}\left(\left(Ax + Bu_c(\theta - \theta^0)\right)^TPx + x^TP\left(Ax + Bu_c(\theta - \theta^0)\right)\right)\\ &\quad + (\theta - \theta^0)\frac{d\theta}{dt}\\ &= -\frac{\gamma}{2}x^TQx + (\theta - \theta^0)\left(\frac{d\theta}{dt} + \gamma u_cB^TPx\right)\end{aligned}$$

If the parameter adjustment law is chosen to be

$$\frac{d\theta}{dt} = -\gamma u_c B^T P x \tag{5.38}$$

we find that the derivative of the Lyapunov function will be negative as long as $x \neq 0$. The state vector x and the error $e = Cx$ will go to zero as t goes to infinity. Notice, however, that the parameter error $\theta - \theta^0$ will not necessarily go to zero.

Output feedback The result obtained is quite restrictive because it requires that all state variables are known. A parameter adjustment law that uses output feedback can be obtained if the Lyapunov function can be chosen so that

$$B^T P = C$$

where C is the output matrix of the system in Eq. (5.34). With this choice of P it follows that

$$B^T P x = Cx = e$$

and the adjustment rule becomes

$$\frac{d\theta}{dt} = -\gamma u_c e$$

The appropriate condition is given by the celebrated Kalman-Yakubovich lemma. The following definition is needed to state this lemma.

DEFINITION 5.5 Positive real transfer function

A rational transfer function G with real coefficients is *positive real* (PR) if

$$\operatorname{Re} G(s) \geq 0 \quad \text{for} \quad \operatorname{Re} s \geq 0 \tag{5.39}$$

A transfer function G is *strictly positive real* (SPR) if $G(s - \varepsilon)$ is positive real for some real $\varepsilon > 0$. □

The concept of SPR is discussed further in Section 5.6. Let it suffice to mention that $G(s) = 1/(s+1)$ is SPR and $G(s) = 1/s$ is PR but not SPR. The following result gives a state space interpretation of SPR.

LEMMA 5.2 Kalman-Yakubovich lemma

Let the time-invariant linear system

$$\frac{dx}{dt} = Ax + Bu$$
$$y = Cx$$

be completely controllable and completely observable. The transfer function

$$G(s) = C(sI - A)^{-1}B$$

is strictly positive real if and only if there exist positive definite matrices P and Q such that

$$A^T P + PA = -Q$$

and

$$B^T P = C$$ □

A proof of this result is given in Section 5.6. There is a more general version of the theorem that applies to systems with a direct term from input to output. The simpler version is sufficient for our purposes.

THEOREM 5.5 MRAS using the Lyapunov rule

Consider the problem of adapting a feedforward gain. Assume that the transfer function G is strictly positive real. Then the parameter adjustment rule

$$\frac{d\theta}{dt} = -\gamma u_c e \tag{5.40}$$

where γ is a positive constant, makes the output error e in Eqs. (5.36) go to zero. □

The control law of Eq. (5.40) is very similar to the control law obtained by the MIT rule, Eq. (5.5). This is illustrated in Fig. 5.14, which shows block diagrams of both systems. The only difference between the systems is that the connection to the first multiplier comes from the model output for the MIT rule and from the command signal for the Lyapunov rule. This seemingly small difference has major consequences, however.

A remark on the assumptions It may seem strange that such drastically different behaviors can be obtained by minor modifications of the system. It also seems strange that it is possible to use arbitrarily high adaptation gains. This is because the assumption that a transfer function is positive real is very strong. It follows from Definition 5.5 that Re $G(i\omega) \geq 0$ if the transfer function $G(s)$ is positive real. This means that the Nyquist curve of G is in the right half-plane. Such a system is stable under proportional feedback with arbitrarily high gain. The closed-loop system can be made arbitrarily insensitive to the gain variations. The result is of limited practical value because of the strong assumptions that are made.

Summary

In this section we have shown that it is possible to construct parameter adjustment rules based on Lyapunov's stability theory. The adjustment rules obtained in this way guarantee that the error goes to zero, but it cannot be asserted that the parameters converge to their correct values. The adjustment rules obtained are similar to those obtained by the MIT rule. However, the rules

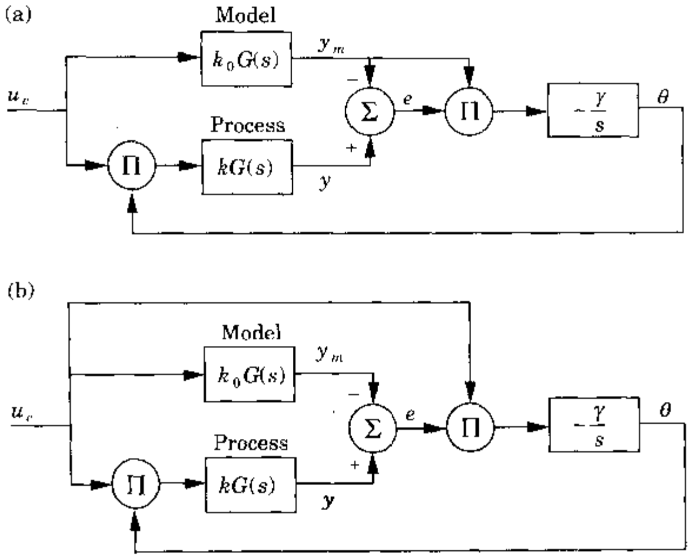

Figure 5.14 Block diagrams of the adaptive systems for feedforward gain compensation obtained by (a) the MIT rule and (b) the Lyapunov rule.

are not normalized. The adjustment rules have the remarkable property that arbitrarily high adaptation gains can be used. This property depends on the strong assumptions that are made. This is discussed further in Chapter 6.

5.6 BOUNDED-INPUT, BOUNDED-OUTPUT STABILITY

Systems can be described from two points of view: the internal or state space view or the external or input-output view. The state space approach is based on a detailed description of the inner structure of the system. In the input-output approach, a system is considered to be a black box that transforms inputs to outputs. In Section 5.5 we approached stability from the state space view. In this section we develop stability theory from the input-output view. In the next section the results are applied to design of adaptive controllers.

We start with a brief presentation of the operator view of dynamical systems. This leads naturally to the concept of bounded-input, bounded-output (BIBO) stability. The fundamental results like the small gain theorem and the passivity theorem are then presented. In Section 5.5 we found that the notion of positive real was essential. This notion, which is closely related to passivity, will also be discussed.

The Operator View of Dynamical Systems

Signals are elements of a normed space X, which we call the signal space. A system S is considered as an operator $S : X \to X$. For simplicity we consider systems with one input and one output and the signals are real functions from R to R. Several choices of norms are considered, for example, the L_2 norm

$$\|u\| = \left(\int_{-\infty}^{\infty} u^2(t)\,dt\right)^{\frac{1}{2}}$$

or the sup norm

$$\|u\| = \sup_{0 \le t \le \infty} |u(t)|$$

A drawback of using L_2 is that it must be assumed *a priori* that all signals go to zero as $t \to \infty$. The notion of extended space is introduced to avoid this assumption. This is introduced as follows.

Let Y be the space of real-valued functions on $[0, \infty)$. Let x be an element of Y. The *truncation* of x at $T > 0$ is defined as

$$x_T(t) = \begin{cases} x(t) & 0 \le t \le T \\ 0 & t > T \end{cases}$$

DEFINITION 5.6 Extended space

If X is a normed linear subspace of Y, then the extended space X_e is the set $\{x \in Y \mid x_T \in X \text{ for some fixed } T \ge 0\}$. □

The extended L_2 space is denoted L_{2e}. There is now a simple way to introduce the notion of the gain of a system.

DEFINITION 5.7 Gain of a nonlinear system

Let the signal space be X_e. The *gain* $\gamma(S)$ *of a system* S is defined as

$$\gamma(S) = \sup_{u \in X_e} \frac{\|Su\|}{\|u\|}$$

where u is the input signal to the system.

Remark. The gain is thus the smallest value such that

$$\|Su\| \le \gamma(S)\|u\| \quad \text{for all } u \in X_e$$

We use supremum because the maximum of $\|Su\|/\|u\|$ may not be assumed for a signal in the class that we are considering. □

We illustrate the definition with a few examples.

EXAMPLE 5.8 Linear systems with signals in L_{2e}

Let the signal space be L_{2e}. Consider a linear system with the transfer function $G(s)$. Assume that $G(s)$ has no poles in the closed right half-plane and that the system is initially at rest. Let u be the input and y the output, and let U and Y be the corresponding Laplace transforms. It follows from Parseval's theorem, Theorem 2.8, that

$$\begin{aligned}\|y\|^2 &= \int_0^\infty y^2(t)\,dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} Y(i\omega)Y(-i\omega)\,d\omega \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} G(i\omega)U(i\omega)G(-i\omega)U(-i\omega)\,d\omega \\ &\le \max_\omega |G(i\omega)|^2 \frac{1}{2\pi}\int_{-\infty}^{\infty} U(i\omega)U(-i\omega)\,d\omega \\ &= \max_\omega |G(i\omega)|^2 \int_0^\infty u^2(t)\,dt = \max_\omega |G(i\omega)|^2 \cdot \|u\|^2\end{aligned}$$

Hence

$$\|y\| \le \max_\omega |G(i\omega)| \cdot \|u\|$$

The gain is thus less than $\max |G(i\omega)|$. We get equality in the above equation if u is a sinusoid with the frequency that maximizes $|G(i\omega)|$. However, such a signal is not in L_{2e}. The value of $\|y\|$ can be made arbitrarily close to $\max |G(i\omega)|$ with a truncated sinusoid in L_{2e} by making T sufficiently large. The gain of the system is thus

$$\gamma(G) = \max_\omega |G(i\omega)| \tag{5.41}$$

□

EXAMPLE 5.9 Linear system with sup norm

Consider a stable linear system with impulse response $h(t)$. We have

$$y(t) = \int_0^\infty h(\tau)u(t-\tau)\,d\tau$$

Using the sup norm, we get

$$|y(t)| = \left|\int_0^\infty h(\tau)u(t-\tau)\,d\tau\right| \le \sup_t |u(t)| \int_0^\infty |h(\tau)|\,d\tau$$

This gives

$$\sup_t |y(t)| \le \gamma(G)\cdot \sup_t |u(t)|$$

where the gain of the system is given by

$$\gamma(G) = \int_0^\infty |h(\tau)|\,d\tau$$

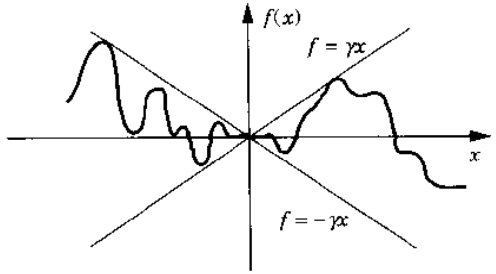

Figure 5.15 Illustration of the gain of a static nonlinearity.

If we let $u_0 = \max_t |u(t)|$, the maximum is assumed for the signal

$$u(s) = u_0 \operatorname{sign}(h(t-s))$$

However, this signal is not in L_{2e}. Since the system is stable, we can get arbitrarily close with a signal in L_{2e} by making T sufficiently large. □

EXAMPLE 5.10 Static nonlinear system

Consider a static system that is described by the nonlinear equation

$$y(t) = f(u(t))$$

For all norms we have

$$|y(t)| \le \max_u |f(u(t))|$$

The gain of the system is thus given by

$$\gamma = \max_u \frac{|f(u)|}{|u|}$$

The gain of a static system has a simple interpretation. A function whose norm is γ can be bounded between the straight lines $y = \pm\gamma u$, as is illustrated in Fig. 5.15. □

Having defined the gain of a system, we can now define stability.

DEFINITION 5.8 BIBO stability

A system is called *bounded-input, bounded-output (BIBO) stable* if the system has bounded gain. □

Notice that this definition refers to stability of a system and not stability of a particular solution. Also notice that a system with bounded gain is BIBO stable but that the converse is not true. The static system $y = u^2$ does not have finite gain, but it is BIBO stable.

Stability Criteria

Having defined the notion of stability, we now give criteria for stability. For this purpose, consider the simple feedback system in Fig. 5.16. We are interested in determining when the gain from u to y is bounded. We have the following theorem.

THEOREM 5.6 The small gain theorem

Consider the system in Fig. 5.16. Let γ_1 and γ_2 be the gains of the systems H_1 and H_2. The closed-loop system is BIBO stable if

$$\gamma_1\gamma_2 < 1 \tag{5.42}$$

and its gain is less than

$$\gamma = \frac{\gamma_1}{1-\gamma_1\gamma_2} \tag{5.43}$$

Outline of proof: For a rigorous proof it must first be established that y exists. If this is true, we have

$$y = H_1 e = H_1(u - H_2 y)$$

Hence

$$\|y\| \le \|H_1 u\| + \|H_1 H_2 y\| \le \gamma_1\|u\| + \gamma_1\gamma_2\|y\|$$

Because of Eq. (5.42) we can solve for $\|y\|$. Hence

$$\|y\| \le \frac{\gamma_1}{1-\gamma_1\gamma_2}\|u\| = \gamma\|u\|$$

which proves BIBO stability and gives the expression (5.43) for the gain of the system. □

Remark 1. The result has a strong intuitive interpretation. It simply says that if the total gain around the loop is less than 1, then the closed-loop system is stable.

Remark 2. For the special case of linear systems with L_2 norms it follows from Example 5.8 that the gain is the maximum magnitude of the transfer function.

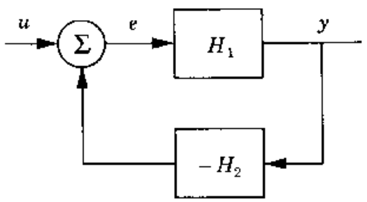

Figure 5.16 Block diagram of a simple feedback loop.

The theorem can be interpreted as an extension of the Nyquist theorem. The condition (5.42) implies that the loop gain is always less than 1. From this interpretation we can also conclude that the result is quite conservative.

Passivity

We now present another stability theorem that is also based on the input-output point of view. The starting point is the notion of passivity, which is an abstract formulation of the idea of energy dissipation. Passive systems are common in engineering. A system composed only of components like resistors, capacitors, and inductors is one example from electrical engineering. A system composed of masses, springs, and dashpots is an example from mechanical engineering. When dealing with electrical systems, we will consider two-port systems in which the current is the input and the voltage is the output. The same concepts apply to mechanical systems, in which the variables are position and force.

Passivity is naturally associated with power dissipation. Such a concept can be defined for linear as well as nonlinear systems. Roughly speaking, the passivity theorem says that a feedback connection of one passive system and one strictly passive system is stable. To state the result formally, we need an abstract notion of passivity. We start with the operator view of systems, in which a system is represented by an operator mapping signals to signals. The signal space is assumed to be L_{2e} with a scalar product defined by

$$\langle x \mid y \rangle = \int_0^\infty x(s)y(s)\,ds = \int_0^T x(s)y(s)\,ds$$

We have the following definition.

DEFINITION 5.9 Passive system

A system with input u and output y is *passive* if

$$\langle y \mid u \rangle \ge 0$$

The system is *input strictly passive* (ISP) if there exists $\varepsilon > 0$ such that

$$\langle y \mid u \rangle \ge \varepsilon\|u\|^2$$

and *output strictly passive* (OSP) if there exists $\varepsilon > 0$ such that

$$\langle y \mid u \rangle \ge \varepsilon\|y\|^2$$

□

Notice that in electrical systems the power is proportional to the product of current and voltage. The definition is thus a very natural abstraction. The following example illustrates the definition of passivity.

EXAMPLE 5.11 Static nonlinear systems

Consider a static nonlinear system characterized by the function $f : R \to R$. We have

$$\langle y \mid u \rangle = \int_0^{\infty} f(u(t))u(t)\, dt$$

The right-hand side is thus nonnegative if

$$xf(x) \geq 0 \tag{5.44}$$

which is the condition for passivity. This condition means that the graph of the curve f is entirely in the first and the third quadrants. The system is input strictly passive if

$$xf(x) \geq \delta |x|^2$$

It is output strictly passive if

$$xf(x) \geq \delta f^2(x)$$

A static system with $f(x) = x + x^3$ is thus input strictly passive, and a static system with $f(x) = x/(1 + |x|)$ is output strictly passive. □

Positive Real Functions

For linear systems the concept of passivity is closely related to the properties positive real and strictly positive real introduced in Definition 5.5 in Section 5.5. The notion of positive real did actually originate from an effort to characterize driving point impedance functions for linear circuits composed of passive components. The driving point impedance function is the transfer function from current to voltage across two terminals in a circuit. The driving point admittance function is the transfer function from voltage to current. In circuit theory it was established that such impedance functions have certain properties that were taken as the definition of positive real. In this section we discuss some properties of positive real functions. It follows from Definition 5.5 that if the transfer function $G(s)$ is PR (SPR), then its inverse $1/G(s)$ is also PR (SPR). This is a direct consequence of the symmetry of admittance functions and impedance functions. It does not matter whether we consider current or voltage as the input to a circuit. Positive real functions can be characterized in many different ways. An alternative to Definition 5.5 that is easier to use is given by the following theorem.

THEOREM 5.7 Conditions for positive realness

A rational transfer function $G(s)$ with real coefficients is PR if and only if the following conditions hold.

(i) The function has no poles in the right half-plane.

(ii) If the function has poles on the imaginary axis or at infinity, they are simple poles with positive residues.

(iii) The real part of G is nonnegative along the $i\omega$ axis, that is,

$$\operatorname{Re}(G(i\omega)) \geq 0 \tag{5.45}$$

A transfer function is SPR if conditions (i) and (iii) hold and if condition (ii) is replaced by the condition that $G(s)$ has no poles or zeros on the imaginary axis.

Proof: Assume that $G(s)$ is PR. Since it is rational, the only singularities are poles. A function assumes all values around a pole. According to Definition 5.5 the function has positive real part for $\operatorname{Re} s \geq 0$. Hence it cannot have poles in this region. Equation (5.45) follows by setting $s = i\omega$ in Definition 5.5. Furthermore, $G(s)$ cannot have multiple poles at infinity because the condition $\operatorname{Re} G(s) \geq 0$ for $\operatorname{Re} s \geq 0$ would then be violated. For the same reason a pole at infinity must also have positive residue.

We have thus shown the necessity. To show sufficiency, we use the fact that a function that is analytic in a region assumes its largest values on the boundary. Consider the function

$$F(s) = e^{G(s)}$$

We have

$$|F(s)| = e^{\operatorname{Re} G(s)} \tag{5.46}$$

Let the region D be bounded by the imaginary axis and an infinite half-circle to the right with the imaginary axis as a diameter. Let Γ be the boundary of D. Assume that conditions (i), (ii), and (iii) hold. Because of condition (iii) we have $|F(s)| > 1$ on the imaginary axis. It now remains to investigate the value of F on the large half-circle. It follows from condition (ii) that G has at most one pole at infinity. We have three cases: $G(s)$ may go to zero; it may go to a constant, which must be positive because of condition (iii); or it may go to infinity as ks, where the constant k must be positive because of condition (ii). We can thus conclude that $|F(s)| > 1$ on Γ. Since F is analytic in D, the condition then also holds on D, and Eq. (5.45) then follows. Notice that it also follows that the function $G(s)$ does not have any zeros inside D. □

We now illustrate the different passivity concepts on linear time-invariant systems.

EXAMPLE 5.12 Linear time-invariant systems

Consider a linear time-invariant system with the transfer function $G(s)$. Assume that $G(s)$ has no poles in the closed right half-plane. It follows from

Parseval's theorem that

$$\langle y \mid u\rangle = \int_0^\infty y(t)u(t)\,dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} Y(i\omega)U(-i\omega)\,d\omega$$
$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} G(i\omega)U(i\omega)U(-i\omega)\,d\omega$$
$$= \frac{1}{\pi}\int_0^{\infty} \operatorname{Re}\{G(i\omega)\}\,U(i\omega)U(-i\omega)\,d\omega \tag{5.47}$$

where Y and U are the Laplace transforms of y and u, respectively. If $G(i\omega)$ is positive real (see Definition 5.5), we have $\operatorname{Re} G(i\omega) \geq 0$, and we get

$$\langle y \mid u\rangle \geq 0$$

which shows that the system is passive. It follows from Definition 5.9 that a positive real transfer function is input strictly passive if

$$\operatorname{Re} G(i\omega) \geq \varepsilon > 0$$

and output strictly passive if

$$\operatorname{Re} G(i\omega) \geq \varepsilon |G(i\omega)|^2$$

The transfer function $G(s) = s + 1$ is thus SPR and ISP but not OSP. The transfer function $G(s) = 1/(s+1)$ is SPR and OSP but not ISP. The transfer function

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

is OSP and ISP but not SPR. □

In control systems applications it is common for transfer functions to be proper or strictly proper. The output strict passivity is therefore the concept that is normally used in these applications.

Proof of the Kalman-Yakubovich Lemma

Having developed the notion of SPR, we can now give a proof of the Kalman-Yakubovic lemma, which was given as Lemma 5.2 in Section 5.5. Consider the linear system

$$\frac{dx}{dt} = Ax + Bu$$
$$y = Cx \tag{5.48}$$

which is assumed to be completely controllable and completely observable. The system has the transfer function

$$G(s) = C(sI - A)^{-1}B \tag{5.49}$$

We will prove that a necessary and sufficient condition for $G(s)$ to be SPR is that there exist positive definite matrices P and Q such that

$$A^T P + PA = -Q \tag{5.50}$$

and

$$B^T P = C \tag{5.51}$$

We will first prove necessity. If we use $V = x^T P x$ as a Lyapunov function, it follows from Theorem 5.1 that the system (5.48) is stable. This implies that the transfer function $G(s)$ is analytic in the closed right half-plane. To prove that $G(s)$ is SPR, it remains to verify condition (iii) in Theorem 5.7. It follows from Eq. (5.50) that

$$-sP - A^T P + sP - PA = (-sI - A)^T P + P(sI - A) = Q$$

To obtain this equation, we have added and subtracted sP. Multiplying the equation with $B^T(-sI - A)^{-T}$ from the left and $(sI - A)^{-1}B$ from the right gives

$$B^T P(sI-A)^{-1}B + B^T(-sI-A)^{-T}PB = B^T(-sI-A)^{-T}Q(sI-A)^{-1}B \tag{5.52}$$

Since $G^T(-s) = G(-s)$, Eq. (5.49) now implies that

$$2\operatorname{Re} G(i\omega) = G(i\omega) + G(-i\omega) = B^T(-i\omega I - A)^{-T}Q(i\omega I - A)^{-1}B \geq 0$$

It now follows from Theorem 5.7 that $G(s)$ is PR. Replacing s by $s - \varepsilon$ in the above calculations, we find in a similar way that

$$\operatorname{Re} G(i\omega - \varepsilon) \geq 0$$

Since the matrix A has all its eigenvalues in the open left half-plane, it follows that the matrix $A + \varepsilon I$ is also stable. It now follows from Theorem 5.7 that $G(s)$ is SPR.

To prove sufficiency, we start with the assumption that the system (5.48) has a transfer function $G(s)$ that is SPR. The proof is based on a direct construction of the matrices P and Q. Consider the expression

$$G(s) + G(-s) = \frac{B(s)}{A(s)} + \frac{B(-s)}{A(-s)} = \frac{A(-s)B(s) + A(s)B(-s)}{A(s)A(-s)} = \frac{Q(s)}{A(s)A(-s)}$$

where

$$Q(s) = q_1(-1)^{n-1}s^{2(n-1)} + q_2(-1)^{n-2}s^{2(n-2)} + \ldots + q_n$$

Notice that polynomial $Q(s)$ has only terms of even power and that all coefficients q_i are positive, since $G(s)$ is SPR. Let Q be a diagonal matrix with elements q_i. Introduce the following realization of the transfer function:

$$\frac{dx}{dt} = \begin{pmatrix} -a_1 & -a_2 & \cdots & -a_{n-1} & -a_n \\ 1 & 0 & & 0 & 0 \\ \vdots & & & \vdots & \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} u$$

With this choice we have

$$(sI - A)^{-1}B = \frac{1}{A(s)}\begin{pmatrix} s^{n-1} \\ s^{n-2} \\ \vdots \\ 1 \end{pmatrix} \tag{5.53}$$

and

$$B^T(-sI - A)^{-T}Q(sI - A)^{-1}B = \frac{Q(s)}{A(s)A(-s)} = G(s) + G(-s) \tag{5.54}$$

Since $G(s)$ is SPR, the matrix A has no eigenvalues in the right half-plane or on the imaginary axis. Let P be the solution to Eq. (5.50). This matrix is positive definite because Q is positive definite and A has all its eigenvalues in the left half-plane. Furthermore, let $\tilde{C} = B^TP$. We now show that $\tilde{C} = C$. Since P is the solution to Eq. (5.50), it follows that

$$\tilde{C}(sI - A)^{-1}B + B^T(-sI - A)^{-T}\tilde{C}^T = B^T(-sI - A)^{-T}Q(sI - A)^{-1}B$$

But according to Eq. (5.54) the right-hand side is equal to $G(s) + G(-s)$. Since a partial fraction expansion is unique, it follows from Eq. (5.52) that

$$G(s) = C(sI - A)^{-1}B = \tilde{C}(sI - A)^{-1}B$$

which implies that $\tilde{C} = C$, and the theorem is proven. □

Test for Positive Realness

It is useful to have an algorithm to test whether a function is positive real. Theorem 5.7 can be used for this purpose. Condition (i) is easily tested by an ordinary Routh-Hurwitz test. Condition (ii) is a straightforward calculation. To test condition (iii), we proceed as follows:

$$G(s) = \frac{B(s)}{A(s)}$$

then

$$\operatorname{Re} G(i\omega) = \operatorname{Re}\frac{B(i\omega)}{A(i\omega)} = \operatorname{Re}\frac{B(i\omega)A(-i\omega)}{A(i\omega)A(-i\omega)}$$

Since the denominator is nonnegative and $G(i\omega)$ is symmetric with respect to the real axis, it suffices to investigate whether the function

$$f(\omega) = \operatorname{Re}\left(B(i\omega)A(-i\omega)\right)$$

is nonnegative for $\omega \ge 0$. Notice that f is an even function of ω. It is thus sufficient to investigate whether $f(\omega)$ has any real zeros. This can be verified directly by solving the equation $f(\omega) = 0$. There is also an indirect procedure. To describe this, introduce the polynomial

$$g(x) = f\left(\sqrt{x}\right)$$

The problem is thus to find whether the polynomial $g(x)$ has any zeros on the interval $(0, \infty)$. This classical problem can be solved as follows:

1. Let $g_1(x) = g(x)$, $g_2(x) = g'(x)$. Form a sequence of functions $\{g_1(x), g_2(x), \ldots, g_n(x)\}$ by letting $-g_{k+2}(x)$ be the remainder when dividing $g_k(x)$ by $g_{k+1}(x)$. Proceed until g_n is a constant.
2. Let $V(x)$ be the number of sign changes in the sequence $\{g_1(x), g_2(x), \ldots, g_n(x)\}$.
3. The number of real zeros of the function $g(x)$ in the interval $[a, b]$ is then $V(a) - V(b)$.

The function sequence $\{g_1(x), g_2(x), \ldots, g_n(x)\}$ is called a *Sturm sequence*. The procedure is illustrated by an example.

EXAMPLE 5.13 Second-order system

Consider the transfer function

$$G(s) = \frac{s^2 + 6s + 8}{s^2 + 4s + 3}$$

First notice that G has no poles in the right half-plane. Furthermore,

$$f(\omega) = \operatorname{Re}\left((-\omega^2 + 6i\omega + 8)(-\omega^2 - 4i\omega + 3)\right) = \omega^4 + 13\omega^2 + 24$$

Hence

$$g(x) = x^2 + 13x + 24$$

We get

$$g_1(x) = x^2 + 13x + 24$$
$$g_2(x) = 2x + 13$$
$$g_3(x) = \frac{73}{4}$$

Since $V(0) = 0$, $V(\infty) = 0$, $g(x)$ has no zeros on the positive real axis. The transfer function $G(s)$ is then SPR. □

An Alternative Test

An alternative test for SPR for a system with a proper transfer function can be obtained from the proof of the Kalman-Yakubovich lemma. Write the matrix A

in controllable canonical form. Solve the equations

$$A^T P + PA = \begin{pmatrix} q_1 & 0 & \cdots & 0 \\ 0 & q_2 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & q_n \end{pmatrix}$$

$$B^T P = C$$

where P is a symmetric matrix. This gives $n + n(n+1)/2$ equations for the unknown elements of P and Q. The transfer function is SPR if $q_i > 0$ for $i = 1, \ldots, n$.

The Passivity Theorem

Having established a notion of passivity, we can now state a key result.

THEOREM 5.8 The passivity theorem

Consider a system obtained by connecting two systems H_1 and H_2 in a feedback loop as in Fig. 5.16. Let H_1 be strictly output passive and H_2 be passive. The closed-loop system is then BIBO stable.

Proof: Since H_1 is strictly output passive, we have

$$\langle y \mid e \rangle > \delta \|y\|^2$$

Since $e = u - H_2 y$, we have

$$\delta \|y\|^2 \le \langle y \mid e \rangle = \langle y \mid u - H_2 y \rangle = \langle y \mid u \rangle - \langle y \mid H_2 y \rangle \qquad (5.55)$$

Since H_2 is passive, we have

$$\langle y \mid H_2 y \rangle \ge 0$$

and it then follows from Eq. (5.55) that

$$\delta \|y\|^2 \le \langle y \mid u \rangle \le \|y\| \, \|u\|$$

where the last inequality follows from Schwartz inequality. We now get

$$\|y\| < \frac{1}{\delta} \|u\|$$

which proves the result. □

Remark. The passivity theorem may also be regarded as an extension of Nyquist's stability theorem. Instability is avoided by having a loop transfer function with a phase lag less than 180°. □

Relations between Passivity and Small Gain Theorems

The small gain theorem (Theorem 5.6) and the passivity theorem (Theorem 5.8) are closely related. To investigate this connection further, we consider signal spaces that are inner product spaces and we show that the small gain theorem can be derived from the passivity theorem. We start with Fig. 5.16 and make a sequence of transformations of the feedback loop that are shown in Fig. 5.17.

Consider the closed-loop system in Fig. 5.17(a). Assume that the system H_1 is strictly output passive and that H_2 is passive. In Fig. 5.17(b) we have introduced two loops that cancel each other. The input-output relations of the encircled loops are $(I + H_1)^{-1} H_1$ and $I - H_2$, respectively. These two systems are shown in Fig. 5.17(c), where we have also added two loops and two gains (1/2 and 2) that cancel each other. The transfer functions of the encircled loops

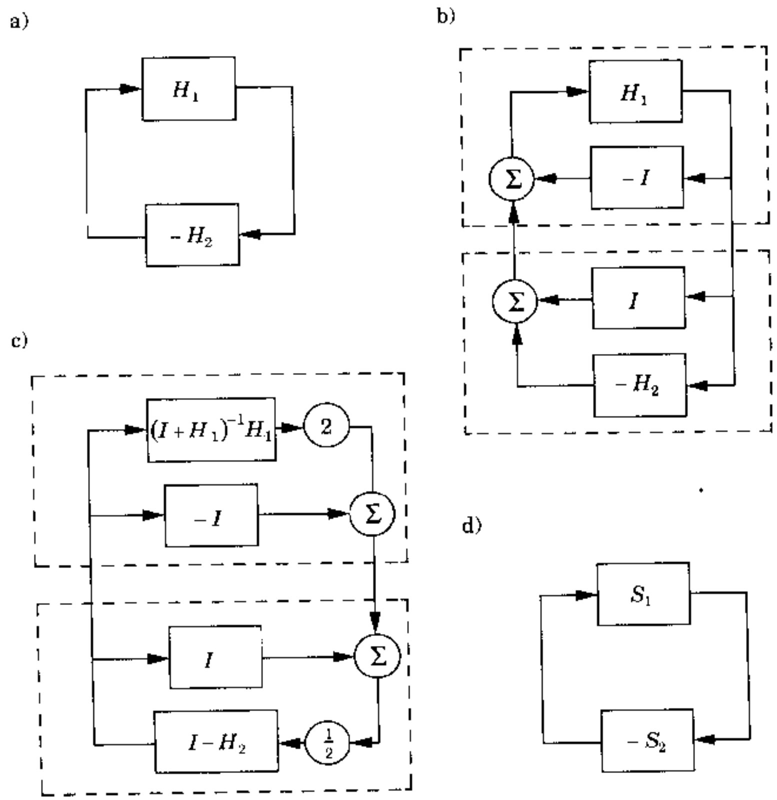

Figure 5.17 Four equivalent systems.

are

$$S_1 = 2(H_1 + I)^{-1}H_1 - (I + H_1)^{-1}(I + H_1) = (H_1 + I)^{-1}(H_1 - I)$$

and

$$S_2 = -\left(I - \frac{1}{2}(I - H_2)\right)^{-1}\frac{1}{2}(I - H_2) = (H_2 + I)^{-1}(H_2 - I)$$

The system obtained after the transformations is shown in Fig. 5.17(d).

The systems in Fig. 5.17(a) and Fig. 5.17(d) are equivalent. We use their equivalence to prove the result. First we observe that if the system $(H + I)^{-1}$ exists, it commutes with H. To prove this, use the identity

$$H + H^2 = H(H + I) = (H + I)H$$

and multiply from the left and the right by $(I + H)^{-1}$; then

$$(H + I)^{-1}H = H(H + I)^{-1}$$

Subtracting $(H + I)^{-1}$ from both sides gives

$$(H + I)^{-1}(H - I) = (H - I)(H + I)^{-1}$$

The systems S and H are related through

$$S = (H - I)(H + I)^{-1} = (H + I)^{-1}(H - I)$$

The input-output relation for the system S is

$$y = Su = (H - I)(H + I)^{-1}u$$

Introduce

$$x = (H + I)^{-1}u$$

We find that

$$y = (H - I)x$$
$$u = (H + I)x$$

Hence

$$\|y\|^2 = \langle y|y\rangle = \langle Hx - x|Hx - x\rangle = \langle Hx|Hx\rangle + \langle x|x\rangle - 2\langle Hx|x\rangle$$

Similarly, we find that

$$\|u\|^2 = \langle u|u\rangle = \langle Hx + x|Hx + x\rangle = \langle Hx|Hx\rangle + \langle x|x\rangle + 2\langle Hx|x\rangle$$

Hence

$$\|y\|^2 = \|u\|^2 - 4\langle Hx|x\rangle \tag{5.56}$$

If H is passive, we have $\langle Hx|x\rangle \geq 0$; hence $\|y\| \leq \|u\|$, which implies that $\gamma(S) \leq 1$. Similarly, we find that $\gamma(S) < 1$ if H is strictly output passive.

It follows from Eq. (5.56) that

$$\langle Hx|x\rangle = \frac{\|u\|^2 - \|y\|^2}{4} = \frac{\|u\|^2 - \|Su\|^2}{4} \geq (1 - \gamma(S))\frac{\|u\|^2}{4}$$

This implies that H is passive if $\gamma(S) \leq 1$ and strictly output passive if $\gamma(S) < 1$.

Notice that the argument would be the same if S and H were complex numbers. The result is an example of the equivalence between complex numbers and operators on inner product spaces.

5.7 APPLICATIONS TO ADAPTIVE CONTROL

The results from input-output stability theory are now used to construct adjustment rules for adaptive systems. So that we can focus on the principles and avoid unnecessary details, only the problem of adjusting a feedforward gain is considered in this section.

Consider a system with transfer function $kG(s)$ where $G(s)$ is known and k is an unknown constant. We will determine an adaptive feedforward compensation so that the transfer function from command signal to output is $k_0G(s)$. This problem was previously considered in Examples 5.1 and 5.3. A parameter adjustment law was also derived for the problem in Section 5.5 using Lyapunov theory. This control law can be represented by the block diagram in Fig. 5.14(b). According to Theorem 5.5 the adaptive system will be stable if the transfer function $G(s)$ is SPR. This condition indicates that the result is related to passivity theory. To establish this, we redraw the block diagram as in Fig. 5.18, which gives a configuration in which the passivity theorem can be applied. To use the passivity theorem, we must investigate the properties of the dashed block in Fig. 5.18. We have the following lemma.

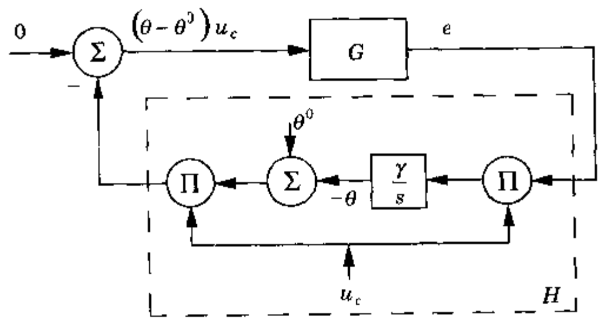

Figure 5.18 Representation of the system with adjustable feedforward gain when using the control law of Eq. 5.40. Compare with Fig. 5.14(b).

LEMMA 5.3 Property of positive real systems

Let r be a bounded square integrable function, and let $G(s)$ be a transfer function that is positive real. The system whose input-output relation is given by

$$y = r\,(G(p)ru)$$

is then passive.

Proof: It follows that

$$\langle y \mid u\rangle = \int_0^\infty y(\tau)u(\tau)\,d\tau = \int_0^\infty (u(\tau)r(\tau))\,(G(p)ru)(\tau)\,d\tau$$

$$= \int_0^\infty w(\tau)\,(G(p)w)\,(\tau)\,d\tau = \langle w \mid Gw\rangle$$

where $w = ru$. Since $G(s)$ is positive real, it follows from Example 5.12 that $\langle w|Gw\rangle \geq 0$, which proves the result. □

By invoking the passivity theorem (Theorem 5.8) we can now obtain an alternative proof of Theorem 5.5. Figure 5.18 shows that the model-reference system can be viewed as a feedback connection of two systems. One system is linear with the transfer function G. It has the signal $(\theta-\theta^0)u_c$ as the input and the model error as the output. The other system has the model error e as the input and the quantity $-(\theta-\theta^0)u_c$ as the output. Since an integrator is positive real, it follows from Lemma 5.3 that the system H is passive. If the transfer function G is proper and strictly positive real, it follows from Example 5.12 that $G(s)$ is output strictly proper. The passivity theorem (Theorem 5.8) then implies that the closed-loop system is BIBO stable. In Fig. 5.18 there are no external inputs, as in Fig. 5.16. The system in Fig. 5.18 may have initial conditions, however, because the process and the model may have different initial conditions. The integrator may also have an initial condition that can be thought of as being generated by an external input signal. Such an input signal can always be chosen to be zero for $t \geq 0$. We thus have a situation covered by Theorem 5.6, where the input signal u is bounded in L_2. The error $e(t)$ goes to zero as t goes to infinity. Notice that the MRAS is stable for all values of $\gamma > 0$ when the SPR condition is satisfied. This implies that the adaptation can be made arbitrarily fast.

Design of Stable Adjustment Mechanisms

The passivity theorem gives a convenient way to construct stable adjustment laws. We simply try to introduce some compensating network so that the transfer function relating the error to $(\theta - \theta^0)u_c$ is strictly positive real, as

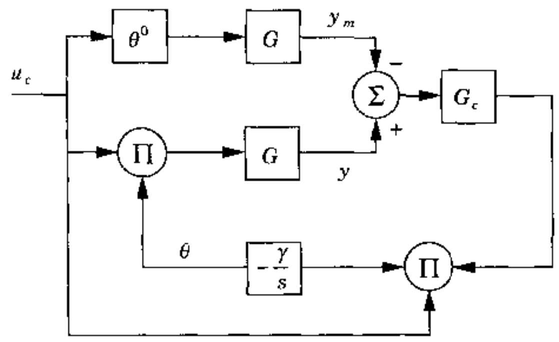

Figure 5.19 A stable parameter adjustment law is obtained if GG_c is SPR.

is illustrated in Fig. 5.19. For systems with output feedback, the problem is to find a compensator G_c such that the transfer function GG_c is strictly positive real. This can be done by using the Kalman-Yakubovich lemma (Lemma 5.2). With pure feedforward control it is natural to assume that G is stable. It can then be written as

$$G(s) = \frac{B(s)}{A(s)}$$

where $A(s)$ has all its zeros in the left half-plane. For a stable polynomial $A(s)$ a polynomial $C(s)$ such that $C(s)/A(s)$ is SPR can always be found. To do this, we introduce the following canonical realization of $1/A(s)$:

$$\frac{dx}{dt} = \begin{pmatrix} -a_1 & -a_2 & \dots & -a_{n-1} & -a_n \\ 1 & 0 & & 0 & 0 \\ \vdots & & & & \\ 0 & 0 & & 1 & 0 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} u$$

Choose a symmetric positive definite matrix Q and solve the equation

$$A^T P + PA = -Q$$

The coefficients of a C polynomial such that $C(s)/A(s)$ is SPR are then the first row of the P matrix.

The polynomial $C(s)$ will have a degree that is at most equal to $\deg A - 1$. For systems with stable zeros and pole excess 1 it is thus possible to find a stable adjustment rule by choosing $G_c(s) = C(s)/B(s)$. However, for systems with higher pole excess than 1 the compensator required to make GG_c strictly positive real will contain derivatives. We will show how to deal with the case in which the pole excess is higher by introducing the augmented error.

PI Adjustments

All adjustment laws discussed so far have been integral controllers. That is, the parameter has always been obtained as the output of an integrator. There are, of course, many other possibilities for choosing the adaptation mechanism H in Fig. 5.18. For instance, it can be expected that quicker adaptation can be achieved by using a proportional and integral adjustment law. This means that the control law of Eq. (5.40) is replaced by

$$\theta(t) = -\gamma_1 u_c(t)e(t) - \gamma_2 \int^t u_c(\tau)e(\tau)\, d\tau \tag{5.57}$$

Since a system with the transfer function

$$H(s) = \gamma_1 + \gamma_2/s$$

is output strictly passive for positive γ_1 and γ_2, it follows from Theorem 5.8 (the passivity theorem) that Eq. (5.57) gives a stable adjustment law if GG_c is positive real.

The Augmented Error

Some progress has now been made to construct stable parameter adjustment rules for the problem of adjusting a feedforward gain. Passivity theory gave good insight and led to the idea of filtering the model error so that GG_c is SPR. However, we have not solved the problem in which G has a pole excess larger than 1. To do this, we first factor the transfer function G as

$$G = G_1G_2$$

where the transfer function G_1 is SPR. The error $e = y - y_m$ can then be written as

$$\begin{aligned} e &= G(\theta - \theta^0)u_c = (G_1G_2)(\theta - \theta^0)u_c \\ &= G_1\left(G_2(\theta - \theta^0)u_c + (\theta - \theta^0)G_2u_c - (\theta - \theta^0)G_2u_c\right) \\ &= G_1\left((\theta - \theta^0)G_2u_c\right) - G_1\left((\theta - \theta^0)G_2u_c - G_2(\theta - \theta^0)u_c\right) \end{aligned}$$

Introduce the *augmented error* ε defined by

$$\varepsilon = e + \eta$$

where η is the *error augmentation* defined by

$$\begin{aligned} \eta &= G_1(\theta - \theta^0)G_2u_c - G(\theta - \theta^0)u_c \\ &= G_1(\theta G_2u_c) - G\theta u_c \end{aligned}$$

The second equality follows because $G\theta^0 u = \theta^0 Gu$ when θ^0 is constant. The augmented error is thus obtained by adding a correction term η to the error. The correction term vanishes when the parameter θ is constant. It follows that

$$\varepsilon = G_1\left((\theta - \theta^0)G_2u_c\right) = G_1(\theta - \theta^0)\bar{u}_c \tag{5.58}$$

where $\bar{u}_c$ is the reference signal filtered through G_2. Equation (5.58) is an error model similar to the ones used previously, and we have the following theorem.

THEOREM 5.9 Stability using augmented error

Consider a model-reference system for adaptation of a feedforward gain for a system with the transfer function G. Let G_1G_2 be a factorization of G such that G_1 is SPR. The parameter adjustment law

$$\frac{d\theta}{dt} = -\gamma\varepsilon(G_2u_c) \tag{5.59}$$

where

$$\varepsilon = e + G_1(\theta G_2u_c) - G(\theta u_c) \tag{5.60}$$

gives a closed-loop system in which the error goes to zero as t goes to infinity.

Proof: Since G_1 is SPR, the discussion of the error model shows that $\varepsilon \in L_2$.

Remark 1. The trivial factorization with $G_1 = 1$ is one possibility.

Remark 2. If the input signal is persistently exciting, it can be shown that the parameters also converge.

Remark 3. Notice that G_2 must be minimum phase to establish that θ converges to θ^0. The reason is that we have to go "backwards" through G_2 to show that $\theta - \theta^0$ goes to zero if the output e goes to zero. That is, the inverse of G_2 must be stable. This is a condition that will be seen again in the general case in Section 5.8. □

A block diagram of the system with augmented error is shown in Fig. 5.20. To implement the augmented error, it is necessary to introduce realizations of the transfer functions G_1 and G_2. The augmented error was introduced by Monopoli. It was a key idea for adaptive control systems having pole excess larger than 1. Application of the idea to general linear systems is discussed in Section 5.8. In Section 5.9 we show that the augmented error appears naturally in the self-tuning regulator.

Summary

The problem of adjusting the gain in a known system has been used to introduce some ideas in the design of stable model-reference adaptive systems. It was first shown that adjustment rules could be obtained for systems in which the plant is strictly positive real. The parameter adjustment rules were similar to those obtained by the gradient method.

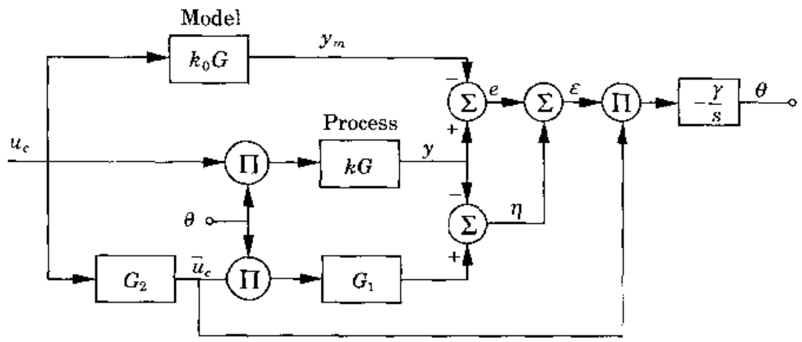

Figure 5.20 Block diagram of a model-reference adaptive system based on the augmented error.

The class of systems could then be extended by using adjustment rules in which the error is filtered. In this way the problem can be solved for stable minimum-phase systems that have pole excess less than 1. The idea of augmented error was introduced to solve the problem of higher pole excess.

5.8 OUTPUT FEEDBACK

We now derive an MRAS for adjusting the parameters of a controller based on output feedback in a fairly general case. A process with one input and one output is considered. It is assumed that the dynamics are linear and that the control problem is formulated as model-following. The key assumption is that the controller can be parameterized in such a way that the error is linear in the controller parameters. The derivation of the MRAS is described as follows:

Step 1: Find a controller structure that admits perfect output tracking.

Step 2: Derive an error model of the form

$$\varepsilon = G_1(p)\left\{\varphi^T(t)(\theta^0 - \theta)\right\} \tag{5.61}$$

where G_1 is a strictly positive real transfer function, θ^0 is the process parameters, and θ is the controller parameters. The right-hand side should be expressed in computable quantities.

Step 3: Use the parameter adjustment law

$$\frac{d\theta}{dt} = \gamma\varphi\varepsilon \tag{5.62}$$

or the normalized law

$$\frac{d\theta}{dt} = \gamma\frac{\varphi\varepsilon}{\alpha + \varphi^T\varphi} \tag{5.63}$$

Notice that the error ε in Eq. (5.61) is linear in the parameters, a condition that imposes restrictions on the models and controllers that can be dealt with. A model of the form (5.61) is typically obtained by algebraic manipulations, filtering, and error augmentation.

We now show one way to apply the design procedure.

Finding a Controller Structure

The first step in the design procedure is to find a suitable controller structure. The tools for doing this were developed in Section 3.2. Let the process be described by the continuous-time model

$$Ay(t) = b_0Bu(t) \tag{5.64}$$

where it is assumed that the polynomials A and B do not have common factors and the polynomial B is monic and assumed to have all its zeros in the left half-plane. Furthermore, the polynomial is normalized so that B is monic. The variable b_0 is called the *instantaneous gain* or the *high-frequency gain*. A general linear controller can be written as

$$Ru(t) = -Sy(t) + Tu_c(t) \tag{5.65}$$

where u_c is the command signal. Since the polynomial B is stable, the corresponding poles can be canceled by the controller. This corresponds to $R = R_1B$. The closed-loop system obtained when the controller is applied to the process (5.64) is described by

$$(AR_1 + b_0S)y = b_0Tu_c \tag{5.66}$$

If polynomial T is chosen to be $T = t_0A_0$, where A_0 is a stable monic polynomial and R_1 and S satisfy

$$AR_1 + b_0S = A_0A_m \tag{5.67}$$

it is possible to achieve perfect model-following with the model

$$A_my_m(t) = b_0t_0u_c(t) \tag{5.68}$$

The Error Model

Having obtained a suitable controller structure, we now proceed to derive an error model. It follows from Eq. (5.67) that

$$A_oA_my = AR_1y + b_0Sy = R_1b_0Bu + b_0Sy \tag{5.69}$$

where the first equality follows from Eq. (5.67) and the second from Eq. (5.64). Introduce the error

$$e = y - y_m$$

It follows from Eqs. (5.69) and (5.68) that

$$A_oA_me = A_oA_m(y - y_m) = b_0(Ru + Sy - Tu_c)$$

or

$$e = \frac{b_0}{A_oA_m}(Ru + Sy - Tu_c)$$

This expression is not yet a suitable error model, because the transfer function $b_0/(A_oA_m)$ is not SPR. Therefore introduce the filtered error

$$e_f = \frac{Q}{P}e = \frac{Q}{P}(y - y_m)$$

where Q is a polynomial whose degree is not greater than $\deg A_oA_m$ such that

$$\frac{b_0Q}{A_oA_m} \tag{5.70}$$

is SPR. The filtered error can be written as

$$e_f = \frac{b_0Q}{A_oA_m}\left(\frac{R}{P}u + \frac{S}{P}y - \frac{T}{P}u_c\right)$$

Let $P = P_1P_2$, where P_2 is a stable monic polynomial of the same degree as R. Rewrite R/P as

$$\frac{R}{P} = \frac{R - P_2 + P_2}{P_1P_2} = \frac{1}{P_1} + \frac{R - P_2}{P}$$

The filtered error then becomes

$$e_f = \frac{b_0Q}{A_oA_m}\left(\frac{1}{P_1}u + \frac{R - P_2}{P}u + \frac{S}{P}y - \frac{T}{P}u_c\right)$$

Let k, l, and m be the degrees of the polynomials R, S, and T, respectively. Introduce a vector of true controller parameters

$$\theta^0 = (r'_1 \ldots r'_k\, s_0 \ldots s_l\, t_0 \ldots t_m)^T \tag{5.71}$$

where r'_i are the coefficients of the polynomial $R - P_2$. Also introduce a vector of filtered input, output, and command signals

$$\varphi^T = \left(\frac{p^{k-1}}{P(p)}u \ldots \frac{1}{P(p)}u \quad \frac{p^l}{P(p)}y \ldots \frac{1}{P(p)}y \quad -\frac{p^m}{P(p)}u_c \ldots -\frac{1}{P(p)}u_c\right) \tag{5.72}$$

The filtered error can then be written as

$$e_f = \frac{b_0Q}{A_oA_m}\left(\frac{1}{P_1}u + \varphi^T\theta^0\right) \tag{5.73}$$

To obtain an error model, we must introduce a parameterization of the controller. In the nominal case in which the parameters are known, the control law can be expressed as

$$u = -P_1(\varphi^T\theta^0) = -P_1\left((\theta^0)^T\varphi\right) = -(\theta^0)^T(P_1\varphi) \tag{5.74}$$

where P_1 is a polynomial in the differential operator. Let θ denote the adjustable controller parameters. The feedback law

$$u = -P_1(\varphi^T\theta)$$

would give the desired error model. However, this control law is not realizable if P_1 has a degree greater than 1 because the term $P_1(\varphi^T\theta)$ contains derivatives of the parameters. However, the control law

$$u = -\theta^T(P_1\varphi) \tag{5.75}$$

is realizable because of Eq. (5.69). If we use this control law, it follows from Eq. (5.70) that the filtered error can be written as

$$\begin{aligned} e_f &= \frac{b_0Q}{A_oA_m}\left(\varphi^T\theta^0 - \frac{1}{P_1}\theta^T(P_1\varphi)\right) \\ &= \frac{b_0Q}{A_oA_m}\left(\varphi^T\theta^0 - \varphi^T\theta - \frac{1}{P_1}\theta^T(P_1\varphi) + \varphi^T\theta\right) \end{aligned}$$

Introduce the signals η and ε, defined by

$$\begin{aligned} \eta &= \frac{1}{P_1}\theta^T(P_1\varphi) - \varphi^T\theta = -\left(\frac{1}{P_1}u + \varphi^T\theta\right) \\ \varepsilon &= e_f + \frac{b_0Q}{A_oA_m}\eta = \frac{b_0Q}{A_oA_m}\varphi^T(\theta^0 - \theta) \end{aligned} \tag{5.76}$$

The signal ε is called the *augmented error*, and η is called the *error augmentation*. The augmented error is computed as follows:

$$\varepsilon = \frac{Q}{P}(y - y_m) + \frac{b_0Q}{A_oA_m}\eta \tag{5.77}$$

With the chosen degrees of P and Q it is straightforward to verify that the computation does not require taking derivatives of the signals y, u, u_c, and y_m. The error model of Eq. (5.76) is also linear in the parameters, and the transfer function $b_0Q/(A_oA_m)$ is SPR. The error model thus satisfies the requirements of Step 2, and the parameters can then be updated by Eq. (5.62) or Eq. (5.63). So far, the derivation has been done along the lines developed in Sections 5.3 and 5.4. However, to show the stability of the closed-loop system, it is not sufficient that the system (5.70) is SPR. It is also necessary that the signals in φ are bounded. This condition can be difficult to show. Furthermore, Eqs. (5.76) are valid only if the control signal is generated from Eq. (5.75). This implies, for instance, that the control signal cannot be saturated. Notice that it is necessary to know the parameter b_0 to compute the augmented error ε.

The derived algorithm thus requires that the high-frequency gain b_0 be known. If the parameter is not known, it can be estimated as follows. The error model of Eq. (5.73) can be written as

$$e_f = b_0\left(\varphi_f^T\theta^0 + u_f\right) \tag{5.78}$$

where

$$\varphi_f = \frac{Q}{A_oA_m}\varphi$$

$$u_f = \frac{Q}{A_oA_mP_1}u$$

A simple gradient estimator for b_0 and θ^0 is then given by

$$\begin{aligned}\frac{d\theta}{dt} &= \gamma'\hat{b}_0\varphi_f\varepsilon_p = \gamma\varphi_f\varepsilon_p\\ \frac{d\hat{b}_0}{dt} &= \gamma\left(\varphi_f^T\theta + u_f\right)\varepsilon_p\end{aligned} \tag{5.79}$$

where ε_p is the prediction error

$$\varepsilon_p = e_f - \hat{e}_f = e_f - \hat{b}_0\left(\varphi_f^T\theta + u_f\right) \tag{5.80}$$

Notice that $\hat{b}_0$ can be absorbed in the adaptation gain if its sign is known.

Realization

The equations needed to implement the general MRAS can now be summarized:

$$\begin{aligned}y_m &= \frac{B_m}{A_m}u_c\\ e_f &= \frac{Q}{P}e = \frac{Q}{P}(y - y_m)\\ \eta &= -\left(\frac{1}{P_1}u + \varphi^T\theta\right)\\ \varepsilon &= e_f + \frac{b_0Q}{A_oA_m}\eta\\ \frac{d\theta}{dt} &= \gamma\varphi\varepsilon\\ u &= -\theta^T(P_1\varphi)\end{aligned}$$

A block diagram of the model-reference adaptive system is shown in Fig. 5.21. The block labeled "Filter" in Fig. 5.21 is a linear system that generates $P_1\varphi$ and φ from the signals u_c, u, and y. The vector φ is composed of three parts

having the same structure. It therefore suffices to discuss one part. Consider, for example, how to generate φ_u and $P_1\varphi_u$ where

$$P_1\varphi_u = \left(\frac{p^{k-1}}{P_2}u \;\ldots\; \frac{1}{P_2}u\right)^T = (x_1 \ldots x_k)^T = x^T$$

and

$$\varphi_u = \left(\frac{p^{k-1}}{P}u \ldots \frac{1}{P}u\right)^T$$

where $P = P_1P_2$ and $k = \deg R = \deg P_2$.

Let the polynomials P_1 and P_2 be

$$\begin{aligned}P_1 &= p^n + \alpha_1 p^{n-1} + \cdots + \alpha_n\\ P_2 &= p^k + \beta_1 p^{k-1} + \cdots + \beta_k\end{aligned}$$

We also assume that $\deg P_1 > \deg P_2$. The vectors x and φ_u can then be realized as follows:

$$\frac{dx}{dt} = \begin{pmatrix} -\beta_1 & -\beta_2 & \ldots & -\beta_{k-1} & -\beta_k\\ 1 & 0 & & 0 & 0\\ \vdots & \ddots & & & \\ 0 & 0 & & 1 & 0\end{pmatrix} x + \begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix} u$$

$$\frac{dz}{dt} = \begin{pmatrix} -\alpha_1 & -\alpha_2 & \ldots & -\alpha_{n-1} & -\alpha_n\\ 1 & 0 & & 0 & 0\\ \vdots & \ddots & & & \\ 0 & 0 & & 1 & 0\end{pmatrix} z + \begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix} x_k$$

where $x_k = 1/P_2 \cdot u$ is the last element of the x vector. The elements of φ_u are the k last elements of the state vector z. Furthermore, $1/P_1 \cdot u$ can also be obtained from the generation of φ_u and $P_1\varphi_u$. To generate the full vectors φ and $P_1\varphi$, we thus need three realizations of the transfer functions P_1 and P_2. The block labeled "Filter" in Fig. 5.21 represents these systems.

Design Parameters

Several parameters must be chosen in the design procedure:

- The model transfer function B_m/A_m,
- The observer polynomial A_o,
- The degrees of polynomials R, S, and T, and
- The polynomials P_1, P_2, and Q.

Many different model-reference adaptive systems can be obtained by different choices of the design parameters. A popular choice of the polynomials is $P_1 = A_m$, $P_2 = A_o$, and $Q = A_oA_m$.

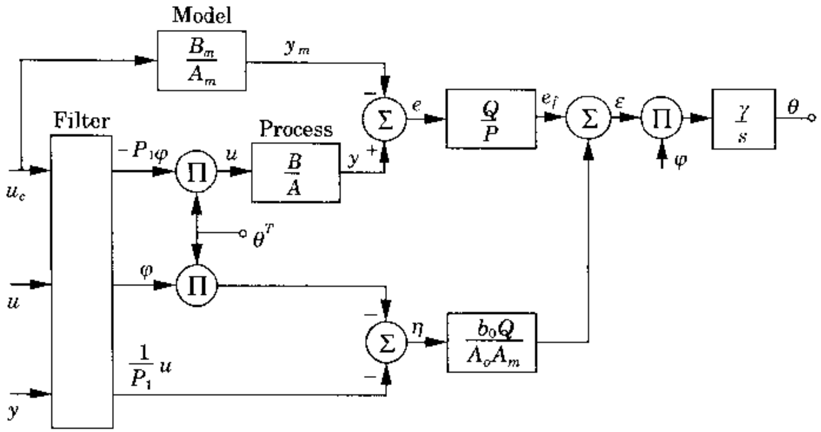

Figure 5.21 Block diagram of a model-reference adaptive system for a SISO system.

A Priori Knowledge

To apply the MRAS procedure, the plant must be minimum-phase and the following prior information must also be known:

- The sign of the instantaneous gain b_0,
- The pole excess of the plant, and
- The order of the plant or the controller complexity.

EXAMPLE 5.14 Second-order MRAS

The performance of the general MRAS is illustrated by a second-order example, given the system

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

and the model

$$G_m(s) = \frac{B_m}{A_m} = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

The polynomials A_o, R, S, and T can be chosen to be

$$A_o(s) = s + a_o$$
$$R(s) = s + r_1$$
$$S(s) = s_0 s + s_1$$
$$T(s) = t_0 s + t_1$$

The Diophantine equation (Eq. 5.67) gives the solution

$$r_1 = 2\zeta\omega + a_o - a$$
$$s_0 = (2\zeta\omega a_o + \omega^2 - a r_1)/k$$
$$s_1 = a_o\omega^2/k$$
$$t_0 = \omega^2/k$$
$$t_1 = a_o\omega^2/k$$

For simplicity we choose

$$Q(s) = A_o(s)A_m(s)$$
$$P_1(s) = A_m(s)$$
$$P_2(s) = A_o(s)$$

Figure 5.22 shows a simulation of the system with $\gamma = 1$, $\zeta = 0.7$, $\omega = 1$, $a_o = 2$, $a = 1$, and $k = 2$. In the simulation it is assumed that $\hat{b}_0 = b_0$. The used values of the filters P_1, P_2, Q, and A_o give a fairly rapid convergence of y to y_m. The parameter estimates at the end of the simulation are still far from the optimal values, but the error is small (see Fig. 5.22c). The controller parameters are shown in Fig. 5.23. The control law at $t = 150$ gives a closed-loop system with a pole in -1.95 and two complex poles corresponding to

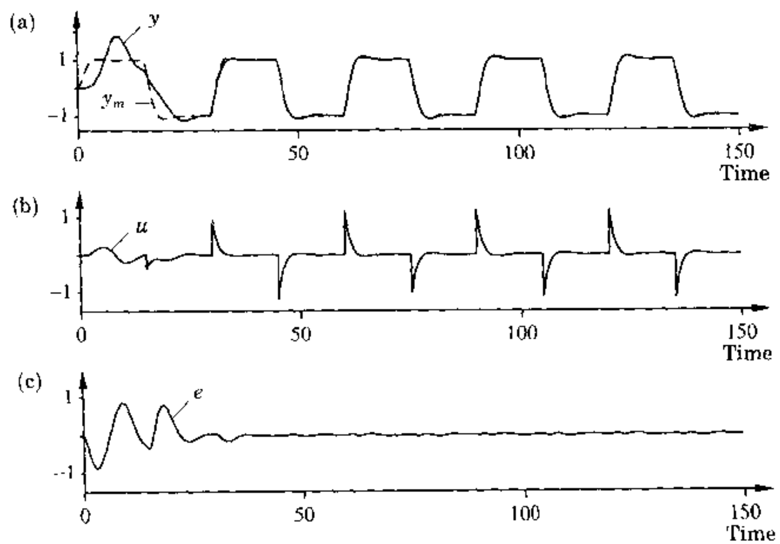

Figure 5.22 Simulation of the system in Example 5.14. (a) The process output (solid line) and the model output (dashed line). (b) The control signal. (c) The error $e = y - y_m$.

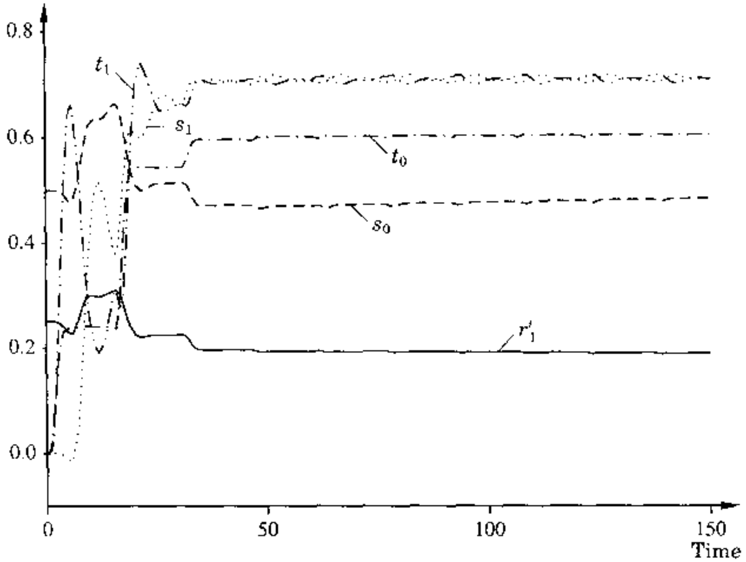

Figure 5.23 The controller parameters in the simulation of the system in Example 5.14.

$\omega = 0.84$ and $\zeta = 0.78$, which should be compared to the roots of $A_o A_m$, which are in -2, and complex poles corresponding to $\omega = 1$ and $\zeta = 0.7$. □

5.9 RELATIONS BETWEEN MRAS AND STR

For a long time, model-reference adaptive systems and self-tuning regulators were regarded as two quite different approaches to adaptive control. In this section we will show that the methods are closely related. The key observation is that the direct self-tuner in which process zeros are canceled (Algorithm 3.3) can be interpreted as a MRAS.

An MRAS for a general continuous-time linear system was derived in Section 5.8. In the derivation it was assumed that the process was minimum-phase and that all its zeros were canceled in the design. We showed that the adjustment law for updating the parameters can be written as

$$\frac{d\theta}{dt} = \gamma \varphi_f \varepsilon \tag{5.81}$$

where φ_f is a filtered regression vector and ε is the augmented error given by Eq. (5.77), that is,

$$\varepsilon = \frac{Q}{P}(y - y_m) + \frac{b_o Q}{A_o A_m}\eta \tag{5.82}$$

Now consider a discrete-time direct self-tuner. When all process zeros are canceled, polynomial B^- is a constant and we get the process model

$$y(t) = \varphi_f^T(t - d_0)\theta$$

In the direct algorithm the estimated parameters are equal to the controller parameters. The least-squares method can be used for the estimation by using the residual

$$\varepsilon(t) = y(t) - \hat{y}(t) = y(t) - \varphi_f^T(t - d_0)\hat{\theta}(t-1)$$

The parameter update can be written as

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi_f^T(t - d_0)\varepsilon(t) \tag{5.83}$$

The residual is given by

$$\varepsilon(t) = y(t) - \hat{y}(t) = y(t) - y_m(t) + y_m(t) - \hat{y}(t) = e(t) + \eta(t) \tag{5.84}$$

A comparison of Eqs. (5.81) and (5.83) show that Eq. (5.83) can be interpreted as a discrete-time version of Eq. (5.81). Notice that the gain γ in the MRAS is replaced by the matrix $P(t)$. This matrix changes the gradient direction φ_f and gives an appropriate step length. Also notice that it follows from Eq. (5.84) that the error augmentation is simply $y - \hat{y}$. The augmented error that required a significant ingenuity to derive in the MRAS context is thus obtained directly from the least-squares equations in the STR. More filtering is required in the MRAS because of the continuous time formulation. Notice that it follows from Eq. (5.83) that

$$\varphi_f^T(t - d_0) = -\operatorname{grad}_\theta \varepsilon(t)$$

The vector $\varphi_f^T(t - d_0)$ can be interpreted as the sensitivity derivative of the prediction error ε with respect to the parameter. The parameter update of Eq. (5.83) is thus a discrete-time version of the MIT rule. The main difference is that the model error $e(t) = y(t) - y_m(t)$ is replaced by the prediction error $\varepsilon(t)$.

Notice that in the identification-based schemes such as self-tuning controllers we normally attempt to obtain a form similar to

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

With the model-reference approach, it is also possible to admit a model of the form

$$y(t) = G(p)\left(\varphi_f^T\theta\right)$$

where $G(p)$ is SPR. In summary we thus find that the MRAS-type algorithms can be obtained in a straightforward way as a direct self-tuning regulator based on a minimum-degree pole placement design with cancellation of the whole B polynomial.

5.10 NONLINEAR SYSTEMS

The Lyapunov method can also be used to find adaptive control laws for nonlinear systems. This is a difficult problem because no general design methods are available. There is, however, much interest in adaptive control of nonlinear systems. For this reason we present some of the current ideas and illustrate them by a few examples.

Feedback Linearization

Before attempting to do adaptive control, we must first have a design method for the case in which the parameters are known. Feedback linearization is a design method that is similar in spirit to pole placement. It can be applied to certain classes of systems. We illustrate it through an example.

EXAMPLE 5.15 Feedback linearization

Consider the system

$$\frac{dx_1}{dt} = x_2 + f(x_1)$$
$$\frac{dx_2}{dt} = u$$

where f is a differentiable function. The first step is to introduce new coordinates

$$\xi_1 = x_1$$
$$\xi_2 = x_2 + f(x_1)$$

The equations then become

$$\frac{d\xi_1}{dt} = \xi_2$$
$$\frac{d\xi_2}{dt} = \xi_2 f'(\xi_1) + u$$

By introducing the control law

$$u = -a_2\xi_1 - a_1\xi_2 - \xi_2 f'(\xi_1) + v$$

we get a linear closed-loop system described by

$$\frac{d\xi}{dt} = \begin{pmatrix} 0 & 1 \\ -a_2 & -a_1 \end{pmatrix} \xi + \begin{pmatrix} 0 \\ 1 \end{pmatrix} v$$

This system is linear with the characteristic equation

$$s^2 + a_1 s + a_2 = 0$$

By transforming back to the original coordinates the control law can be written as

$$u = -a_2 x_1 - \Big(a_1 + f'(x_1)\Big)\Big(x_2 + f(x_1)\Big) + v \qquad \square$$

The closed-loop system obtained in the example will behave like a linear system. This is the reason why the method is called *feedback linearization*. The system in Example 5.15 is quite special. Applying the same procedure for a system described by

$$\frac{dx}{dt} = f(x) + ug(x)$$

we first pick

$$\xi_1 = h(x)$$

as a new state variable. The time derivative of ξ_1 is

$$\frac{d\xi_1}{dt} = h'(x)\Big(f(x) + ug(x)\Big)$$

If $h'(x)g(x) = 0$, we introduce the new state variable

$$\xi_2 = h'(x)f(x)$$

We proceed as long as the control variable u does not appear explicitly on the right-hand side. In this way we obtain the state variables $\xi_1 \ldots \xi_r$, which are combined to the vector $\xi \in R^r$, where $r \le n$. We also introduce the new state variable $\eta_1 \ldots \eta_{n-r}$, which are combined into the vector $\eta \in R^{n-r}$. This can be done in many different ways. We obtain the following system of equations:

$$\begin{aligned}
\frac{d\xi_1}{dt} &= \xi_2 \\
\frac{d\xi_2}{dt} &= \xi_3 \\
&\vdots \\
\frac{d\xi_r}{dt} &= \alpha(\xi,\eta) + u\beta(\xi,\eta) \\
\frac{d\eta}{dt} &= \gamma(\xi,\eta)
\end{aligned} \tag{5.85}$$

Notice that the state variables ξ represents a chain of r integrators, where the integer r is the nonlinear equivalence of pole excess. The variables η will not appear if $r = n$. This case corresponds to a system without zeros. This actually occurs in Example 5.15, where $r = n = 2$.

A design procedure, which is the nonlinear analog of pole placement, can be constructed if $\beta(\xi,\eta) \neq 0$. If this is the case, we can introduce the feedback law

$$u = \frac{1}{\beta(\xi,\eta)}\Big(-a_r\xi_1 - a_{r-1}\xi_2 - \ldots - a_1\xi_r - \alpha(\xi,\eta) + b_0 v\Big)$$

The closed-loop system then becomes

$$\frac{d\xi}{dt} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & \\ \vdots & & & & \\ -a_r & -a_{r-1} & -a_{r-2} & \dots & -a_1 \end{pmatrix} \xi + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ b_0 \end{pmatrix} v \tag{5.86}$$

$$\frac{d\eta}{dt} = \gamma(\xi, \eta)$$

The relation between v and ξ_1 is given by a *linear* dynamical system with the transfer function

$$\frac{\Xi_1(s)}{V(s)} = G(s) = \frac{b_0}{s^r + a_1 s^{r-1} + \dots a_r}$$

This differential equation has a triangular structure. The part corresponding to the state vector ξ is a linear system that is decoupled from the variable η. If $\xi = 0$, the behavior of the system (5.86) is governed by

$$\frac{d\eta}{dt} = \gamma(0, \eta) \tag{5.87}$$

This equation represents the *zero dynamics*. It is necessary for this system to be stable if the proposed control design is going to work. For linear systems the zero dynamics are the dynamics associated with the zeros of the transfer function. Feedback linearization is the nonlinear analog of pole placement with cancellation of all process zeros.

Adaptive Feedback Linearization

We now show how feedback linearization can be extended to deal with the situation in which the process model has unknown parameters. The approach will be similar to the idea used to derive model-reference adaptive controllers. Let us start with an example that is an adaptive version of Example 5.15.

EXAMPLE 5.16 Adaptive feedback linearization

Consider the system

$$\frac{dx_1}{dt} = x_2 + \theta f(x_1)$$

$$\frac{dx_2}{dt} = u$$

where θ is an unknown parameter and f is a known differentiable function. Applying the certainty equivalence principle gives the following control law:

$$u = -a_2 x_1 - \left(a_1 + \hat{\theta} f'(x_1)\right)\left(x_2 + \hat{\theta} f(x_1)\right) + v \tag{5.88}$$

Introducing this into the system equations gives an error equation that is nonlinear in the parameter error. This makes it very difficult to find a parameter adjustment law that gives a stable system. Therefore it is necessary to use another approach.

Proceeding as in Example 5.15 and introducing the new coordinates

$$\xi_1 = x_1$$

$$\xi_2 = x_2 + \hat{\theta} f(x_1)$$

where $\hat{\theta}$ is an estimate of θ, we have

$$\frac{d\xi_1}{dt} = \frac{dx_1}{dt} = x_2 + \theta f(x_1) = \xi_2 + (\theta - \hat{\theta}) f(\xi_1)$$

$$\frac{d\xi_2}{dt} = \frac{d\hat{\theta}}{dt} f(x_1) + \hat{\theta}\left(x_2 + \theta f(x_1)\right) f'(x_1) + u$$

Choosing the control law to be

$$u = -a_2\xi_1 - a_1\xi_2 - \hat{\theta}\left(x_2 + \hat{\theta} f(x_1)\right) f'(x_1) - f(x_1)\frac{d\hat{\theta}}{dt} + v \tag{5.89}$$

we get

$$\frac{d\xi}{dt} = \begin{pmatrix} 0 & 1 \\ -a_2 & -a_1 \end{pmatrix} \xi + \begin{pmatrix} f(\xi_1) \\ \hat{\theta} f(\xi_1) f'(\xi_1) \end{pmatrix} \tilde{\theta} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} v$$

A comparison with the certainty equivalence control law given by Eq. (5.88) shows that the major difference is the presence of the term $d\hat{\theta}/dt$ in Eq. (5.89).

In analogy with the model-reference adaptive system, let us assume that it is desired to have a system in which the transfer function from command signal to output has the transfer function

$$G(s) = \frac{a_2}{s^2 + a_1 s + a_2}$$

Introduce the following realization of the transfer function:

$$\frac{dx_m}{dt} = \begin{pmatrix} 0 & 1 \\ -a_2 & -a_1 \end{pmatrix} x_m + \begin{pmatrix} 0 \\ a_2 \end{pmatrix} u_m$$

and let $e = \xi - x_m$ be the error vector. If we choose

$$v = a_2 u_m \tag{5.90}$$

we find that the error equation becomes

$$\frac{de}{dt} = \begin{pmatrix} 0 & 1 \\ -a_2 & -a_1 \end{pmatrix} e + \begin{pmatrix} f(\xi_1) \\ \hat{\theta} f(\xi_1) f'(\xi_1) \end{pmatrix} \tilde{\theta} = Ae + B\tilde{\theta}$$

where

$$A = \begin{pmatrix} 0 & 1 \\ -a_2 & -a_1 \end{pmatrix} \qquad B = \begin{pmatrix} f(\xi_1) \\ \hat{\theta} f(\xi_1) f'(\xi_1) \end{pmatrix}$$

The matrix A has all eigenvalues in the left half-plane if $a_1 > 0$ and $a_2 > 0$. It is then possible to find a matrix P such that

$$A^T P + PA = -I$$

Choosing the Lyapunov function

$$V = e^T Pe + \frac{1}{\gamma}\tilde{\theta}^2$$

we find that

$$\frac{dV}{dt} = e^T(A^T P + PA)e + 2\tilde{\theta}B^T Pe + \frac{2}{\gamma}\tilde{\theta}\frac{d\tilde{\theta}}{dt}$$

If the law for updating the parameters is chosen to be

$$\frac{d\hat{\theta}}{dt} = \gamma B^T Pe$$

we find that

$$\frac{d\tilde{\theta}}{dt} = \frac{d}{dt}(\theta - \hat{\theta}) = -\frac{d\hat{\theta}}{dt} = -\gamma B^T Pe$$

and the derivative of the Lyapunov function becomes

$$\frac{dV}{dt} = -e^T e$$

This function is negative as long as any component of the error vector is different from zero. With the control law given by (5.89) and (5.90) the tracking error will thus always go to zero. □

Backstepping

Unfortunately, adaptive feedback linearization cannot be applied to all systems that can be linearized by feedback. The reason is that higher derivatives of the parameter estimate will appear in the control law for systems of higher order. There is, however, another nonlinear design technique called *backstepping* that can be used. We first introduce this method and later show how it can be used for adaptive control. In feedback linearization we introduced new state variables and a nonlinear feedback so that the equations describing the transformed variables had a particular structure. A similar idea is used in backstepping, but the transformed equations have a different form. To show the key ideas without too many technical complications, we consider a simple stabilization problem. To simplify the writing, we frequently drop the arguments of functions.

EXAMPLE 5.17 Stabilization by backstepping

Consider the system described by

$$\begin{aligned}\frac{dx_1}{dt} &= x_2 + f(x_1)\\ \frac{dx_2}{dt} &= x_3\\ \frac{dx_3}{dt} &= u\end{aligned} \tag{5.91}$$

Introduce $\xi_1 = x_1$. Then

$$\frac{d\xi_1}{dt} = x_2 + f(\xi_1) = -\xi_1 + x_2 + \xi_1 + f(\xi_1)$$

If we introduce the function

$$a_1(\xi_1) = \xi_1 + f(\xi_1)$$

and the state variable

$$\xi_2 = x_2 + a_1(\xi_1) \tag{5.92}$$

the differential equation for ξ_1 can be written as

$$\frac{d\xi_1}{dt} = -\xi_1 + \xi_2$$

The derivative of the variable ξ_2 is given by

$$\frac{d\xi_2}{dt} = \frac{dx_2}{dt} + \frac{da_1}{d\xi_1}(-\xi_1 + \xi_2) = -\xi_2 + x_3 + \xi_2 + \frac{da_1}{d\xi_1}(-\xi_1 + \xi_2)$$

If we introduce the function

$$a_2(\xi_1, \xi_2) = \xi_2 + \frac{da_1}{d\xi_1}(-\xi_1 + \xi_2)$$

and the state variable

$$\xi_3 = x_3 + a_2(\xi_1, \xi_2)$$

the differential equation for ξ_2 can be written as

$$\frac{d\xi_2}{dt} = -\xi_2 + \xi_3$$

Taking derivatives of ξ_3 and using Eqs. (5.91), we find that

$$\frac{d\xi_3}{dt} = u + \frac{\partial a_2}{\partial \xi_1}(-\xi_1 + \xi_2) + \frac{\partial a_2}{\partial \xi_2}(-\xi_2 + \xi_3)$$

Introducing the function

$$a_3(\xi_1, \xi_2, \xi_3) = \xi_3 + \frac{\partial a_2}{\partial \xi_1}(-\xi_1 + \xi_2) + \frac{\partial a_2}{\partial \xi_2}(-\xi_2 + \xi_3)$$

we find that the differential equation for ξ_3 can be written as

$$\frac{d\xi_3}{dt} = -\xi_3 + a_3(\xi_1, \xi_2, \xi_3) + u$$

The feedback

$$u = -a_3(\xi_1, \xi_2, \xi_3)$$

gives the closed-loop system described by

$$\frac{d\xi}{dt} = \begin{pmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{pmatrix} \xi \tag{5.93}$$

This system is clearly stable, and its state ξ goes to zero exponentially. Notice that by a slight modification of the procedure we can have any number in the diagonal of the system matrix.

The transformation was obtained recursively. Notice that if the variable x_2 was a control variable that could be chosen freely, the "control law"

$$x_2 = -a_1(\xi_1)$$

would give

$$\frac{d\xi_1}{dt} = -\xi_1$$

The state variable ξ_2 defined by Eq. (5.92) can thus be interpreted as the difference between x_2 and the "stabilizing feedback" $-a_1(\xi_1)$.

Similarly, if x_3 was a control variable that could be chosen freely, the "control law"

$$x_3 = -a_2(\xi_1, \xi_2)$$

would give the closed-loop system

$$\frac{d\xi_1}{dt} = -\xi_1 + \xi_2$$

$$\frac{d\xi_2}{dt} = -\xi_2$$

The state variable ξ_3 can be interpreted as the difference between x_3 and the "stabilizing feedback" $-a_2(\xi_1, \xi_2)$.

The procedure was originally derived by applying this reasoning recursively, and the name "backstepping" derives from this.

In the example the system was transformed to a triangular form given by Eq. (5.93). There are many other possibilities. □

Adaptive Backstepping

The key idea of backstepping is to derive an error equation and to construct a control law and a parameter adjustment law such that the state of the error equation goes to zero. The idea is illustrated by a simple example.

EXAMPLE 5.18 Adaptive stabilization by backstepping

Consider the system

$$\frac{dx_1}{dt} = x_2 + \theta f(x_1)$$

$$\frac{dx_2}{dt} = x_3$$

$$\frac{dx_3}{dt} = u$$

where f is a known function and θ an unknown parameter. We derive a control law that stabilizes the system when the parameter θ is unknown. Introduce a new state variable $\xi_1 = x_1$. We write the derivative of ξ_1 as a sum of terms in which one of them depends on known quantities only. For this purpose we introduce the parameter estimate $\hat{\theta}$ and the error $\tilde{\theta} = \theta - \hat{\theta}$. The derivative of ξ_1 then becomes

$$\frac{d\xi_1}{dt} = -\xi_1 + \xi_1 + x_2 + \hat{\theta} f(\xi_1) + \tilde{\theta} f(\xi_1)$$

Introduce the next state variable ξ_2 as

$$\xi_2 = x_2 + a_1(\xi_1, \hat{\theta})$$

where

$$a_1(\xi_1, \hat{\theta}) = \xi_1 + \hat{\theta} f(\xi_1) \tag{5.94}$$

The differential equation for ξ_1 can then be written as

$$\frac{d\xi_1}{dt} = -\xi_1 + \xi_2 + \tilde{\theta} f \tag{5.95}$$

We now proceed to rewrite the derivative of ξ_2 as a sum of two terms in which the first depends only on ξ_1, ξ_2, and $\hat{\theta}$. Hence

$$\frac{d\xi_2}{dt} = \frac{dx_2}{dt} + \frac{\partial a_1}{\partial \xi_1} \cdot \frac{d\xi_1}{dt} + \frac{\partial a_1}{\partial \hat{\theta}} \cdot \frac{d\hat{\theta}}{dt}$$

Equation (5.95) gives the desired separation of terms in $d\xi_1/dt$. Some work is required to obtain a similar expression for $d\hat{\theta}/dt$. We have

$$\frac{d\xi_2}{dt} = x_3 + \frac{\partial a_1}{\partial \xi_1}\left(-\xi_1 + \xi_2 + \tilde{\theta} f\right) + \frac{\partial a_1}{\partial \hat{\theta}} \cdot \frac{d\hat{\theta}}{dt} \tag{5.96}$$

Following the idea of backstepping, we consider x_3 to be a control variable that can be chosen freely. The Lyapunov function

$$2V = \xi_1^2 + \xi_2^2 + \tilde{\theta}^2$$

can be used to find a control law and an adaptation law that stabilizes the error equation for variables ξ_1 and ξ_2. After some calculations we find that the derivative of V is given by

$$\frac{dV}{dt} = -\xi_1^2 + \xi_1\xi_2 + x_3\left(\xi_2 + \frac{\partial a_1}{\partial \hat{\theta}}\frac{d\hat{\theta}}{dt}\right) + \tilde{\theta}\left(\xi_1 f + \xi_2\frac{\partial a_1}{\partial \xi_1}f(\xi_1) - \frac{d\hat{\theta}}{dt}\right)$$

The term containing $\tilde{\theta}$ can be eliminated by choosing

$$\frac{d\hat{\theta}}{dt} = b_2(\xi_1, \xi_2)$$

where

$$b_2 = \xi_1 f(\xi_1) + \xi_2\frac{\partial a_2}{\partial \xi_1}f(\xi_1) \tag{5.97}$$

The function $b_2(\xi_1, \xi_2)$ can be interpreted as a good way to choose the parameter update rate $d\hat{\theta}/dt$ based on ξ_1 and ξ_2. The "control variable" x_3 can be chosen to give

$$\frac{dV}{dt} = -\xi_1^2 - \xi_2^2$$

Using b_2 as an estimate of $d\hat{\theta}/dt$, we now rewrite Eq. (5.96) as

$$\begin{aligned}\frac{d\xi_2}{dt} = &-\xi_1 - \xi_2 + x_3 + \xi_1 + \xi_2 + \frac{\partial a_1}{\partial \xi_1}(-\xi_1 + \xi_2 + \tilde{\theta}f)\\ &+ \frac{\partial a_1}{\partial \hat{\theta}}b_2 + \frac{\partial a_1}{\partial \hat{\theta}}\left(\frac{d\hat{\theta}}{dt} - b_2\right)\end{aligned} \tag{5.98}$$

Now define

$$a_2\left(\xi_1, \xi_2, \hat{\theta}\right) = \xi_1 + \xi_2 + \frac{\partial a_1}{\partial \xi_1}\left(-\xi_1 + \xi_2\right) + \frac{\partial a_1}{\partial \hat{\theta}}b_2 \tag{5.99}$$

and introduce the state variable ξ_3 as

$$\xi_3 = x_3 + a_2(\xi_1, \xi_2, \hat{\theta})$$

The differential equation (5.98) can be written as

$$\frac{d\xi_2}{dt} = -\xi_1 - \xi_2 + \xi_3 + \frac{\partial a_1}{\partial \xi_1}\tilde{\theta}f + \frac{\partial a_1}{\partial \hat{\theta}}\left(\frac{d\hat{\theta}}{dt} - b_2\right) \tag{5.100}$$

The derivative of ξ_3 becomes

$$\frac{d\xi_3}{dt} = u + \frac{\partial a_2}{\partial \xi_1}\cdot\frac{d\xi_1}{dt} + \frac{\partial a_2}{\partial \xi_2}\cdot\frac{d\xi_2}{dt} + \frac{\partial a_2}{\partial \hat{\theta}}\cdot\frac{d\hat{\theta}}{dt} \tag{5.101}$$

Notice that the control variable u now appears explicitly on the right-hand side. In the stabilization problem the error is equal to the vector ξ and the error equation is obtained by combining Eqs. (5.95), (5.100), and (5.101). Following the general MRAS approach, we now attempt to find a feedback law and a parameter adjustment rule that stabilizes the error equation. Choosing

$$2V = \xi_1^2 + \xi_2^2 + \xi_3^2 + \tilde{\theta}^2$$

as a possible Lyapunov function, we get, after straightforward but tedious calculations,

$$\begin{aligned}\frac{dV}{dt} = &-\xi_1^2 - \xi_2^2 + \xi_2\xi_3 + \xi_2\frac{\partial a_1}{\partial \hat{\theta}}\left(\frac{d\hat{\theta}}{dt} - b_2\right)\\ &+ \xi_3\left(u + \frac{\partial a_1}{\partial \hat{\theta}}\left(\frac{d\hat{\theta}}{dt} - b_2\right) + \frac{\partial a_2}{\partial \hat{\theta}}\frac{d\hat{\theta}}{dt}\right)\\ &+ \tilde{\theta}\left(\xi_1 f + \xi_2\frac{\partial a_1}{\partial \xi_1}f + \xi_3\left(\frac{\partial a_2}{\partial \xi_1} + \frac{\partial a_1}{\partial \xi_1}\frac{\partial a_2}{\partial \xi_2}\right)f - \frac{d\hat{\theta}}{dt}\right)\end{aligned}$$

The term that contains $\tilde{\theta}$ can be eliminated by updating the parameters in the following way:

$$\frac{d\hat{\theta}}{dt} = \xi_1 f + \xi_2\frac{\partial a_1}{\partial \xi_1}f + c(\xi_1, \xi_2)\xi_3 \tag{5.102}$$

where

$$c(\xi_1, \xi_2) = \left(\frac{\partial a_2}{\partial \xi_1} + \frac{\partial a_1}{\partial \xi_1}\frac{\partial a_2}{\partial \xi_2}\right)f$$

Furthermore, introducing

$$b_3(\xi_1, \xi_2, \xi_3) = b_2 + c\xi_3$$

and

$$a_3 = \xi_2 + \xi_3 + \frac{\partial a_2}{\partial \xi_1}(-\xi_1 + \xi_2) + \frac{\partial a_2}{\partial \xi_2}\left(-\xi_1 \quad \xi_2 + \xi_3 - \xi_3^2\frac{\partial a_1}{\partial \hat{\theta}}c\right) + \frac{\partial a_2}{\partial \hat{\theta}}b_3$$

we find that

$$\frac{d\hat{\theta}}{dt} - b_2 = c\xi_3$$

The derivative of the Lyapunov function can then be written as

$$\frac{dV}{dt} = -\xi_1^2 - \xi_2^2 - \xi_3^2 + \xi_3(u + a_3)$$

The feedback law

$$u = -a_3(\xi_1, \xi_2, \xi_3) \tag{5.103}$$

gives

$$\frac{dV}{dt} = -\xi_1^2 - \xi_2^2 - \xi_3^2$$

and we find that dV/dt is negative as long as $|\xi| \neq 0$. □

Summary

The examples given should give some of the flavor of nonlinear adaptive control. The results obtained depend on clever changes of coordinates. A reasonable characterization of the class of systems in which the methods apply is not available. Nevertheless, we can make some interesting observations from the examples. First, we can notice that the adaptive control laws that are obtained differ significantly from those obtained from the certainty equivalence principle. In the nonlinear approaches the control law and the rule for updating the parameters are obtained simultaneously. An estimate of the rate of change of the parameters appears in the feedback law. Many problems remain to be solved.

5.11 CONCLUSIONS

The fundamental ideas behind the MRAS have been covered in this chapter, including

- Gradient methods,
- Lyapunov and passivity design, and
- Augmented error.

In all cases the rule for updating the parameters is of the form

$$\frac{d\theta}{dt} = \gamma\varphi\varepsilon$$

or, in the normalized form,

$$\frac{d\theta}{dt} = \gamma\frac{\varphi\varepsilon}{\alpha + \varphi^T\varphi}$$

In the gradient method the vector φ is the negative gradient of the error with respect to the parameters. Estimation of parameters or approximations may be needed to obtain the gradient. In other cases, φ is a regression vector, which is found by filtering inputs, outputs, and command signals. The quantity ε is the augmented error, which also can be interpreted as the prediction error of the estimation problem. It is customary to use an augmented error that is linear in the parameters.

The gradient method is flexible and simple to apply to any system structure. The calculations required are the determination of the sensitivity derivative. Since the sensitivity derivative cannot be obtained for an unknown process, it is necessary to make several approximations. The initial values of the parameters must be such that the closed-loop system is stable. Empirical evidence indicates that the system is stable for small adaptation gains but that high gains lead to instability. It is difficult to find the bounds. In Chapter 6 we give more insight into the properties of the gradient method.

A general MRAS is derived in Section 5.8 on the basis of the model-following design in Chapter 3. This algorithm includes as special cases many of the MRAS designs given in the literature. The estimation of the parameters can be done in several ways other than those given in Eqs. (5.62) and (5.63). Various modifications are discussed in Chapter 6.

PROBLEMS

5.1 Consider the process

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

where a is an unknown parameter. Determine a controller that can give the closed-loop system

$$G_m(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

Determine model-reference adaptive controllers based on gradient and stability theory, respectively. (Compare Problem 3.2.)

5.2 Consider the simple MRAS in Fig. 5.4 with $G = 1/s$. Let the parameter adjustment law be Eq. (5.57) (i.e., of PI type). Determine the differential equation for θ, and discuss how γ_1 and γ_2 influence the convergence rate.

5.3 Consider a position servo described by

$$\frac{dv}{dt} = -av + bu$$

$$\frac{dy}{dt} = v$$

where parameters a and b are unknown. Assume that the control law

$$u = \theta_1(u_c - y) - \theta_2 v$$

is used and that it is desired to control the system in such a way that the transfer function from command signal to process output is given by

$$G_m(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

Determine an adaptive control law that adjusts parameter θ_1 and θ_2 so that the desired objective is obtained.

5.4 An integrator

$$G_p(s) = \frac{b}{s}$$

is to be controlled by a zero-order continuous-time controller

$$u(t) = s_0 y(t) + t_0 u_c(t)$$

The desired response model is given by

$$G_m(s) = \frac{b_m}{s + a_m}$$

Derive, using the Lyapunov theory, a parameter update law of an MRAS guaranteeing that the error $e = y - y_m$ goes to zero. Try the Lyapunov function

$$V(x) = \frac{1}{2}\left(e^2 + \frac{1}{b}\left(bs_0 - a_m\right)^2 + \frac{1}{b}\left(bt_0 - b_m\right)^2\right)$$

where

$$e(t) = y(t) - y_m(t)$$

5.5 Consider the problem of adaptation of a feedforward gain in Example 5.1 when

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

(a) Introduce the augmented error, and determine an MRAS based on stability theory.

(b) Show that the derived adaptation law in part (a) gives a stable closed-loop system.

5.6 Determine conditions in which a second-order transfer function

$$G(s) = \frac{b_0 s^2 + b_1 s + b_2}{s^2 + a_1 s + a_2}$$

is strictly positive real.

5.7 Show that $B(s)/A(s)$ is SPR if $A(s)$ is a stable polynomial and the B polynomial is the first row of the P-matrix defined by the Lyapunov equation

$$A^T P + PA = -Q$$

where the matrix A is

$$A = \begin{pmatrix} -a_1 & -a_2 & \dots & -a_{n-1} & -a_n \\ 1 & 0 & & 0 & 0 \\ \vdots & & \ddots & & \\ 0 & 0 & & 1 & 0 \end{pmatrix}$$

and Q is a symmetric positive definite matrix. Show that the system of equations for solving p_1, p_2, and p_3 in Example 5.6 has a unique solution only if all the eigenvalues of A are in the left half-plane.

5.8 Show that the transfer function

$$G(s) = 1 + s$$

is SPR and ISP but not OSP.

5.9 Show that the transfer function

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

is SPR and OSP but not ISP.

5.10 Show that the transfer function

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

is OSP and ISP but not SPR.

5.11 Consider the system

$$G(s) = G_1(s)G_2(s)$$

where

$$G_1(s) = \frac{b}{s+a}$$
$$G_2(s) = \frac{c}{s+d}$$

where a and b are unknown parameters and c and d are known. Discuss how to make an MRAS based on the gradient approach. (Compare Problem 3.3.) Let the desired model be described by

$$G_m(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

5.12 A process has the transfer function

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

where b is a time-varying parameter. The system is controlled by a proportional controller

$$u(t) = k\,(u_c(t) - y(t))$$

It is desirable to choose the feedback gain so that the closed-loop system has the transfer function

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

Design an MRAS that gives the desired result, and investigate the system by simulation. (Compare Problem 3.4.)

5.13 The general MRAS procedure in Section 5.8 was derived for known instantaneous gain b_0. If b_0 is unknown, we may use the following augmented error:

$$\varepsilon = \frac{Q}{A_0 A_m}\left(\left(b_0 - \hat{b}_0\right)\left(\varphi^T\theta + \frac{u}{P_1}\right) + b_0\varphi^T\left(\theta - \theta^0\right)\right)$$

where $\hat{b}_0$ is the estimate of b_0. Discuss how this augmented error can be obtained and how it may be used to update the parameters b_0 and θ.

5.14 Study the parameter adjustment law in Example 5.2. Make a simulation program that implements the adaptive system. Repeat the simulation in Fig. 5.5. Investigate the behavior of the parameters and the error. Explore how the behavior is influenced by the adaptation gain γ.

5.15 Repeat the simulation in Problem 5.4 for different types of input signals. Change the amplitude and the nature of the signals. Can you find values of the adaptation gain that work well for different inputs?

5.16 Consider the system in Example 5.5. Assume that u_c is a step that implies that y_m will be time-varying. Investigate by analysis or simulate the stability limit and compare with the limit obtained in the example, in which u_c and y_m were constant.

5.17 Consider a first-order system with the transfer function

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

where a and b are unknown parameters. Assume that the system is controlled by the control law

$$u = \theta_1 u_c - \theta_2 y$$

Compare by simulation the properties of the systems obtained with the MIT rule and the one derived from Lyapunov theory. Use the same parameter values as in Example 5.2. (*Hint*: The algorithms are given in Examples 5.2 and 5.7.)

5.18 Investigate the properties of the system in Example 5.7 by simulation.

5.19 Investigate through simulation the convergence rate of the parameters in Example 5.2 when the control law of Eqs. (5.9) is used. How will the parameter adjustment change if an adaptation rule based on stability theory is used? For instance, plot the phase plane for the parameters.

5.20 Consider the process

$$G(s) = \frac{50}{s(s+4)}$$

and the criterion

$$\int_0^\infty \left((y-u_c)^2 + \rho u^2\right)\, dt$$

Let the control law have the form

$$u(t) = -s_o(y - u_c)$$

or

$$u(t) = -\frac{s_0 p + s_1}{p + r_1}(y - u_c)$$

Determine the controller parameters through explicit minimization of the criterion, and let the gradients be obtained from an estimated model of the process. (*Hint*: See Trulsson and Ljung, 1985.)

5.21 Consider the system in Example 5.14. Figure 5.22(c) shows the rapid decrease in the error, while the parameters converge much more slowly. Explain the slow parameter convergence by analyzing the sensitivity of the closed-loop poles with respect to the estimated parameters.

5.22 Consider a system described by

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

where a and b are unknown parameters. Find a simple control law that can control the plant well, and derive an adaptive algorithm that gives good performance.

REFERENCES

The model-reference approach was developed by Whitaker and his colleagues around 1958. One early reference giving the basic ideas using the gradient method is:

Osburn, P. V., H. P. Whitaker, and A. Kezer, 1961. "New developments in the design of adaptive control systems." Paper No 61-39, February 1961, Institute of Aeronautical Sciences.

The problem with stability of the gradient method was analyzed by using Lyapunov stability theory in:

Butchart, R. L., and B. Shackcloth, 1965. "Synthesis of model reference adaptive control systems by Lyapunov's second method." *Proceedings of the 1965 IFAC Symposium on Adaptive Control.* Teddington, U.K.

and explored further in:

Parks, P. C., 1966. "Lyapunov redesign of model reference adaptive control systems." *IEEE Trans. Automat. Contr.* **AC-11**: 362–365.

The different approaches to MRAS are treated in:

Landau, Y. D., 1979. *Adaptive Control: The Model Reference Approach.* New York: Marcel Dekker.

Parks, P. C., 1981. "Stability and convergence of adaptive controllers: Continuous systems." In *Self-tuning and Adaptive Control: Theory and Applications*, eds. C. J. Harris and S. A. Billings. Stevenage, U.K.: Peter Peregrinus.

A comparison of the Lyapunov and the input-output stability approaches is given in:

Narendra, K. S., and L. S. Valavani, 1980. "A comparison of Lyapunov and hyperstability approaches to adaptive control of continuous systems." *IEEE Trans. Automat. Contr.* **AC-25**: 243–247.

The augmented error method was introduced in:

Monopoli, R. V., 1974. "Model reference adaptive control with an augmented error signal." *IEEE Trans. Automat. Contr.* **AC-19**: 474–484.

A unification of MRAS and self-tuning controllers is found in:

Egardt, B., 1979. "Unification of some continuous-time adaptive control schemes." *IEEE Trans. Automat. Contr.* **AC-24**: 588–592.

Stability of continuous-time MRAS is discussed in:

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

Goodwin, G. C., and D. Q. Mayne, 1987. "A parameter estimation perspective of continuous time model reference adaptive control." *Automatica* **23**: 57–70.

The main problem in the stability analysis of adaptive controllers is the boundedness of the variables of the system. Proofs of boundedness and stability are found in:

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

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

The error model plays an important role in the analysis of the MRAS. Different generic error models are discussed in:

Narendra, K. S., and Y.-H. Lin, 1980. "Design of stable model reference adaptive controllers." In *Applications of Adaptive Control*, eds. K. S. Narendra and R. V. Monopoli. New York: Academic Press.

PI adjustment of the parameters in the MRAS is discussed in Landau (1979) above and in:

Hang, C. C., and P. C. Parks, 1973. "Comparative studies of model reference adaptive control systems." *IEEE Trans. Automat. Contr.* **AC-18**: 419–428.

Textbooks on MRAS are, for instance, Landau (1979) and:

Narendra, K. S., and A. M. Annaswamy, 1989. *Stable Adaptive Systems.* Englewood Cliffs, N.J.: Prentice-Hall.

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

Lyapunov theory and passivity theory are basic tools for the stability analysis. Some general references are:

Hahn, W., 1967. *Stability of Motion.* Berlin: Springer-Verlag.

Popov, V. M., 1973. *Hyperstability of Control Systems.* Berlin: Springer-Verlag.

Vidyasagar, M., 1978. *Nonlinear Systems Analysis.* Englewood Cliffs, N.J.: Prentice-Hall.

Vidyasagar, M., 1986. "New directions of research in nonlinear system theory." *Proceedings IEEE* **74**: 1060–1091.

Slotine, J.-J. E., and W. Li, 1991. *Applied Nonlinear Control.* Englewood Cliffs, N.J.: Prentice-Hall.

Khalil, H. K., 1992. *Nonlinear Systems.* New York: Macmillan.

Early work on passivity and input-output stability was done by Popov and a little later by Zames and Sandberg. The theory is summarized in:

Desoer, C. A., and M. Vidyasagar, 1975. *Feedback Systems: Input-output Properties.* New York: Academic Press.

A proof of the Popov-Kalman-Yakubovich lemma is given in:

Lefschetz, S., 1965. *Stability of Nonlinear Control Systems*, pp. 114–118. New York: Academic Press.

Discrete-time MRAS is discussed in detail in Egardt (1979) and Landau (1979), above.

The explicit criterion minimization approach to adaptive control can be found in:

Tsypkin, Y. Z., 1971. *Adaptation and Learning in Automatic Systems.* New York: Academic Press.

Goodwin, G. C., and P. J. Ramadge, 1979. "Design of restricted complexity adaptive regulators." *IEEE Trans. Automat. Contr.* **AC-24**: 584–588.

Trulsson, E., and L. Ljung, 1985. "Adaptive control based on explicit criterion minimization." *Automatica* **21**: 385–399.

The backstepping method was invented by Kokotovic and his students. An overview of the method is given in the 1991 Bode Lecture; see:

Kokotovic, P. V., 1992. "The joy of feedback: nonlinear and adaptive control." *IEEE Control Systems Magazine* **12**(3): 7–17.

Additional details are found in:

Kokotovic, P. V., ed., 1991. *Foundations of Adaptive Control.* Berlin: Springer-Verlag.

Kokotovic, P. V., I. Kanellakopoulos, and M. Krstić, 1992. "On letting adaptive control be what it is: nonlinear feedback." *Proceedings of the IFAC Symposium on Adaptive Control and Adaptive Signal Processing.* Grenoble, France.

Kokotovic, P. V., M. Krstić, and I. Kanellakopoulos, 1992. "Backstepping to passivity: recursive design of adaptive systems." *Proceedings of the IEEE Conference on Decision and Control*, pp. 3276–3280. Tucson, Arizona.

CHAPTER 6

PROPERTIES OF ADAPTIVE SYSTEMS

6.1 INTRODUCTION

Some theoretical problems were discussed in earlier chapters in connection with description or derivation of specific algorithms. In particular we used equilibrium analysis to analyze the self-tuners and stability theory to derive some model-reference algorithms. In this chapter we attempt to bring together theory of a more general character. The theory has several different goals:

- To present some mathematical tools that are useful in analysis of adaptive systems.
- To analyze the behavior of adaptive systems in nonideal cases.
- To give ideas for new algorithms and for improvement of old algorithms.

In this chapter we focus on the first two issues. The behavior of specific algorithms can be understood through analysis of stability, convergence, and performance. Stability proofs require certain assumptions. It is also of considerable interest to understand what happens when the assumptions are violated. Analysis of performance may give useful insight into performance limits; it is helpful to know whether the performance of a particular algorithm is close to the theoretical limits. A good theory should also give clues to the construction of new algorithms.

Unfortunately, there is no collection of results that can be called a theory of adaptive control in the sense specified. There is instead a scattered body of results, which gives only partial results. One reason for this is that the behavior of adaptive systems is quite complex because of their time-varying and nonlinear character. Readers who are familiar only with linear systems

theory, in which most problems can be answered in great detail, should thus be warned.

The closed-loop systems obtained with adaptive control are nonlinear and sometimes also stochastic. Such systems are also very difficult to analyze. To obtain some insight with a reasonable effort, it is therefore necessary to make some simplifications. It is often possible to analyze the equilibrium conditions. The local behavior in the neighborhood of the equilibria can also be explored by using linearization. The global behavior of the systems can, however, be very complex, particularly if the design parameters are chosen badly.

In Section 6.2 we show that the adaptive control problem has a special nonlinear structure that can be exploited in the analysis. We first show that very complex, even chaotic, behavior can be observed if the adaptation gain is chosen to be too high.

Section 6.3 presents an analysis of a system with adaptation of a feedforward gain. Such systems can be described by linear time-varying systems in which the time variation originates from the command signal. The particular case of periodic variations can be dealt with by so-called Floquet theory. The analysis reveals that very complex behavior can be obtained even in this simple case.

The properties of indirect discrete-time adaptive systems are investigated in Section 6.4. In this case it is natural to investigate parameter estimation and the control design separately. There is interaction between these problems because the identification is done in closed loop and the control design influences the signals generated by feedback. The analysis brings out the importance of persistency of excitation and the dangers with singularities in the control design. A consequence of this is that it is desirable to have as few parameters as possible and to have external excitation. In Section 6.5 we make a similar analysis of the direct algorithm. One of the conditions required for the proof is that the complexity of the model used must be at least as complex as the process to be controlled. A characteristic feature of direct adaptive algorithms is that the closed-loop behavior can converge to the desired behavior even if the parameters do not converge.

It is reasonable to assume that if the adaptation rate is small, the parameter estimates will change more slowly than the other variables in a system. The closed-loop system can then be viewed as having different time scales. This has been emphasized in the descriptions of both the self-tuning regulator and the model reference adaptive controller. The analysis can then be simplified by considering the slow and fast modes separately. Averaging analysis is a good analytical tool for this. A short presentation of this technique is given in Section 6.6. A significant advantage of the averaging technique is that it makes it possible to reduce the dimensionality of the problem to the number of parameters in the algorithm. The averaging method also makes it possible to explore the behavior in detail. A drawback of the averaging results is that they hold for small adaptation gains but the theory does not give quantitative results about smallness.

It is a characteristic feature of feedback that a controller can often be designed by using a simplified model of a real process. This is one of the reasons why automatic control has been so successful in applications. So far, we have analyzed the behavior of some adaptive algorithms under the simplifying assumption that the structure of the process is the same as the model used to design the adaptive controller. Having obtained the tool of averaging, we are in a position to investigate the consequences of the simplifying assumptions made in the earlier sections, and we can explore how adaptive systems behave in the presence of unmodeled dynamics, that is, when the order of the process is different from that of the model used to derive the adaptive controller. Analysis of several examples in Section 6.7 leads us to various modifications of the algorithms that will improve their robustness to unmodeled dynamics.

In Section 6.8 we show that averaging techniques can be used to analyze stochastic self-tuning regulators. The equilibrium points of the algorithms and their local behavior can often be obtained without too much effort. In Section 6.9, different ways are discussed to make the adaptive algorithms robust with respect to the assumptions made in the idealized cases.

6.2 NONLINEAR DYNAMICS

We have mentioned several times that adaptive systems are inherently nonlinear. A natural approach to understand the behavior of adaptive systems is thus to use tools from the theory of nonlinear dynamical systems. We first investigate the structure of adaptive systems. This reveals that they have a very special structure. Some tools from dynamical systems are then reviewed briefly, and we apply them to a very simple system. This analysis reveals that adaptive systems behave in the expected way when the adaptation gain is small. However, the behavior can be very complex for large adaptation gains. The analysis also indicates the difficulties involved in the approach. We also investigate the special case of adaptation of a feedforward gain. In this case the problem is simplified significantly because it can be described as a linear time-varying system. A reasonably complete analysis can be performed when the command signal is periodic. This analysis reveals that the system is well behaved for small adaptation gains but that the behavior is quite complex for large adaptation gains.

Structure of Equations Describing Adaptive Systems

Consider a process controlled by an indirect adaptive controller as shown in Fig. 3.1. We will first consider the case in which parameters of a continuous-time model are estimated by using a gradient procedure. Assume that the system to be controlled is linear. Let ϑ denote the controller parameters and ν the external driving signals. The signal ν is typically composed of the command

signal u_c and nonmeasurable disturbances acting on the process. With constant controller parameters the closed-loop system can be written as

$$\begin{aligned} \frac{d\xi}{dt} &= A(\vartheta)\xi + B(\vartheta)\nu \\ \eta &= \begin{pmatrix} e \\ \varphi \end{pmatrix} = C(\vartheta)\xi + D(\vartheta)\nu \end{aligned} \tag{6.1}$$

The state vector ξ includes the states of the system, the reference model, the data filter, and the auxiliary state variables that may have to be introduced to calculate the error e and the regression vector φ used in the parameter adjustment mechanism. The vector η consists of the error and the regression vector that are used by the parameter estimator.

Furthermore, let θ denote the process parameters. A normalized gradient scheme for estimating the parameters can be described by

$$\frac{d\hat{\theta}}{dt} = \gamma \frac{\varphi(\vartheta,\xi)e(\vartheta,\xi)}{\alpha + \varphi(\vartheta,\xi)^T\varphi(\vartheta,\xi)} \tag{6.2}$$

The control design can be represented by a nonlinear function $\vartheta = \chi(\hat{\theta})$, which maps the estimated parameters into controller parameters. This map becomes the identity for direct algorithms.

For constant ϑ the system (6.1) is linear. The solution can then also be characterized by the operators $G_{e\nu}$ and $G_{\varphi\nu}$, which relate e and φ to ν. These operators depend on the controller parameters ϑ. Equation (6.2) can then be written as

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

The adaptive system is thus described by Eqs. (6.1) and (6.2), which have a very special structure. Equation (6.1) is linear in the states and the external driving signals. The controller parameters appear in the coefficients of matrices A, B, C, and D. Nonlinearities appear in the product φe in Eq. (6.2), in the design map χ, and in the functions $A(\vartheta)$, $B(\vartheta)$, $C(\vartheta)$, and $D(\vartheta)$ in Eq. (6.1). The equations for an adaptive system have a similar form in the discrete-time case. For a system with recursive least-squares estimation the equations can be written as

$$\begin{aligned} \xi(t+1) &= A(\vartheta)\xi(t) + B(\vartheta)\nu(t) \\ \eta(t) &= \begin{pmatrix} e(t) \\ \varphi(t) \end{pmatrix} = C(\vartheta)\xi(t) + D(\vartheta)\nu(t) \\ \hat{\theta}(t+1) &= \hat{\theta}(t) + P(t+1)\varphi(t)e(t) \\ P(t+1) &= P(t) - P(t)\varphi(t)(\lambda + \varphi^T(t)P(t)\varphi(t))^{-1}\varphi^T(t)P(t) \end{aligned} \tag{6.3}$$

It is useful to try to exploit the special structure of the equations to get a deeper understanding of adaptive systems. One special feature is that the state of the closed-loop system is naturally separated into two parts, ξ and $\hat{\theta}$. Moreover, it is reasonable to assume that $\hat{\theta}$ changes more slowly than ξ.

Analysis

Let us briefly summarize how a nonlinear system such as Eqs. (6.1) and (6.2) or Eqs. (6.3) can be analyzed. It is a comparatively simple task to find the equilibrium solutions by solving the algebraic equations

$$\frac{d\xi}{dt} = 0$$
$$\frac{d\hat{\theta}}{dt} = 0$$

for continuous-time systems. For discrete-time systems the equivalent equations become

$$\xi(t+1) = \xi(t)$$
$$\hat{\theta}(t+1) = \hat{\theta}(t)$$

It may happen that proper equilibria do not exist. In such cases there may be integral manifolds where the parameters $\hat{\theta}$ are constant although the state ξ varies with time. We are then led to averaging analysis, which is discussed in depth in Section 6.6. Equilibria having been found, it is natural to determine the local behavior by linearizing the equations around the equilibria and applying standard linear theory. A complication is that critical cases in which the eigenvalues are zero frequently occur. Having determined possible equilibria, we can proceed to investigate how the nature of the equilibria changes with important parameters of the system. It is of particular interest to investigate changes in which the nature of the local equilibria changes (bifurcation analysis). When the local properties are investigated, it is natural to proceed to find the global properties. There are no general tools for this, and we have to resort to simulations and approximations. Phase plane analysis is useful for two-dimensional systems.

Analysis of a Simple Discrete-Time System

To illustrate how the analysis can be done, we discuss a simple example. Consider a discrete-time adaptive controller that is based on estimation of the parameter θ in the model

$$y(t+1) = \theta y(t) + u(t) \tag{6.4}$$

Let the controller be

$$u(t) = -\hat{\theta}(t)y(t) + y_0 \tag{6.5}$$

where $\hat{\theta}$ is an estimate of θ and y_0 the setpoint. If the process is indeed described by Eq. (6.4) and if the estimate $\hat{\theta}$ is correct, the controller gives a

deadbeat response. The parameter is estimated by using a normalized gradient algorithm

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \gamma \frac{y(t)\Big(y(t+1) - \hat{\theta}(t)y(t) - u(t)\Big)}{\alpha + y^2(t)} \tag{6.6}$$

where γ and α are parameters. This is equal to Kaczmarz's projection algorithm if $\gamma = 1$ and $\alpha = 0$.

To analyze the closed-loop system, we must also have a description of the actual process. We assume that this is given by

$$y(t+1) = \theta_0 y(t) + a + u(t) \tag{6.7}$$

Notice that, because of the presence of the parameter a on the right-hand side, this model is different from the model (6.4) used to design the adaptive controller. Equations (6.4), (6.5), (6.6), and (6.7) thus describe a very simple case of adaptive control of a process with a constant unmodeled disturbance. Using Eq. (6.5) to eliminate u in Eqs. (6.6) and (6.7), we find that the closed-loop system can be described by the equations

$$\begin{aligned} y(t+1) &= \Big(\theta_0 - \hat{\theta}(t)\Big)y(t) + a + y_0 \\ \hat{\theta}(t+1) &= \hat{\theta}(t) + \gamma \frac{y(t)\Big(\Big(\theta_0 - \hat{\theta}(t)\Big)y(t) + a\Big)}{\alpha + y^2(t)} \end{aligned} \tag{6.8}$$

This is a second-order nonlinear system. To explore the behavior of this system, we follow the procedure of nonlinear analysis.

Equilibrium Analysis Equations (6.8) have the equilibrium solution

$$\begin{aligned} y &= y_0 \\ \hat{\theta} &= \theta_0 + \frac{a}{y_0} \end{aligned} \tag{6.9}$$

Notice that the equilibrium value of the output is always equal to the setpoint in spite of the disturbance. This is a phenomenon that we have observed before in adaptive systems. (Compare with Example 3.5 and Example 5.2.) Unmodeled dynamics, however, give a parameter error.

Linearizing Eqs. (6.8) around the equilibrium equations (6.9), we find that the system matrix is

$$A = \begin{pmatrix} -\dfrac{a}{y_0} & -y_0 \\ -\gamma\dfrac{a}{\alpha + y_0^2} & 1 - \gamma\dfrac{y_0^2}{\alpha + y_0^2} \end{pmatrix} \tag{6.10}$$

This matrix has the characteristic polynomial

$$z^2 + a_1 z + a_2$$

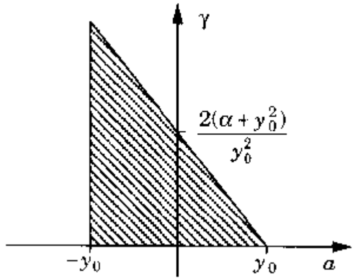

Figure 6.1 Stability region for the closed-loop system.

where

$$a_1 = \frac{a}{y_0} - 1 + \gamma \frac{y_0^2}{\alpha + y_0^2}$$
$$a_2 = -\frac{a}{y_0}$$

It follows from the stability criterion for discrete-time systems (Schur-Cohn) that the characteristic polynomial has all its roots inside the unit disc if

$$a_2 < 1$$
$$a_2 - a_1 + 1 > 0$$
$$a_2 + a_1 + 1 > 0$$

Inserting the expressions for a_1 and a_2 into these conditions gives

$$\begin{aligned} &\text{(i)} \quad && \frac{a}{y_0} > -1 \\ &\text{(ii)} \quad && \gamma < 2\frac{(1 - a/y_0)(\alpha + y_0^2)}{y_0^2} \\ &\text{(iii)} \quad && \gamma > 0 \end{aligned} \tag{6.11}$$

The equilibrium is stable if parameters a and γ are inside the triangular region shown in Fig. 6.1. To have a stable equilibrium, it must thus be required that the magnitude of the disturbance a is less than the magnitude of the command signal y_0. In addition the adaptation gain γ should not be too large. It is interesting to see the consequences of unmodeled dynamics. If there are no unmodeled dynamics ($a = 0$), then the condition for local stability of the equilibrium becomes

$$0 < \gamma < 2\frac{\alpha + y_0^2}{y_0^2}$$

Stability is thus guaranteed simply by choosing a reasonable value of γ.

Global Properties We now investigate the global properties when the parameters are chosen in such a way that the equilibrium is stable. To get some guidelines for the analysis, we first simulate the system. In Fig. 6.2 we show a phase portrait for the case in which $\alpha = 0.1$, $\gamma = 0.1$, $\theta_0 = 1.5$, $y_0 = 1$, and $a = 0.9$. It follows from Eqs. (6.9) that the equations have an equilibrium for $y = 1.0$ and $\hat{\theta} = 2.4$ and from condition (ii) in Eqs. (6.11) that the equilibrium is stable provided that $0 < \gamma < 0.22$. The equilibrium is thus stable for the chosen value of the adaptation gain. Remember that the system is a discrete-time system. The discrete solution points are connected with straight lines to give a continuous graph. All trajectories shown in the simulation are approaching the equilibrium. Solutions with initial values $\hat{\theta}(0) = 0$ appear to have large excursions, and the trajectory with $\hat{\theta}(0) = 2.5$ seems to be oscillatory. To understand the behavior intuitively, we consider the equations for y and $\hat{\theta}$ separately. It follows from Eqs. (6.8) that if $\hat{\theta}$ is constant, then the motion of y is governed by

$$y(t+1) = (\theta_0 - \hat{\theta})y(t) + a + y_0$$

This is a first-order difference equation with the equilibrium solution

$$y = f(\hat{\theta}) = \frac{a + y_0}{1 + \hat{\theta} - \theta_0} \tag{6.12}$$

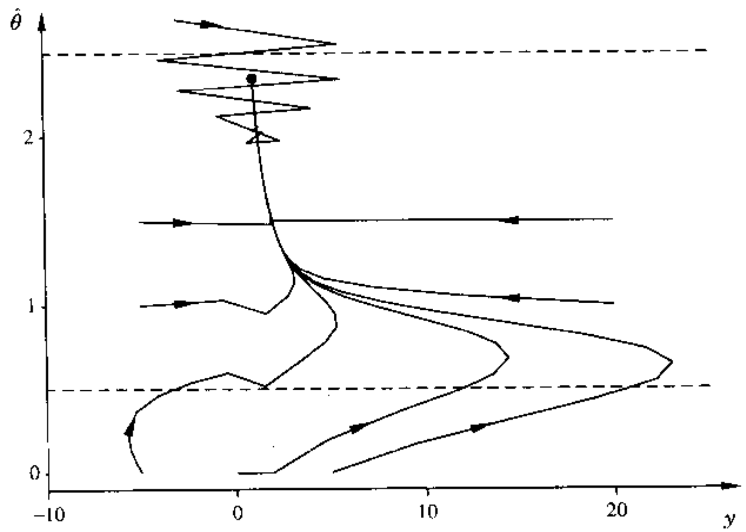

Figure 6.2 Phase portrait for the system in the stable case. Parameter values are $\alpha = 0.1$, $\gamma = 0.1$, $\theta_0 = 1.5$, $y_0 = 1$, and $a = 0.9$. The dashed lines indicate the interval $\theta_0 - 1 < \hat{\theta} < \theta_0 + 1$. The dot is the equilibrium point.

If parameter $\hat{\theta}$ is constant, the solution is stable if

$$\theta_0 - 1 < \hat{\theta} < \theta_0 + 1$$

and unstable otherwise. These bounds are shown as dashed lines in Fig. 6.2. If the parameter $\hat{\theta}$ is kept constant, y diverges monotonically at the lower bound and diverges in an oscillatory manner with period 2 at the upper bound. In reality, parameter $\hat{\theta}$ will of course change. The smaller the adaptation gain is, the smaller rate of change. With the numbers used in the simulation the bounds are 0.5 and 2.5. The behavior shown in Fig. 6.2 can thus be explained qualitatively. The solution approaches the curve (6.12) and then moves along this curve. The variable y appears to grow exponentially for $\hat{\theta} < 0.5$; it decays exponentially for $0.5 < \hat{\theta} < 1.5$ and decays in an oscillatory manner for $1.5 < \hat{\theta} < 2.5$. The variable grows in an oscillatory manner for $\hat{\theta} > 2.5$.

We now turn our attention to the equation for the parameter estimate. Introducing

$$\tilde{\theta} = \hat{\theta} - \theta_0$$

we find

$$\tilde{\theta}(t+1) = \left(1 - \gamma \frac{y^2(t)}{\alpha + y^2(t)}\right) \tilde{\theta}(t) + \gamma \frac{ay(t)}{\alpha + y^2(t)} \tag{6.13}$$

This equation implies that the signals y and $\tilde{\theta}$ cannot be unbounded because Eq. (6.13) is always stable when γ is sufficiently small. For large values of $y(t)$ the added term is small, and the solution will decay. It thus appears as though the equilibrium solution that is locally stable may also be globally stable in this case. A more precise discussion of this is given in Section 6.5.

Unstable Local Equilibria We now investigate what happens when parameters are such that the local equilibrium is unstable. We first observe that the instabilities may occur by violating any of the conditions given in Eqs. (6.11). Analyzing how the eigenvalues change with the parameters shows that the eigenvalue passes the unit circle with complex values if condition (i) is violated, through $z = -1$ if condition (ii) is violated and through $z = 1$ if condition (iii) is violated. We consider the situation in which the value of the adaptation gain is too large. Increasing the gain means that the solution will become unstable with period 2. Consider the case in which $\theta_0 = 1$, $\alpha = 0.1$, $y_0 = 1$, and $a = 0.9$. The equilibrium is $y = 1$ and $\theta = 1.9$. It follows from the stability criterion that the equilibrium is stable if $\gamma < 0.22$. With $\gamma = 0.5$ the linearized closed-loop system is unstable. Figure 6.3 shows a simulation of the system. The behavior of the system is typical for the case with unmodeled dynamics. The output y and the parameter estimate $\hat{\theta}$ appear to approach their equilibrium values. The equilibrium is unstable and a diverging oscillation with period 2 appears when y and $\hat{\theta}$ come sufficiently close to their equilibrium. The variables then oscillate with large excursions. When this happens, the modeling error becomes less significant, and the process output y and the parameter estimate $\hat{\theta}$ approach their equilibrium values. The process then repeats all over

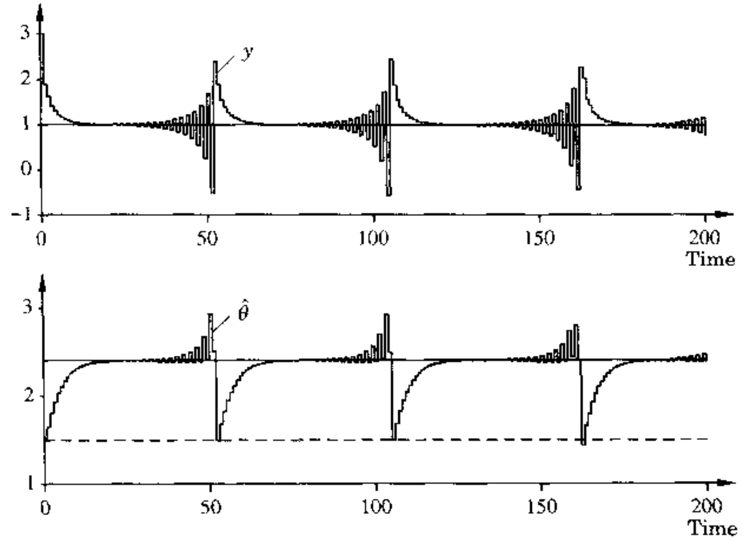

Figure 6.3 Simulation of a simple adaptive controller with unmodeled dynamics. The equilibrium values of y and $\hat{\theta}$ are indicated by solid straight lines. The true parameter value θ_0 is indicated by a dashed straight line.

again. The phenomenon that has been observed in many adaptive systems is called *bursting*.

The simulation shown in Fig. 6.3 represents a very complex behavior. Although essentially the same phenomenon repeats itself, the solution is *not* periodic. This is seen more clearly if the system is simulated for a longer time. Figure 6.4 shows a phase plane when the simulation time is extended to 10,000 time units. The solution is very irregular. There is, however, some pattern in the motion, as is indicated in the figure. For example, the state moves close to the curve given by Eq. (6.12) for part of the motion. The behavior shown is in fact an example of chaotic behavior. The pattern shown in Fig. 6.4 is called a *strange attractor*.

Structural Stability

Structural stability is an important concept in nonlinear dynamics. Intuitively, a system is structurally stable if small changes in the equations will not lead to drastic changes in the behavior of the system. A necessary condition for structural stability in the continuous-time case is that all equilibria are such that the linearized equations do not have eigenvalues whose real parts are zero. The equilibria are then said to be *hyperbolic*. Stability and structural

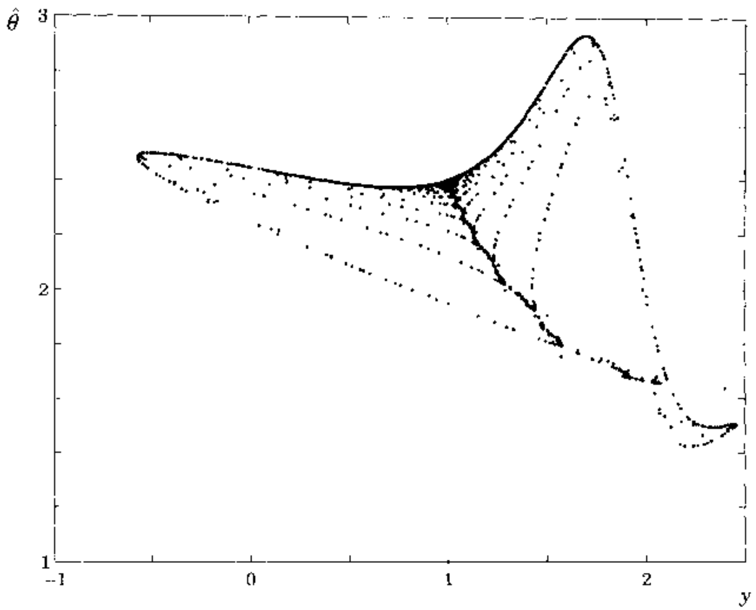

Figure 6.4 Phase plane plot corresponding to the case in Fig. 6.3 when over 10,000 time units are simulated.

stability in adaptive systems are closely related to persistency of excitation. We illustrate this by two examples.

EXAMPLE 6.1 Lack of excitation leads to instability

Consider the model-reference adaptive system shown in Fig. 5.14(b). Assume that the input signal is $u_c(t) = e^{-t}$. The system can then be described by the equations

$$\frac{de}{dt} = -e + k\tilde{\theta}u_c$$
$$\frac{d\tilde{\theta}}{dt} = -\gamma e u_c$$
$$\frac{du_c}{dt} = -u_c$$

where $\tilde{\theta} = \hat{\theta} - \theta_0$. The equilibrium is $e = \tilde{\theta} = u_c = 0$. Linearization around this point gives a linear system with the system matrix.

$$A = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

This matrix has the eigenvalues −1, 0, and −1, and the system is clearly not stable. □

EXAMPLE 6.2 Persistency of excitation gives structural stability

Consider the same system as in Example 6.1, but assume now that the command signal is a step, that is, $u_c(t) = 1$. The system is then described by the equations

$$\frac{de}{dt} = -e + k\tilde{\theta}$$
$$\frac{d\tilde{\theta}}{dt} = -\gamma e$$

The equilibrium is $e = \tilde{\theta} = 0$. Linearization around this fixed point gives a linear system with the system matrix.

$$A = \begin{pmatrix} -1 & k \\ -\gamma & 0 \end{pmatrix}$$

This matrix has the characteristic polynomial

$$s^2 + s + \gamma k$$

and the equilibrium is thus stable if γk is positive. □

Figure 2.10 in Chapter 2, which illustrates a case of identification under closed-loop conditions, is a typical example of structural instability. Additional examples are given in Section 6.9.

6.3 ADAPTATION OF A FEEDFORWARD GAIN

The special case of adaptation of a feedforward gain has been discussed many times because of its simplicity. Let us therefore consider the structure of the equations in this case too. For the system in Fig. 5.14 we get

$$\begin{aligned} \frac{d\xi}{dt} &= A\xi + B\theta u_c \\ e &= C\xi - y_m \\ \varphi &= \begin{cases} -y_m & \text{MIT rule} \\ -u_c & \text{Lyapunov rule} \end{cases} \end{aligned} \tag{6.14}$$

where A, B, and C are matrices that give a realization of the transfer function $kG(s)$. Notice that in this case the matrices A, B, and C, the regression vector φ, and the error e do not depend on the controller parameters explicitly. Furthermore, the parameter is updated as

$$\frac{d\hat{\theta}}{dt} = \gamma\varphi e(\xi) \tag{6.15}$$

for a gradient scheme. If u_c is a function of time, then y_m is also a function of time, and Eqs. (6.14) and (6.15) are simply time-varying linear differential equations. Such equations can have a complex behavior. We illustrate this by an example before proceeding.

EXAMPLE 6.3 Adaptation of a feedforward gain

In Example 5.1 we derived an adaptation law for adjusting the feedforward gain by using the MIT rule. The behavior of the system was illustrated in Fig. 5.3. The system is described by

$$\frac{dy}{dt} = k\hat{\theta}(t)u_c(t) - y(t)$$

and the parameter adjustment rule is

$$\frac{d\hat{\theta}}{dt} = -\gamma y_m(t)e(t) = -\gamma y_m(t)\left(y(t) - y_m(t)\right)$$

Since the signal y_m can be computed from the command signal u_c, both u_c and y_m can thus be regarded as known time-varying signals. The adaptive system is described by the equation

$$\frac{d}{dt}\begin{pmatrix}\hat{\theta}\\ y\end{pmatrix} = \begin{pmatrix}0 & -\gamma y_m(t)\\ ku_c(t) & -1\end{pmatrix}\begin{pmatrix}\hat{\theta}\\ y\end{pmatrix} + \begin{pmatrix}\gamma y_m^2(t)\\ 0\end{pmatrix} \tag{6.16}$$

The system can thus be described by a time-varying linear differential equation of second order. In Fig. 6.5 we show three simulations for the case in which $G(s) = 1/(s+1)$, $k = k_0 = 1$, and $\gamma = 11$. The reference signal is sinusoidal in all cases. The frequency is $\omega = 1$ in the first case, $\omega = 2$ in the second, and $\omega = 3$ in the third. The controller parameter converges to the correct value for $\omega = 1$ and $\omega = 3$, but it diverges for $\omega = 2$. We thus have a situation in which the system is stable for one input but unstable for another. The system is stable for low frequencies of the input signal. As the frequency increases, it becomes unstable. It becomes stable again as the frequency is increased further. This pattern repeats itself as the frequency is increased further. □

Example 6.3 shows that the system has quite a complex behavior that cannot be explained by the intuitive argument of the previous section. To understand what is happening, we analyze the equations describing the system. Equation (6.16) can be written as

$$\frac{dx}{dt} = A(t)x + B(t) \tag{6.17}$$

This is a linear system with time-varying parameters. In the particular case in which the input u_c is periodic and we connect the adaptation when model output y_m has also become periodic, the system is also periodic. For such systems there is a well-developed theory that can be used to understand the behavior of the system.

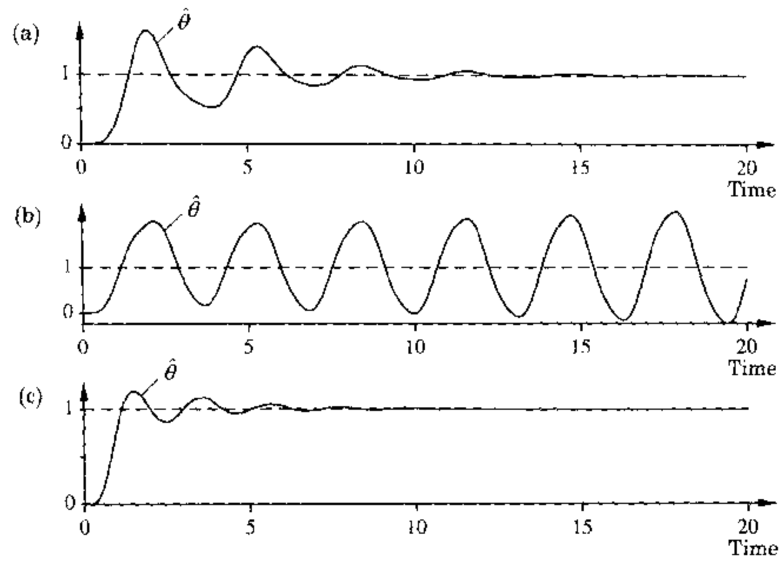

Figure 6.5 Behavior of the controller gain for an MRAS using the MIT rule. The input signal is a unit amplitude sinusoidal with frequency (a) 1; (b) 2; and (c) 3 rad/s. The system has the transfer function $G(s) = 1/(s+1)$, the parameters are $k = k_0 = 1$, and the adaptation gain is $\gamma = 11$. The dashed lines indicate the correct values of the gain.

Floquet Theory

To investigate the stability properties of (6.17), we consider the homogeneous part, when $A(t)$ is periodic with period τ and continuous for all t. The solution is given by

$$x(t) = \Phi(t, t_0)x(t_0)$$

where $\Phi(t, t_0)$ satisfies the linear matrix differential equation

$$\frac{d\Phi}{dt} = A(t)\Phi \tag{6.18}$$

Since $A(t)$ is periodic with period τ, it follows that $A(t+\tau) = A(t)$. This implies that if $\Phi(t)$ is a solution, then $\Phi(t+\tau)$ is also a solution. Since the two solutions to Eq. (6.18) differ only in their initial conditions it follows that

$$\Phi(t+\tau) = \Phi(t)W \tag{6.19}$$

where W is a nonsingular constant matrix. Since the matrix $\Phi(t)$ is nonsingular for all t, it follows that W is also nonsingular. By repeated use of this equation we find that

$$\Phi(t+n\tau) = \Phi(t)W^n$$

where $t < \tau$. We thus obtain the following result.

THEOREM 6.1 Stability of linear periodic system

The periodic differential equation (6.17) is stable if and only if all eigenvalues of the matrix W have magnitudes less than 1. □

This result is actually all we need for stability analysis. We can, however, also obtain a slightly more general result. Notice that we can compute W simply by integrating the differential equation over one period with the initial condition equal to the identity matrix.

THEOREM 6.2 Solution of periodic systems

The solution to the matrix differential equation (6.18) has the form

$$\Phi(t) = D(t)e^{Ct}$$

where C is a constant matrix and D is periodic with period τ.

Proof: Since the matrix W in Eq. (6.19) is nonsingular, there exists a matrix C such that

$$W = e^{C\tau} \tag{6.20}$$

Introduce the matrix function $D(t)$ defined by

$$D(t) = \Phi(t)e^{-Ct}$$

Then

$$D(t+\tau) = \Phi(t+\tau)e^{-C(t+\tau)} = \Phi(t)We^{-C\tau}e^{-Ct} = D(t)$$

and the theorem is proven. □

Remark. From Eq. (6.20) we see that the differential equation (6.18) is stable if the matrix C has all its eigenvalues in the left half-plane, which means that the matrix W should have all its eigenvalues inside the unit disc. Stability can thus be determined by numerical integration over one period. □

We now show how the results can be used to investigate the stability of the system in Example 6.3.

EXAMPLE 6.4 Parametric excitation

Consider the system in Example 6.3. Let the command signal be $u_c(t) = \sin \omega t$. After a transient the model output becomes

$$y_m(t) = \frac{1}{\sqrt{1+\omega^2}} \sin(\omega t - \arctan(\omega))$$

To determine the stability of Eq. (6.16), we compute W by integrating Eq. (6.18) over one period, that is, $\tau = 2\pi/\omega$, with the initial condition $\Phi(0) = I$. Then

from Eq. (6.19) we get $W = \Phi(\tau)$. Choosing $\omega = 2$ and integrating to $\tau = \pi$ give

$$\Phi(\tau) = \begin{pmatrix} 0.4373 & 0.7283 \\ 0.2389 & 0.4967 \end{pmatrix}$$

with eigenvalues 0.049 and 0.885 for $\gamma = 10$ and

$$\Phi(\tau) = \begin{pmatrix} 0.5609 & 0.9960 \\ 0.2642 & 0.5463 \end{pmatrix}$$

with eigenvalues 0.041 and 1.067 for $\gamma = 11$. It can thus be concluded that the adaptive system will be stable for $\gamma = 10$ but unstable for $\gamma = 11$.

This calculation can be repeated for many frequencies and many values of the adaptation gain to determine the values of ω and γ for which the system is stable. The result of such a calculation is shown in Fig. 6.6. Notice in particular that Fig. 6.6 explains the behavior observed in the numerical experiment in Example 6.3, in which the system goes through a region of instability as the frequency of the input signal increases. Notice, however, that the system is stable for low adaptation gains. □

Example 6.3 indicates that even very simple adaptive systems can exhibit complex behavior. The mechanism of periodic excitation can also give rise to instabilities in more complex adaptive systems. The analysis can be made in the same way as for the simple example, but the details are much more complicated. The behavior is typically associated with periodic excitation and comparatively high adaptation gains. The phenomenon illustrated in Fig. 6.5

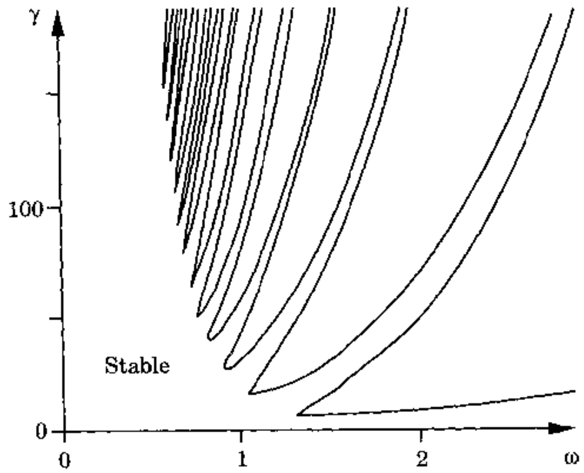

Figure 6.6 Stability region for adjustment of a feedforward gain with the MIT rule.

is an example of *parametric excitation*, that is, a system can be made unstable by changing its parameters periodically. A classical example is the Mathieu equation:

$$\frac{d^2y}{dt^2} + \alpha\frac{dy}{dt} + (\beta + \gamma \cos \omega t)y = 0$$

For $\alpha = 0$ this equation describes a pendulum whose pivot point is oscillating vertically. It is well known that the normal equilibrium, with the pendulum hanging down, can be made unstable by a proper choice of the parameters.

EXAMPLE 6.5 Lyapunov redesign

In Example 6.3 we found that the MIT rule could give instabilities for large adaptation gains. Under the strong assumption that the transfer function of the process is strictly positive real, however, the control law derived from stability theory is stable for all values of the adaptation gain. We illustrate this in the simulation in Fig. 6.7, in which the Lyapunov rule is applied to the system in Example 6.3. Compare with Fig. 6.5. □

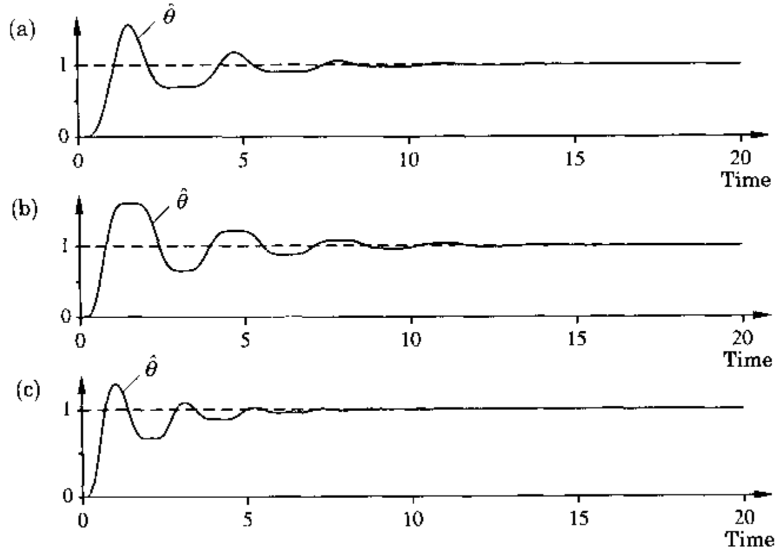

Figure 6.7 Behavior of the controller gain for an adaptive system based on Lyapunov stability theory when the input signal is a unit amplitude sinusoidal with frequency (a) 1; (b) 2; and (c) 3 rad/s. The system has the transfer function $G(s) = 1/(s+1)$, the parameters are $k = k_0 = 1$, and the adaptation gain is $\gamma = 11$. The dashed lines indicate the correct values of the gain.

Summary

A discrete-time system and the feedforward gain example have been discussed in this section. The examples show that adaptive controllers can have rather strange properties. The phenomena could be explained by using simple mathematics, but there will be difficulties in the general cases. It is therefore appropriate to consider some simplified situations in the coming sections. First, indirect and direct self-tuning regulators are discussed under idealized assumptions. Second, the adaptive control problem is divided into two parts with different time scales, and averaging techniques are used to analyze properties of the closed-loop systems.

6.4 ANALYSIS OF INDIRECT DISCRETE-TIME SELF-TUNERS

In this section we analyze the properties of indirect discrete-time self-tuners of the type illustrated by the block diagram in Fig. 1.19. Since such controllers contain a recursive parameter estimator and a control design calculation, it is natural to investigate these subsystems separately. Since identification is performed in closed loop, there may also be undesirable effects due to interaction of control and identification. We start by investigating the properties of the recursive parameter estimator. Second, the design calculations must be considered. It is particularly important to understand when the design calculations are poorly conditioned so that small changes in process parameter estimates may cause large changes in the controller parameters.

It would be highly desirable to determine whether the adaptive system can track parameters of a time-varying system. This is a very difficult problem, and we therefore limit the analysis to the case in which the real system has constant parameters. This can be considered as a first test of an adaptive algorithm. To carry out the analysis, we also assume that the real system is described by models that are compatible with the models used for parameter estimation. In this case it makes sense to talk about the "true parameters." In reality, however, we also have to deal with the fact that the models that we use are approximations. This is called the nonideal case. This problem is discussed later in Section 6.9.

Properties of Recursive Estimators

To investigate recursive estimators, it is necessary to make some assumptions on how the data was generated. In this section we make the assumption that the data is generated by a model having the same structure as the the model used in the estimation. It is also necessary to specify the nature of the disturbances—for example, whether they are deterministic or stochastic. We also find that it is important for there to be sufficient excitation and that identification under closed-loop conditions may cause difficulties.

The deterministic case, in which data is generated from a system that is compatible with the model used in the estimator, is particularly simple. In this case it is possible to derive general properties of the estimators.

Projection or Gradient Algorithms

The properties of the projection or gradient algorithm are now investigated in the ideal case in which data is generated by the model

$$y(t) = \varphi^T(t)\theta^0 \tag{6.21}$$

We have the following result.

THEOREM 6.3 Projection algorithm properties

Let the estimator

$$\begin{aligned}\hat\theta(t) &= \hat\theta(t-1) + \frac{\gamma\varphi(t)}{\alpha + \varphi^T(t)\varphi(t)} e(t)\\ e(t) &= y(t) - \varphi^T(t)\hat\theta(t-1) = \varphi^T(t)\left(\theta^0 - \hat\theta(t-1)\right)\end{aligned} \tag{6.22}$$

with $\alpha \geq 0$ and $0 < \gamma < 2$, be applied to data generated by Eq. (6.21). It then follows that

(i) $\quad \|\hat\theta(t) - \theta^0\| \leq \|\hat\theta(t-1) - \theta^0\| \leq \|\hat\theta(0) - \theta^0\| \qquad t \geq 1$

(ii) $\quad \displaystyle\lim_{t\to\infty} \frac{e(t)}{\sqrt{\alpha + \varphi^T(t)\varphi(t)}} = 0$

(iii) $\quad \displaystyle\lim_{t\to\infty} \|\hat\theta(t) - \hat\theta(t-k)\| = 0 \qquad$ for any finite k

Proof: Introduce $\tilde\theta(t) = \hat\theta(t) - \theta^0$ and

$$V(t) = \tilde\theta^T(t)\tilde\theta(t) = \|\tilde\theta(t)\|^2$$

It follows that

$$e(t) = \varphi^T(t)\theta^0 - \varphi^T(t)\hat\theta(t-1) = -\varphi^T(t)\tilde\theta(t-1)$$

Subtracting θ^0 from both sides of the parameter equation in Eqs. (6.22) and taking the norm, we get

$$\begin{aligned}V(t) - V(t-1) &= 2\frac{\gamma\varphi^T(t)\tilde\theta(t-1)e(t)}{\alpha + \varphi^T(t)\varphi(t)} + \frac{\gamma^2\varphi^T(t)\varphi(t)e^2(t)}{(\alpha + \varphi^T(t)\varphi(t))^2}\\ &= \chi(t)\frac{\gamma e^2(t)}{\alpha + \varphi^T(t)\varphi(t)}\end{aligned}$$

where

$$\chi(t) = -2 + \frac{\gamma\varphi^T(t)\varphi(t)}{\alpha + \varphi^T(t)\varphi(t)} \leq -\delta < 0$$

and the inequality follows from $\alpha \geq 0$ and $0 < \gamma < 2$. Property (i) has thus been established. It follows from the preceding equation that

$$V(t) = V(0) + \sum_{k=1}^{t}\chi(k)\frac{\gamma e^2(k)}{\alpha + \varphi^T(k)\varphi(k)}$$

Hence

$$\sum_{k=1}^{t}\frac{\gamma e^2(k)}{\alpha + \varphi^T(k)\varphi(k)} \leq \frac{1}{\delta}\left(V(0) - V(t)\right)$$

Since $0 \leq V(t) \leq V(0)$, it follows that the normalized error

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

is in l_2, that is, squared summable, and thus property (ii) follows. From Eqs. (6.22),

$$\begin{aligned}\|\hat\theta(t) - \hat\theta(t-1)\|^2 &= \frac{\gamma^2\varphi^T(t)\varphi(t)e^2(t)}{(\alpha + \varphi^T(t)\varphi(t))^2}\\ &= \frac{\gamma^2 e^2(t)}{\alpha + \varphi^T(t)\varphi(t)}\left(1 - \frac{\alpha}{\alpha + \varphi^T(t)\varphi(t)}\right)\end{aligned}$$

It follows from property (ii) that the right-hand side of the preceding equation goes to zero as $t \to \infty$ if $\alpha > 0$. Hence

$$\begin{aligned}\|\hat\theta(t) - \hat\theta(t-k)\|^2 &= \left\|\sum_{i=1}^{k}\hat\theta(t-i+1) - \hat\theta(t-i)\right\|^2\\ &\leq \sum_{i=1}^{k}\|\hat\theta(t-i+1) - \hat\theta(t-i)\|^2\end{aligned}$$

where the right-hand side goes to zero as $t \to \infty$ for finite k. □

Remark 1. For $\gamma = 1$ and $\alpha = 0$ the algorithm reduces to Kaczmarz's projection algorithm.

Remark 2. Notice that the result does *not* imply that the estimates $\hat\theta(t)$ converge.

Remark 3. The function $V(t)$ can be interpreted as a discrete-time Lyapunov function. □

Theorem 6.3 is useful because it gives some properties of the estimator that are valid no matter how the regressors $\varphi(t)$ are generated. Additional conditions are required to guarantee that the estimates converge. The theorem will also be useful to prove convergence of the indirect adaptive schemes.

Parameter Convergence of Gradient Algorithms

We now give conditions for the estimates to converge to the true parameter values. Notice that to pose such a problem, it is necessary to assume that data is generated by a model that is compatible with the model used to formulate the estimate. Parameter convergence is closely related to system identification. The properties of identifiability and persistency of excitation play an essential role. The convergence rate depends on the algorithm used and the amount of excitation. We first consider the gradient algorithm, which is simpler than the least-squares algorithm, although it converges at a considerably slower rate. A typical projection or gradient algorithm is given by Eqs. (6.22), where $\alpha \geq 0$ and $0 < \gamma < 2$. The estimation error is given by

$$\tilde{\theta}(t) = \hat{\theta}(t) - \theta^0 = A(t-1)\tilde{\theta}(t-1) \tag{6.23}$$

where

$$A(t-1) = I - \frac{\gamma\varphi(t)\varphi^T(t)}{\alpha + \varphi^T(t)\varphi(t)}$$

The problem of analyzing convergence rates is thus equivalent to analyzing the stability of Eq. (6.23). Notice that

$$A(t-1)\varphi(t) = \left(I - \frac{\gamma\varphi(t)\varphi^T(t)}{\alpha + \varphi^T(t)\varphi(t)}\right)\varphi(t) = \varphi(t)\left(1 - \frac{\gamma\varphi^T(t)\varphi(t)}{\alpha + \varphi^T(t)\varphi(t)}\right)$$

The second factor on the right-hand side is a scalar. This implies that the vector $\varphi(t)$ is an eigenvector to $A(t-1)$ with an eigenvalue that is less than 1. The eigenvalue is zero for $\gamma = 1$ and $\alpha = 0$. The following lemma is useful to analyze Eq. (6.23).

LEMMA 6.1 Stability of a time-varying system

Consider the time-varying system

$$\begin{aligned} x(t+1) &= A(t)x(t) \\ y(t) &= C(t)x(t) \end{aligned} \tag{6.24}$$

Assume that there exists a symmetric matrix $P(t) > 0$ such that

$$A^T(t)P(t+1)A(t) - P(t) = -C^T(t)C(t) \tag{6.25}$$

Then Eqs. (6.24) are stable. Moreover, if the system is uniformly completely observable, that is, if there exist $\beta_1 > 0$, $\beta_2 > 0$, and $N > 1$ such that

$$0 < \beta_1 I \leq \sum_{k=t}^{t+N-1} \Phi^T(k,t)C^T(k)C(k)\Phi(k,t) \leq \beta_2 I < \infty$$

for all t and where $\Phi(k,t)$ is the fundamental matrix, then Eqs. (6.24) are also exponentially stable.

Proof: Introduce the function

$$V(t) = x^T(t)P(t)x(t)$$

Hence

$$\begin{aligned} V(t+1) - V(t) &= x^T(t)A^T(t)P(t+1)A(t)x(t) - x^T(t)P(t)x(t) \\ &= -x^T(t)C^T(t)C(t)x(t) \leq 0 \end{aligned}$$

The function V can be considered a Lyapunov function for a discrete-time system. To prove stability for a discrete-time system using Lyapunov theory, we have to show that the difference

$$\Delta V(t) = V(t+1) - V(t) \leq 0$$

and that the matrix $P(t)$ is positive definite. Iterating the system equations N steps gives

$$\begin{aligned} V(t+N) - V(t) &= -\sum_{k=t}^{t-N-1} x^T(k)C^T(k)C(k)x(k) \\ &= -x^T(t)\left(\sum_{k=t}^{t+N-1} \Phi^T(k,t)C^T(k)C(k)\Phi(k,t)\right)x(t) \\ &\leq -\beta_1 x^T(t)x(t) \leq -\frac{\beta_1}{\lambda_{\max}P(t)}V(t) \end{aligned}$$

where $\lambda_{max}(Pt)$ is the largest eigenvalue of $P(t)$. Hence

$$V(t+N) \leq \left(1 - \frac{\beta_1}{\lambda_{\max}P(t)}\right)V(t) = \beta_3 V(t)$$

From Eq. (6.25) it follows that

$$\begin{aligned} P(t) &= C^T(t)C(t) + A^T(t)P(t+1)A(t) \\ &= C^T(t)C(t) \\ &\quad + A^T(t)\left(C^T(t+1)C(t+1) + A^T(t+1)P(t+2)A(t+1)\right)A(t) \\ &\ \vdots \\ &= \sum_{k=t}^{\infty} \Phi^T(k,t)C^T(k)C(k)\Phi(k,t) \\ &> \sum_{k=t}^{t+N-1} \Phi^T(k,t)C^T(k)C(k)\Phi(k,t) \geq \beta_1 I \end{aligned}$$

This shows that $\lambda_{max}(P(t)) > \beta_1$ and $\beta_3 < 1$, which implies that $V(t)$ goes to zero exponentially. Furthermore,

$$P(t+N) = P(t) - \sum_{k=t}^{t+N-1} \Phi^T(k,t)C^T(k)C(k)\Phi(k,t) \leq \beta_3 P(t)$$

or

$$P(t) \leq \frac{1}{1-\beta_3} \sum_{k=t}^{t+N-1} \Phi^T(k,t)C^T(k)C(k)\Phi(k,t) \leq \frac{\beta_2}{1-\beta_3} I$$

The matrix $P(t)$ is thus bounded from above and below. Since $V(t)$ goes to zero exponentially and $P(t)$ is bounded, it follows that the system (6.24) is exponentially stable. □

Applying this lemma to Eq. (6.23), we get the following theorem.

THEOREM 6.4 Exponential stability

The difference equation (Eq. 6.23) is globally exponentially stable if there exist positive constants β_1, β_2, and N such that for all t,

$$0 < \beta_1 I \leq \sum_{k=t}^{t+N-1} \varphi(k)\varphi^T(k) \leq \beta_2 I < \infty \tag{6.26}$$

Proof: Choose $P = I$ and

$$C(t) = \frac{\sqrt{\gamma(2\alpha + (2-\gamma)\varphi^T\varphi)}}{\alpha + \varphi^T\varphi}\varphi^T$$

where the argument t of φ is suppressed. A straightforward calculation shows that Eq. (6.25) is satisfied, so the system is stable. To prove exponential stability, first observe that uniform observability of $(A(k), C(k))$ is equivalent to uniform observability of $((A(k) - B(k)C(k)), C(k))$. Choosing

$$B(k) = -\frac{\gamma}{\sqrt{(\gamma(2\alpha + (2-\gamma)\varphi^T\varphi)}}\varphi$$

we find that $A(k) - B(k)C(k) = I$, and uniform asymptotic stability then corresponds to Eq. (6.26). □

Notice that Eq. (6.26) is closely related to persistent excitation. (Compare with Definition 2.1.) It is thus found that exponential convergence of the gradient algorithm is closely connected to whether the input signal to the system is persistently exciting of sufficiently high order.

It should be pointed out that condition (6.26) is a persistent excitation condition for the regressors, not the external reference signal for the system. The excitation can be provided by the command signals and by the disturbances acting on the process. Notice, however, that excitation may be lost by feedback, which can introduce relations between the variables appearing in the regression vector. We discuss this later in this section.

Recursive Least Squares

Parameter convergence for recursive least squares is first discussed for the simple model (6.21), which is linear in the parameters and for which are no disturbances. Let the parameter vector have n elements. The parameters can be calculated exactly from n data points, provided that the vectors $\varphi(1), \ldots, \varphi(n)$ are linearly independent. The least-squares estimate is given by

$$\begin{aligned}\hat{\theta}(n) &= \left(\sum_{k=1}^{n} \varphi(k)\varphi^T(k)\right)^{-1} \sum_{k=1}^{n} \varphi(k)y(k) \\ &= \left(\sum_{k=1}^{n} \varphi(k)\varphi^T(k)\right)^{-1} \sum_{k=1}^{n} \varphi(k)\varphi^T(t)\theta^0 = \theta^0\end{aligned} \tag{6.27}$$

The correct state is obtained in n steps. If the estimate is instead calculated by recursive least squares, the following estimate is obtained:

$$\hat{\theta}(t) = \left(P^{-1}(0) + \sum_{k=1}^{t} \varphi(k)\varphi^T(k)\right)^{-1} \left(\sum_{k=1}^{t} \varphi(k)y(k) + P^{-1}(0)\hat{\theta}(0)\right) \tag{6.28}$$

where $\hat{\theta}(0)$ is the initial estimate and $P(0)$ is the initial covariance of the estimator. By making $P(0)$ positive definite but arbitrarily large, the result from the recursive estimation can be made arbitrarily close to the true value.

From this analysis we obtain the following result.

THEOREM 6.5 Property of RLS

Let the recursive least squares be applied to data generated by Eq. (6.21). Let $P(0)$ be positive definite and let $\hat{\theta}(0)$ be bounded. Assume that

$$\beta(t)I \leq \sum_{k=1}^{t} \varphi(k)\varphi^T(k)$$

where $\beta(t)$ goes to infinity. Then the estimate converges to θ^0. □

This discussion shows that in the deterministic case it is possible to obtain parameter estimators that converge in a finite number of steps. The key assumption is that the regressors are linearly independent, so $\sum \varphi(k)\varphi^T(k)$ is of full rank. When the parameters are changing, a least-squares estimator, in which the covariance matrix P is regularly reset to αI, is a good implementation. This procedure is called *covariance resetting*. To obtain an estimate that reacts rapidly to parameter changes, it is also possible to have several estimators in parallel, which are reset sequentially.

Results similar to Theorem 6.3 can also be established for the least-squares algorithm and several of its variants. The key is to replace function $V(t)$ in Theorem 6.3 by

$$V(t) = \tilde{\theta}^T(t)P^{-1}(t)\tilde{\theta}(t)$$

and add assumptions that guarantee that the eigenvalues of P stay bounded. One way to do this is to use the constant trace algorithm (see Section 11.5).

So far, only the general model (6.21) has been discussed. The properties of estimates of parameters of discrete-time transfer functions will now be considered. The uniqueness of the estimates is first explored. For this purpose we assume that the data is actually generated by

$$A^0(q)y(t) = B^0(q)u(t) + e(t+n) \tag{6.29}$$

where A^0 and B^0 are relatively prime. If $e = 0$, $\deg A > \deg A^0$, and $\deg B > \deg B^0$, it follows from Theorem 2.1 that the estimate is not unique because the columns of the matrix Φ are linearly dependent. Theorem 2.10 gives conditions for uniqueness of the least-squares estimate.

The Stochastic Case

Consider the model

$$y(t) = \varphi^T(t)\theta^0 + e(t)$$

where $\{e(t)\}$ is a sequence of independent Gaussian $(0, \sigma)$ random variables. The least-squares estimator is given by Eq. (6.28). The covariance of the estimate for large t is (see Theorem 2.2)

$$P(t) = \sigma^2 \left(\sum_{k=1}^{t} \varphi(k)\varphi^T(k) \right)^{-1}$$

By taking the covariance of the estimate as a measure of the rate of convergence, it is found that under uniform persistent excitation the matrix P converges at the rate $1/t$. This implies that the estimates converge at the rate $1/\sqrt{t}$.

THEOREM 6.6 Convergence of RLS

Let the least-squares method for estimating parameters of a transfer function be applied to data generated by the model of Eq. (6.29) where $\{e(t)\}$ is a sequence of uncorrelated random variables with zero mean and variance σ^2. Assume that the estimated model has the same structure as the process generating the data, that is, the ideal case. Further assume that the input signal is persistently exciting of order $\deg A + \deg B + 1$. Then

(i) $\hat{\theta}(t) \to \theta^0$ in the mean square as $t \to \infty$

(ii) $\operatorname{var}(\hat{\theta} - \theta^0) \approx \dfrac{\sigma^2}{t} \left(\lim_{t\to\infty} \dfrac{1}{t} \Phi^T \Phi \right)^{-1}$

□

Remark 1. The estimates do not converge to the true parameters when $e(t)$ is correlated with $e(s)$ for $t \neq s$.

Remark 2. Theorem 6.6 gives the convergence rate for the parameter error in the ideal case. More complex behavior can be obtained when the different components of the regression vector have different convergence rates (see Example 2.11). □

Unmodeled Dynamics

So far, it has been assumed that the true process is compatible with the model used in parameter estimation. It frequently happens that the true process is more complex than the estimated model. This is often referred to as *unmodeled dynamics*. The problem is complex, and a careful analysis is lengthy; roughly speaking, the parameters will converge to a value that minimizes the least-squares criterion:

$$V(\theta) = \frac{1}{T} \sum_{0}^{T} (Ay_f(t) - Bu_f(t)) \tag{6.30}$$

where y_f and u_f are the filtered process input and output, that is,

$$y_f = H_f y$$
$$u_f = H_f u$$

and the parameter θ represents the coefficients of the polynomials A and B. The minimum exists under certain regularity conditions, and the minimizing θ is unique under the condition of persistency of excitation. The minimizing value will depend on the data filter H_f and the spectrum of the reference signal and the disturbances.

Identification in Closed Loop

When discussing parameter estimation in Chapter 2, we observed that identifiability could be lost if the input was generated by feedback from the output. The reason is that the feedback introduces dependencies in the regression vector. (Compare with Example 2.10.) Since this is very important for the behavior of direct adaptive controllers, we will investigate the problem in a little more detail. In this analysis we will consider what happens when we perform system identification to data generated by feedback. Consider a process described by

$$Ay(t) = Bu(t) + v(t) \tag{6.31}$$

with the controller

$$Ru = Tu_c - Sy$$

where polynomials R, S, and T have constant parameters. The closed-loop system is given by

$$y = \frac{BT}{AR+BS}u_c + \frac{R}{AR+BS}v$$
$$u = -\frac{AT}{AR+BS}u_c + \frac{S}{AR+BS}v$$

With a system identification experiment it is possible to determine the transfer functions

$$G_1 = \frac{BT}{AR+BS} \qquad G_2 = -\frac{AT}{AR+BS}$$
$$G_3 = \frac{R}{AR+BS} \qquad G_4 = \frac{S}{AR+BS}$$

that appear in these equations. There are no problems with identifiability if the input signal u_c is persistently exciting of sufficiently high order because the polynomials A and B are then readily determined from G_1 and G_2. However, if the command signal is zero and all excitation comes from the disturbance, we can determine only the polynomial

$$A_c = AR + BS \tag{6.32}$$

To achieve identifiability, it must also be required that the signal v be persistently exciting of sufficiently high order. The question of identifiability of polynomials A and B then becomes a problem of uniquely determining A and B from Eq. (6.32) when polynomials R and S are known. If A_0 and B_0 are solutions, the general solution is

$$A = A_0 + QS \quad B = B_0 - QR$$

where Q is an arbitrary polynomial. When the model structure is specified, the highest degree of A is also given. The solution is thus unique only if polynomials R and S have sufficiently high degree. To achieve identifiability in closed loop, it is therefore important that the controller be of sufficiently high order. It is natural to assume that R and S have the same degree. Identifiability is then obtained if

$$\deg R = \deg S \geq \deg A \tag{6.33}$$

In Example 3.2, in which $\deg A = 2$, $\deg B = 1$, and $\deg R = \deg S = 1$, we do not have identifiability under closed loop with $u_c = 0$. However, if it is required that the controller has integral action as in Example 3.10, we have $\deg R = \deg S = 2$, and the condition (6.33) holds. To achieve identifiability, it must, of course, be required that the disturbance be persistently exciting.

Also observe that if a pole placement design is used, all models that are estimated will give the correct closed-loop characteristic polynomial.

Design Calculations

The design calculations are an important part of indirect adaptive systems. Theoretically, the design procedure is represented by the function χ, which maps process parameters $\hat{\theta}$ to controller parameters ϑ. The properties of χ will, of course, depend on the parameterization of the model and the design procedure chosen. The function can often be quite complicated. It is important that the map gives unique controller parameters and that there are no singularities in the map.We discuss the properties of the map in some simple cases.

Consider the process model

$$Ay = Bu \tag{6.34}$$

where it is assumed that A has degree n and B has degree $n-1$. The model thus has $2n$ parameters. If pole placement design is used, the controller parameters are given by

$$AR + BS = A_o A_m \tag{6.35}$$

where R and S have the same degree m as the observer polynomial A_o. The minimum-degree solution corresponds to $m = n - 1$, but an observer of higher order is often preferable to improve the robustness of the system. Without loss of generality, R can be monic. The controller then has $2m + 1$ parameters. The function χ is thus a map from R^{2n} to R^{2m+1}, where $m \geq n - 1$. Since Eq. (6.35) becomes singular when polynomials A and B have a common factor, it follows that the map χ has singularities. The problem with design singularities is illustrated by an example.

EXAMPLE 6.6 Singularities for pole placement design

Consider the model of Eq. (6.34) with

$$A(q) = q^2 + a_1 q + a_2$$
$$B(q) = b_0 q + b_1$$

In Example 3.2 a controller was designed for

$$A_m(q) = q^2 + a_{m1} q + a_{m2}$$
$$A_o(q) = q + a_o$$

In this case the controller and process parameters are

$$\vartheta = \begin{pmatrix} r_1 & s_0 & s_1 \end{pmatrix} \qquad \theta = \begin{pmatrix} a_1 & a_2 & b_0 & b_1 \end{pmatrix}$$

and the map $\chi : R^4 \to R^3$ is given by

$$
\begin{aligned}
r_1 &= \frac{a_o a_{m2} b_0^2 + (a_2 - a_{m2} - a_o a_{m1}) b_0 b_1 + (a_o + a_{m1} - a_1) b_1^2}{b_1^2 - a_1 b_0 b_1 + a_2 b_0^2} \\
s_0 &= \frac{b_1 (a_o a_{m1} - a_2 - a_{m1} a_1 + a_1^2 + a_{m2} - a_1 a_o)}{b_1^2 - a_1 b_0 b_1 + a_2 b_0^2} \\
&\quad + \frac{b_0 (a_{m1} a_2 - a_1 a_2 - a_o a_{m2} + a_o a_2)}{b_1^2 - a_1 b_0 b_1 + a_2 b_0^2} \\
s_1 &= \frac{b_1 (a_1 a_2 - a_{m1} a_2 + a_o a_{m2} - a_o a_2)}{b_1^2 - a_1 b_0 b_1 + a_2 b_0^2} \\
&\quad + \frac{b_0 (a_2 a_{m2} - a_2^2 - a_o a_{m2} a_1 + a_o a_2 a_{m1})}{b_1^2 - a_1 b_0 b_1 + a_2 b_0^2}
\end{aligned}
\qquad (6.36)
$$

The map χ is singular when the denominator in Eqs. (6.36) vanishes, that is, when

$$b_1^2 - a_1 b_0 b_1 + a_2 b_0^2 = 0$$

□

Singularities of the type in Example 6.6 will appear for practically all design methods. Since the singularities are algebraic surfaces, the parameter estimates must pass them if the algorithms are not initialized properly. There are several ways to avoid the difficulties. One possibility is to test for common factors and to cancel them if they appear, but such a procedure will require test quantities. It will also make χ discontinuous, which creates difficulties in the analysis. Another and better solution is to find design techniques such that the mapping χ is smooth. This is an open research problem, which so far has received little attention.

The following example illustrates what happens if no precautions are taken with cancellations.

EXAMPLE 6.7 Indirect adaptive system with design singularities

Consider the system in Example 6.6, and let the controller be an indirect adaptive system that is based on estimation of the parameters of the model. The desired dynamics A_m are chosen to correspond to a second-order system with $\omega = 1.5$ and $\zeta = 0.707$. The observer polynomial is chosen to be $A_o = z$.

Figure 6.8 shows the results obtained when the adaptive algorithm is applied to a first-order system

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

Notice the strange behavior of the output. This would have been even worse if the control signal had not been kept bounded in the simulation. The parameter estimates converge very quickly to values such that A and B have a common factor. The Diophantine equation is then singular, as shown in Example 6.6,

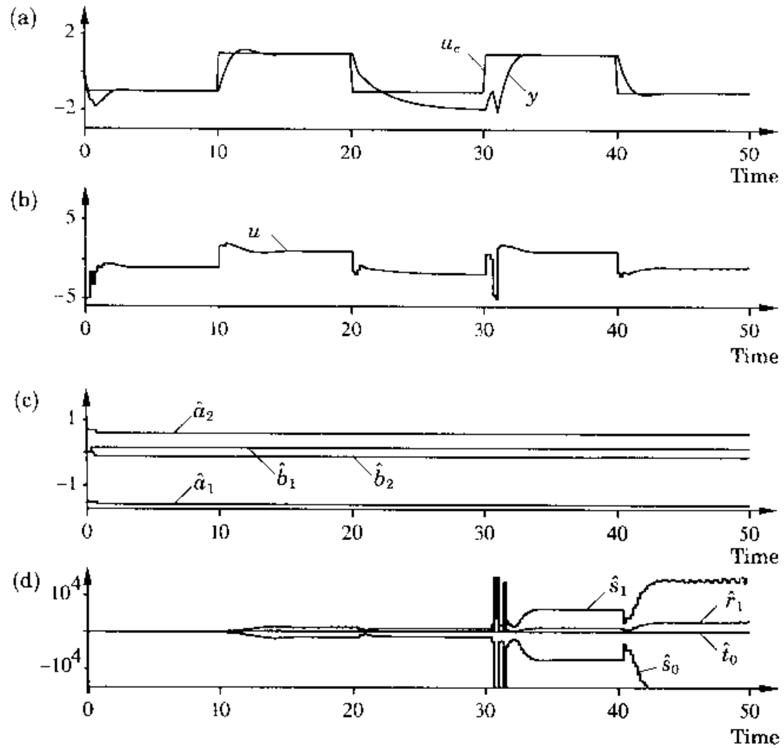

Figure 6.8 Simulation of an indirect adaptive pole placement controller based on a second-order process model of a first-order process. (a) Output and reference value. (b) Control signal. (c) Estimated process parameters. (d) Calculated controller parameters.

and the controller parameters become very large. The consequences of canceling a possible common factor and making a design for a first-order system are illustrated in Fig. 6.9. In this particular case a factor is canceled if poles and zeros are so close that

$$\left| A\left(\frac{-b_1}{b_0}\right) \right| = \left| \frac{b_1^2 - a_1 b_0 b_1 + a_2 b_0^2}{b_0^2} \right| \le 0.01 \qquad (6.37)$$

The performance is now very good. □

Summary

Parameter convergence for indirect adaptive algorithms depends critically on the assumptions of identifiability and persistency of excitation. Analysis of

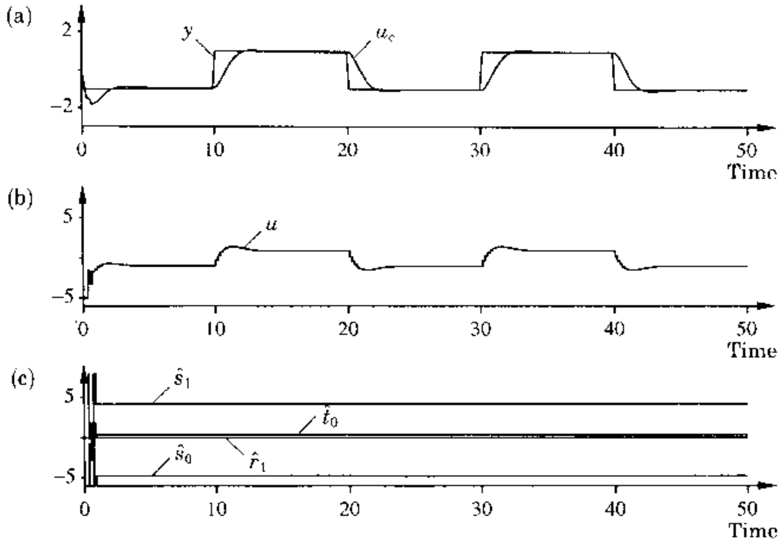

Figure 6.9 Simulation of an indirect adaptive pole placement controller based on a second-order process model. A possible common factor in the estimated process transfer function is canceled before the control law is calculated if the condition of Eq. (6.37) holds. (a) Output and reference value. (b) Control signal. (c) Calculated controller parameters.

the convergence rate of estimators shows that the convergence rate depends drastically on the underlying process being deterministic or stochastic. It also depends on the algorithm. A least-squares algorithm in the deterministic case gives convergence in a finite number of steps, provided that the input is persistently exciting. The gradient algorithms give exponential but generally much slower convergence than the least-squares algorithm. The convergence rate is much slower in the stochastic case. Analysis of the convergence rate for estimators gives only partial insight into the convergence rate of adaptive algorithms. To obtain a detailed understanding, it is necessary to consider that the input to the system is generated by feedback.

6.5 STABILITY OF DIRECT DISCRETE-TIME ALGORITHMS

Stability was discussed in connection with model-reference adaptive system in Chapter 5. It was in fact the key design issue in the MRAS. The problem was easy to resolve in the cases in which all the state variables were measured and for output feedback of systems in which the dynamics were SPR or could easily

be made SPR. In these cases the MRAS has the property that arbitrarily large adaptation gains can be used.

A stability proof for a direct discrete-time adaptive control law (MRAS or STR) for a general linear system will now be given. Some simplifications will be made in the algorithm to avoid too many technicalities.

The Algorithm

Direct algorithms for adaptive control were discussed in Section 3.5. We give the proof for a simple algorithm of this type. Consider a process described by the difference equation

$$A^*(q^{-1})y(t) = B^*(q^{-1})u(t-d) \tag{6.38}$$

Let the desired response from command signal to process output be characterized by

$$A_m^*(q^{-1})y(t) = t_0 u_c(t-d)$$

This specification implies that all process zeros are canceled. Furthermore, let the observer polynomial be A_o. A direct algorithm can then be formulated as follows. Estimate parameters of the model

$$A_o^* A_m^* y(t+d) = R^* u(t) + S^* y(t) = \varphi^T(t)\theta \tag{6.39}$$

where

$$\theta = \begin{pmatrix} r_0 & r_1 & \ldots & r_k & s_0 & s_1 & \ldots & s_l \end{pmatrix}^T$$

$$\varphi(t) = \begin{pmatrix} u(t) & u(t-1) & \ldots & u(t-k) & y(t) & y(t-1) & \ldots & y(t-l) \end{pmatrix}^T \tag{6.40}$$

The parameters are estimated by using the following projection estimator:

$$\begin{aligned} \hat\theta(t) &= \hat\theta(t-1) + \frac{\gamma\varphi(t-d)}{\alpha + \varphi^T(t-d)\varphi(t-d)}\, e(t) \\ e(t) &= y(t) - \varphi^T(t-d)\hat\theta(t-1) \end{aligned} \tag{6.41}$$

with $0 < \gamma < 2$ and $\alpha > 0$. This estimator is the same as Eqs. (6.22) except that φ now has index $t-d$ instead of t. The properties given in Theorem 6.3 are still valid.

The control law is

$$\hat R^* u(t) + \hat S^* y(t) = t_0 A_o^* u_c(t) \tag{6.42}$$

or, equivalently,

$$\hat\theta^T(t)\left(A_o^* A_m^* \varphi(t)\right) = t_0 A_o^* u_c(t) \tag{6.43}$$

where $u_c(t)$ is the desired setpoint. Notice that it must be required that $\hat\theta_1(t) = \hat r_0(t) \neq 0$; otherwise, the control law is not causal.

Preliminaries

Since the proof consists of several steps, we outline the basic idea. The properties of the estimator were given in Theorem 6.3, which proved that the estimates are bounded and that a normalized prediction error converges to zero. However, the theorem does not show that the estimates converge. By introducing the control law and the properties of the system to be controlled, it can then be established that the signals are bounded and that the controlled output converges to the command signal.

If the input and output signals of the system can be shown to be bounded, then φ given by Eqs. (6.40) is bounded. If $\varphi(t-d)$ is bounded for all t, it follows from Property (ii) of Theorem 6.3 that the prediction error $e(t)$ goes to zero. The following result is useful to establish the boundedness of φ.

LEMMA 6.2 Key technical lemma

Let $\{s_t\}$ be a sequence of real numbers and let $\{\sigma_t\}$ be a sequence of vectors such that

$$\|\sigma_t\| \le c_1 + c_2 \max_{0\le k\le t} |s_k|$$

Assume that

$$z_t = \frac{s_t^2}{\alpha_1 + \alpha_2 \sigma_t^T \sigma_t} \to 0 \tag{6.44}$$

and that

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

where $\alpha_1 > 0$ and $\alpha_2 > 0$. Then $\|\sigma_t\|$ is bounded.

Proof: The result is trivial if s_t is bounded. Hence assume that s_t is not bounded. Then there exists a subsequence $\{t_n\}$ such that $|s_{t_n}| \to \infty$ and $|s_t| \le s_{t_n}$ for $t \le t_n$. For this sequence it follows that

$$\left| \frac{s_t^2}{\alpha_1 + \alpha_2 \sigma_t^T \sigma_t} \right| \ge \frac{s_t^2}{\alpha_1 + \alpha_2 (c_1 + c_2 |s_t|)^2} \ge \frac{1}{\alpha_3 c_2^2} > 0$$

where $0 < \alpha_3 < \alpha_2$. This contradicts Eq. (6.44) and proves the statements. □

Main Result

The main result can now be stated as the following theorem.

THEOREM 6.7 Boundedness and convergence

Consider a system described by Eq. (6.38). Let the system be controlled with the adaptive control algorithm given by Eqs. (6.40), (6.41), and (6.42) where the command signal u_c is bounded. Assume that

A1: The time delay d is known.

A2: Upper bounds on the degrees of the polynomials A^* and B^* are known.

A3: The polynomial B has all its zeros inside the unit disc.

A4: The sign of $b_0 = r_0$ is known.

Then

(i) The sequences $\{u(t)\}$ and $\{y(t)\}$ are bounded.

(ii) $\lim_{t\to\infty} \left| A_m^*(q^{-1}) y(t) - t_0 u_c(t-d) \right| = 0$

Proof: Introduce the control error

$$\begin{aligned}
\varepsilon(t) &= A_o^* \left(A_m^* y(t) - t_0 u_c(t-d)\right) = P^* y(t) - t_0 A_o^* u_c(t-d) \\
&= P^* y(t) - \theta^T(t-d) \left(P^* \varphi(t-d)\right) \\
&= P^* e(t) + P^* \left(\theta^T(t-1) \varphi(t-d)\right) - \theta^T(t-d) \left(P^* \varphi(t-d)\right) \\
&= P^* e(t) + \sum_{i=0}^{\deg P} p_i \left(\theta(t-1-i) - \theta(t-d)\right)^T \varphi(t-d-i)
\end{aligned} \tag{6.45}$$

where $P = A_o A_m$ has been introduced to simplify the writing. The first two equalities are trivial. The third is obtained from Eq. (6.39), the fourth from Eqs. (6.41), and the last by expanding the expression.

It now follows from properties (ii) and (iii) of Theorem 6.3 that

$$\lim_{t\to\infty} \frac{\varepsilon(t)}{\sqrt{\alpha + \varphi^T(t-d)\varphi(t-d)}} = 0$$

It follows from the first equality in Eq. (6.45) that

$$A_o^* A_m^* y(t) = \varepsilon(t) + t_0 A_o^* u_c(t)$$

Since the polynomials A_o and A_m are stable and since u_c is bounded, it follows that

$$|y(t)| \le \alpha_1 + \beta_1 \max_{0\le k\le t} |\varepsilon(k)|$$

Moreover, since the polynomial B is stable, it follows that

$$|u(t-d)| \le \alpha_2 + \beta_2 \max_{0\le k\le t} |y(k)|$$

Hence

$$|\varphi(t-d)| \le \alpha_3 + \beta_3 \max_{0\le k\le t} |\varepsilon(k)|$$

If we apply Lemma 6.2, it follows that $\varphi(t)$ is bounded and that $\varepsilon(t) \to 0$ as $t \to \infty$. Since the polynomial A_o^* is stable, property (ii) also follows. □

Remark 1. We used an algorithm for which the details of the proof are simple. With minor modification the results can be extended to cover many of the direct algorithms given in Section 3.5.

Remark 2. A minor modification of the algorithm is necessary to ensure that $\hat{r}_0 \neq 0$. One way to do this is as follows: If $\hat{r}_0(t) = 0$, modify γ to give $\hat{r}_0(t) \neq 0$. Theorem 6.3 will still be valid with this modification of the algorithm. Since the estimator properties enter into the proof only via Theorem 6.3, the result still holds.

Remark 3. Notice that it does not follow that the parameter estimates converge. The fact that the control error nonetheless goes to zero depends on an interplay between the estimation and the control algorithms. This property is a special feature of direct algorithms.

Remark 4. The minimum-phase property is used to conclude that u is bounded when y is bounded.

Remark 5. Notice the similarity between Eq. (6.45) and the augmented error introduced in Chapter 5. □

Discussion

It has been established that a direct adaptive controller gives a closed-loop system with bounded signals and desired asymptotic properties, provided that Assumptions A1–A4 are valid. Assumptions A1 and A2 are necessary to write down the algorithm. Knowledge of the time delay (with a resolution corresponding to the sampling period) is essential. The signals will not be bounded if d is too small. Assumption A3 implies that the sampled system is minimum-phase; it is required because all process zeros are canceled in the design procedure. The error equation will not be linear in the parameters if this is not done. Assumption A4 is essential, since b_0 is absorbed in the adaptation gain γ, to guarantee that $\hat{r}_0(t) \neq 0$ for all times. Assumption A2 implies that the adaptive control law must have a sufficient number of parameters. This means that the model used to design the adaptive controller must be at least as complex as the process to be controlled. The consequences of violating the assumptions will be discussed later.

Extensions

The results can be extended in several different directions. Similar results can also be given in the continuous-time case, in which the underlying model can be written as

$$A(p)y(t) = B(p)u(t)$$

where A and B are polynomials in the differential operator $p = d/dt$. Assumptions A1–A4 are then replaced by the following assumptions:

A1′: The pole excess $\deg A - \deg B$ is known.

A2′: Upper bounds on the degrees of the polynomials A and B are known.

A3′: The polynomial B has all its zeros in the left half-plane.

A4′: The sign of b_0 is known.

The results can also be extended to systems with disturbances generated from known dynamics.

The gradient estimation algorithm can be replaced by other, more efficient methods. Theorem 6.3 then needs to be generalized. Many types of least-squares-like algorithms can be covered by replacing the function $V = \tilde{\theta}^T\tilde{\theta}$ in Theorem 6.3 by $V = \tilde{\theta}^T P^{-1}\tilde{\theta}$ and adding assumptions that guarantee that the eigenvalues of P stay bounded. Other control laws can also be treated. One important situation that has not been treated is the case in which the control signal is kept bounded by saturation. Theorem 6.3 still holds, but Theorem 6.7 does not, since Eq. (6.42) does not hold when the control signal saturates.

Gronwall-Bellman Lemma

The essential idea in the proof of Theorem 6.7 is the separation of the adaptive controller into two parts. First, some properties of the estimator are established that are independent of how the control signal is generated. Second, properties of the controlled system are derived. Convergence and stability are derived on the basis of the key technical lemma (Lemma 6.2). This procedure can be used for many different adaptive schemes.

The key technical lemma is a simplified version of the Gronwall-Bellman lemma, which is a standard tool for proving the existence of solutions to ordinary differential equations. There are both continuous-time and discrete-time versions of this lemma.

LEMMA 6.3 Gronwall-Bellman lemma: Continuous time

If $u, v \geq 0$, if c_1 is a positive constant, and if

$$u(t) \leq c_1 \int_0^t u(s)v(s)\,ds \tag{6.46}$$

then

$$u(t) \leq c_1 \exp\left(\int_0^t v(s)\,ds\right)$$ □

LEMMA 6.4 Gronwall-Bellman lemma: Discrete time

If $u, v \geq 0$, if c_1 is a positive constant, and if

$$u(t) \leq c_1 \sum_{k=0}^{t-1} u(k)v(k) \tag{6.47}$$

then

$$u(t) \leq c_1 \exp\left(\sum_{k=0}^{t-1} v(k)\right)$$ □

By using the Gronwall-Bellman lemma, many direct adaptive algorithms can be analyzed in the following way:

- Show that growth conditions such as Eq. (6.46) or Eq. (6.47) hold.
- Show properties analogous to Eq. (6.44) for the signals u and v.
- Use the Gronwall-Bellman lemma to get stability.

These steps can be used as a template for proving convergence and stability for adaptive algorithms.

6.6 AVERAGING

The results in the previous sections do not permit a detailed investigation of adaptive control algorithms. For example, no information about transient behavior is available until much more detailed analysis is undertaken. The conventional methods for investigating nonlinear systems involve investigation of equilibria and analysis of the local behavior near the equilibria. Such an approach will give only local properties, although in some special cases it may be possible to proceed further and obtain global properties. The results of the analysis can then be augmented by simulations. For purposes of this discussion it is useful to write the equations of motion of the complete system in a comprehensive form such as Eqs. (6.1) and (6.2) or Eqs. (6.3). In an adaptive system it is natural to separate the states of the system and the process parameters. The process parameters are changing more slowly than the states. This separation of time scales is used in the averaging theory to gain more insight about the properties of the closed-loop system. The idea of averaging originated in the analysis of planetary motion.

The Averaged Equations

The analysis of the dynamics of adaptive systems is generally quite complicated because the complete system is often of high order. Analysis of a direct algorithm for a discrete-time second-order system with four unknown parameters using a gradient method leads to a difference equation of order 8 (two states of the system, four parameters, and two difference equations to generate the regression variables). Ten more equations are obtained if a least-squares estimation algorithm is used.

Because of the special properties of adaptive systems, however, there is an approximate method that will simplify the analysis considerably. The basic idea is that the parameters change much more slowly than the other variables of the system. This property is intrinsic to the adaptive algorithms. If this were not the case, we could hardly justify using the notion of parameters.

To describe the averaging methods, consider the adaptive system described by Eqs. (6.1) and (6.2). The rate of change of the parameter $\hat{\theta}$ can be made

arbitrarily small by choosing the adaptation gain γ sufficiently small. For simplicity we use the simple gradient algorithm

$$\frac{d\hat{\theta}}{dt} = \gamma\varphi(\vartheta,\xi)e(\vartheta,\xi) \tag{6.48}$$

The product φe on the right-hand side depends on ϑ and ξ, where $\vartheta = \vartheta(\hat{\theta})$ varies slowly and ξ varies fast. The key idea in the averaging method is to approximate the product φe by

$$G(\hat{\theta}) = \operatorname{avg}\left\{\varphi\left(\vartheta(\hat{\theta}),\xi(\vartheta(\hat{\theta}),t)\right)e\left(\vartheta(\hat{\theta}),\xi(\vartheta(\hat{\theta}),t)\right)\right\}$$

where $\operatorname{avg}\{\cdot\}$ denotes the average and $\xi(\vartheta(\hat{\theta}),t)$ is computed under the assumption that the parameters $\hat{\theta}$ are constant. The average can be computed in many ways. Typical examples are

$$\operatorname{avg}\left\{f\left(\hat{\theta},\xi(\hat{\theta},t),t\right)\right\} = \frac{1}{T}\int_0^T f\left(\hat{\theta},\xi(\hat{\theta},t),t\right)dt$$

$$\operatorname{avg}\left\{f\left(\hat{\theta},\xi(\hat{\theta},t),t\right)\right\} = \lim_{T\to\infty}\int_0^T f\left(\hat{\theta},\xi(\hat{\theta},t),t\right)dt$$

$$\operatorname{avg}\left\{f\left(\hat{\theta},\xi(\hat{\theta},t),t\right)\right\} = E\,f\left(\hat{\theta},\xi(\hat{\theta},t),t\right)$$

The first alternative is applicable when f is periodic with period T, and the last equation applies when ξ is a stationary stochastic process. Notice that the averaged equations can be calculated only when the signals are bounded. This implies that the closed-loop system must be stable if the parameters $\hat{\theta}$ are fixed. The calculation of $\xi(\vartheta(\hat{\theta}),t)$ is a straightforward exercise in linear system analysis. However, the expressions may be complex for high-order systems. Symbolic calculation is a useful tool for carrying out the calculations. The use of averaging thus results in the following averaged nonlinear differential equation for the parameters:

$$\frac{d\bar{\theta}}{dt} - \gamma\operatorname{avg}\left\{\varphi\left(\vartheta(\bar{\theta}),\xi(\vartheta(\bar{\theta}),t)\right)e\left(\vartheta(\bar{\theta}),\xi(\vartheta(\bar{\theta}),t)\right)\right\} = 0 \tag{6.49}$$

This equation can also be written as

$$\frac{d\bar{\theta}}{dt} - \gamma\operatorname{avg}\left\{\left(G_{\varphi v}(\vartheta(\bar{\theta}),p)v\right)\left(G_{ev}(\vartheta(\bar{\theta}),p)v\right)\right\} = 0 \tag{6.50}$$

Notice that the transfer functions G_{ev} and $G_{\varphi v}$ depend on the averaged parameter $\bar{\theta}$. When the averaged equations are obtained, the behavior of the state variables ξ can be obtained by linear analysis.

Several averaging theorems give conditions for $\bar{\theta}$ being close to $\hat{\theta}$. The conditions typically require smoothness of the functions involved and periodicity or near periodicity of the time functions. There are also stochastic averaging theorems. Notice that averaging analysis was used in Theorems 4.1 and 4.2.

A significant advantage of averaging theory is that it reduces the dimensions of the problem. The theorems require that the adaptation gain be small, but experience has shown that averaging often gives a good approximation, even for large adaptation gains.

When the averaging equations are obtained, analysis proceeds in the conventional manner by investigation of the equilibria of the averaged equations and linearization at the equilibria to determine the local behavior. Notice that the averaged equations may possess equilibria (i.e., solutions to $\text{avg}\{\varphi e\} = 0$) even if the exact equations do not have an equilibrium. This corresponds to the case in which the true parameters are meandering in the neighborhood of the equilibrium to the averaged equation.

Sinusoidal Driving Forces

A simple case of averaging is when the external driving force is sinusoidal, that is, $\nu(t) = u_0 \sin \omega t$. The signals φ and e are then given by

$$\varphi(t) = G_{\varphi\nu}(\vartheta, \omega)\nu(t)$$
$$e(t) = G_{e\nu}(\vartheta, \omega)\nu(t)$$

Notice that controller parameters ϑ depend on $\bar{\theta}$. The following result is useful for calculation of the averages.

LEMMA 6.5 Averaging for sinusoidal input

Let G_v and G_w be stable transfer functions, and let v and w denote the steady-state responses of the corresponding systems to the input $u_c = u_0 \sin \omega t$. The mean value of the product vw is then given by

$$\begin{aligned}\text{avg}(vw) &= \frac{u_0^2}{2} |G_v(i\omega)| \, |G_w(i\omega)| \cos(\arg G_v(i\omega) - \arg G_w(i\omega)) \\ &= \frac{u_0^2}{2} \operatorname{Re}(G_v(i\omega) G_w(-i\omega))\end{aligned}$$

Proof: The signals v and w have the amplitudes $|G_v(i\omega)|$ and $|G_w(i\omega)|$; their phase angles are $\arg G_v(i\omega)$ and $\arg G_w(i\omega)$. Integrating over one period gives the result. □

A true parameter equilibrium exists if the equation

$$G_{e\nu}\left(\vartheta(\bar{\theta}), \omega\right) = 0$$

has a unique solution. To derive a necessary condition we consider the averaged equation

$$\frac{d\bar{\theta}}{dt} = \gamma \operatorname{Re}\left\{G_{\varphi\nu}\left(\vartheta(\bar{\theta}), \omega\right) R_\nu G_{e\nu}^T\left(\vartheta(\bar{\theta}), -\omega\right)\right\} \tag{6.51}$$

where

$$R_\nu = \text{avg}\left(\nu \nu^T\right)$$

A necessary condition for Eq. (6.51) to have a unique parameter equilibrium is that ν and $\bar{\theta}$ have equal dimension and that R_ν be of full rank. To have a unique parameter equilibrium for slow external driving signals, it is thus necessary that the number of estimated parameters be less than or equal to the number of external driving signals and that the external driving signals be persistently exciting. This result indicates that there may be some disadvantages to overparameterization, contrary to what is indicated in Theorem 6.7. The local stability of the equilibrium $\bar{\theta}^0$ is given by the linearized equation

$$\frac{dx}{dt} = Ax$$

where x denotes the deviation from the equilibrium $\bar{\theta} - \bar{\theta}^0$ and

$$A = G_{\varphi\nu}\left(\vartheta(\bar{\theta}^0), \omega\right) R_\nu \frac{\partial}{\partial\theta} G_{e\nu}^T\left(\vartheta(\bar{\theta}^0), \omega\right)$$

The preceding equations can be applied to slow or constant perturbations by setting $\omega = 0$, provided that the assumptions of averaging are fulfilled.

An Example of Averaging Analysis

Consider a process with the transfer function $kG(s)$ and an adjustable feedforward gain. Find a feedforward gain $\hat{\theta}$ such that the input-output behavior matches the transfer function $k_0 G_m(s)$ as well as possible. It is assumed that $k > 0$ and $k_0 > 0$. The case $G_m = G$ was discussed in Chapter 5. Two different algorithms for updating the gain were proposed in Chapter 5: the MIT rule and the SPR rule. The algorithms are

$$\begin{aligned}\frac{d\hat{\theta}}{dt} &= -\gamma y_m e \qquad \text{(MIT)} \\ \frac{d\hat{\theta}}{dt} &= -\gamma u_c e \qquad \text{(SPR)}\end{aligned} \tag{6.52}$$

where u_c is the command signal, $y_m = k_0 G_m u_c$ is the model output, and e is the error defined by

$$e(t) = y - y_m = kG(p)\left(\hat{\theta}(t) u_c(t)\right) - k_0 G_m(p) u_c(t)$$

The analysis in Section 5.5 shows that the MIT rule gives a closed-loop system that is globally stable for any adaptation gain γ in the "ideal" case, when $G = G_m$ and G is SPR. In the presence of unmodeled dynamics it is, of course, highly unrealistic to assume that a transfer function is SPR. So far, no stability result has been given for the MIT rule. However, Example 5.5 indicates that the

MIT rule will be unstable for sufficiently high adaptation gains if the system is not SPR.

We now investigate the algorithms under nonideal conditions, using averaging. Inserting the expressions for y_m and e into the equations for the parameters, we get

$$\begin{aligned} \frac{d\hat{\theta}}{dt} + \gamma(k_0 G_m u_c)\left(kG(\hat{\theta}u_c) - k_0 G_m u_c\right) &= 0 \\ \frac{d\hat{\theta}}{dt} + \gamma u_c\left(kG(\hat{\theta}u_c) - k_0 G_m u_c\right) &= 0 \end{aligned} \tag{6.53}$$

where the first equation holds for the MIT rule and the second holds for the SPR rule. The corresponding averaging equations are

$$\begin{aligned} \frac{d\bar{\theta}}{dt} + \gamma\left(\bar{\theta}kk_0 \operatorname{avg}\{(G_m u_c)(Gu_c)\} - k_0^2 \operatorname{avg}\{(G_m u_c)^2\}\right) &= 0 \\ \frac{d\bar{\theta}}{dt} + \gamma\left(\bar{\theta}k \operatorname{avg}\{u_c(Gu_c)\} - k_0 \operatorname{avg}\{u_c(G_m u_c)\}\right) &= 0 \end{aligned} \tag{6.54}$$

The equilibrium parameters are

$$\begin{aligned} \bar{\theta}_{\text{MIT}} &= \frac{k_0}{k}\frac{\operatorname{avg}\{(G_m u_c)^2\}}{\operatorname{avg}\{(G_m u_c)(Gu_c)\}} \\ \bar{\theta}_{\text{SPR}} &= \frac{k_0}{k}\frac{\operatorname{avg}\{u_c(G_m u_c)\}}{\operatorname{avg}\{u_c(Gu_c)\}} \end{aligned} \tag{6.55}$$

The equilibrium values correspond to the true parameters for all command signals u_c only if $G = G_m$ (i.e., there are no unmodeled dynamics). When $G \neq G_m$, the equilibrium obtained will depend on the command signal as well as on the unmodeled dynamics. Notice that the equilibrium value obtained for the MIT rule minimizes the actual mean square error.

The stability conditions for the averaged equations (Eqs. 6.54) are

$$\begin{aligned} \gamma \operatorname{avg}\{(G_m u_c)(Gu_c)\} &> 0 \qquad \text{(MIT)} \\ \gamma \operatorname{avg}\{u_c(Gu_c)\} &> 0 \qquad \text{(SPR)} \end{aligned}$$

The averaged equation when the MIT rule is used will thus give a stable equilibrium for all command signals if $G_m = G$. The stability condition depends on the command signal and the process dynamics as well as on the response model.

For the SPR rule the stability condition depends only on the command signal and on the process dynamics. The equilibrium is stable for all command signals if G is SPR. For processes that are not SPR the equilibrium may well be unstable. Consider the case of a command signal composed of a constant and a sum of sinusoids:

$$u_c(t) = a_0 + 2\sum_{k=1}^{n} a_k \sin \omega_k t$$

If Lemma 6.5 is used, the stability conditions for $\gamma > 0$ become

$$a_0^2 G_m(0) + \sum_{k=1}^{n} a_k^2 |G_m(i\omega_k)|\,|G(i\omega_k)| \cos\{\arg G_m(i\omega_k) - \arg G(i\omega_k)\} > 0$$

$$a_0^2 G(0) + \sum_{k=1}^{m} a_k^2 \operatorname{Re} G(i\omega_k) > 0$$

For a single sinusoidal command signal the MIT rule gives a stable equilibrium if the phase lags of G_m and G differ by at most 90° at the frequencies of the input signal. The SPR rule, on the other hand, gives a stable equilibrium if the phase lag of the process is at most 90°.

For command signals containing several sinusoidals the equilibrium can still be stable, provided that the command signal is dominated by components with frequencies in the range in which the phase lag of the process is less than 90°. Notice that it helps to filter the command signal so that the signals in the frequency range in which the plant has a phase shift of more than 90° are attenuated. In the MIT rule, reduction of the gain of the model can also be reduced at high frequencies. It follows from Eqs. (6.54) that the convergence rate of the parameters is strongly signal-dependent. The value of normalization as described in Section 5.3 is that the convergence rate becomes less dependent on the signal amplitudes. The preceding calculations are illustrated by an example.

EXAMPLE 6.8 Sinusoidal command signal

Consider a reference model with the transfer function

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

Assume that the process has the transfer function

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

Furthermore, let the command signal be a sinusoid with unit amplitude and frequency ω. Equations (6.55) give the equilibrium values

$$\begin{aligned} \bar{\theta}_{\text{MIT}} &= \frac{k_0}{k}\frac{b^2+\omega^2}{b^2} \\ \bar{\theta}_{\text{SPR}} &= \frac{k_0}{k}\frac{a(b^2+\omega^2)}{b(ab-\omega^2)} \qquad \omega < \sqrt{ab} \end{aligned}$$

The stability conditions show that the MIT rule is stable for all ω, but the SPR rule is stable only if $\omega < \sqrt{ab}$. Figure 6.10 shows the estimates of the gain for the case behavior $a = 1$ and $b = 10$ when the input signals have frequencies

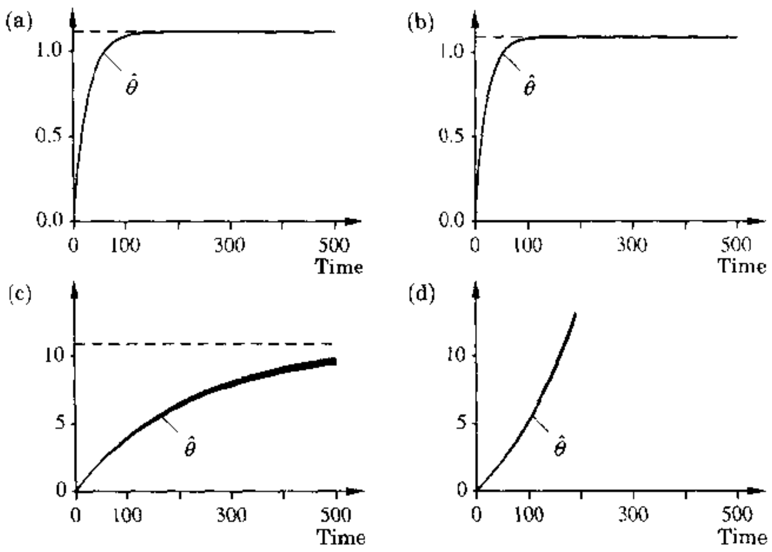

Figure 6.10 Estimated feedforward gains obtained by the MIT rule with sinusoidal input signals having frequencies (a) $\omega = 3$; (b) $\omega = 3.4$; and the SPR rule when (c) $\omega = 3$; (d) $\omega = 3.4$, for a system with $G_m = 1/(s+1)$ and $G = 10/((s+1)(s+10))$. The dashed lines are the equilibrium values obtained from averaging analysis.

$\omega = 3$ and $\omega = 3.4$. The equilibrium values predicted by the averaging theory are also shown in the figure. The SPR is unstable for $\omega = 3.4 > \sqrt{10}$. Also notice the drastic difference in the equilibrium values between the different updating methods. The desired equilibrium value is $\theta_0 = k_0/k$.

The behavior is well predicted by the averaging analysis. Notice the difference in convergence rates. Initially, when $\bar{\theta} = 0$, the rates of changes are given by

$$\dot{\bar{\theta}}_{\text{MIT}} = \gamma k_0^2 \operatorname{avg}\{(G_m u_c)^2\}$$
$$\dot{\bar{\theta}}_{\text{SPR}} = \gamma k_0 \operatorname{avg}\{u_c(G_m u_c)\}$$

These expressions clearly show that the initial rates decrease with increasing frequency because $|G_m(i\omega)|$ decreases with frequency. For the SPR rule the rate decreases even more because of the phase lag between u_c and $G_m u_c$. □

In conclusion, we find that averaging analysis gives useful insights. It shows that analysis of the ideal case can be quite misleading. Even in the simple case of adjustment of a feedforward gain, unmodeled dynamics together with high-frequency excitation signals may lead to instability of the equilibrium. The equilibrium analysis also makes interesting contributions to the comparison of the MIT and SPR rules. First, the equilibrium of the MIT rule has a good physical interpretation as the parameter that minimizes the mean square error. Second, the apparent advantage of the SPR rule that very high adaptation gains can be used vanishes. In practical situations, there are always unmodeled dynamics. In the presence of unmodeled dynamics the gain must be kept small to maintain stability.

6.7 APPLICATION OF AVERAGING TECHNIQUES

In the previous sections, idealized cases were investigated. The convergence and stability analysis of self-tuning regulators were based on Assumptions A1–A4 and the premise that there are no disturbances. In Chapter 5 the stability of MRAS was proved under the SPR assumption on certain transfer functions. Assumption A2 in Theorem 6.7 implies that the model used to design the adaptive controller must be at least as complex as the process to be controlled. This is highly unrealistic because real processes are often distributed and also nonlinear.

In practice, adaptive controllers are based on simplified models. It is therefore of interest to investigate what happens when the process is more complex than assumed in the design of the controller. In this case the process is said to have *unmodeled dynamics*. If a controller is able to control processes with unmodeled dynamics and/or disturbances, we say that the controller is *robust*.

Analysis of a Simple MRAS

A simple model-reference adaptive system for a process of first order was derived in Example 5.2 by using the MIT rule. In Example 5.7 the same problem was considered, and an MRAS was obtained by using Lyapunov's stability theory. We now use averaging theory to investigate the properties of the controller. In designing the adaptive controller it is assumed that the nominal transfer function of the process is

$$G(s) = \frac{b}{s+a} \tag{6.56}$$

which is not necessarily the true transfer function of the process. The desired closed-loop system has the transfer function

$$G_m(s) = \frac{b_m}{s + a_m}$$

A model-reference adaptive control law was derived in Example 5.7 by using Lyapunov theory. A block diagram of the closed-loop system is given in

Fig. 5.11. The system is described by the equations

$$
\begin{aligned}
\frac{d\hat{\theta}_1}{dt} &= -\gamma u_c e \\
\frac{d\hat{\theta}_2}{dt} &= \gamma y e \\
e &= y - y_m \\
y &= G(p)u \\
y_m &= G_m(p)u_c \\
u &= \hat{\theta}_1 u_c - \hat{\theta}_2 y
\end{aligned}
\tag{6.57}
$$

where u_c is the reference signal, u is the process input, y is the process output, y_m is the output of the reference model, e is the error, $\hat{\theta}_1$ is the adjustable feedforward gain, and $\hat{\theta}_2$ is the adjustable feedback gain.

It is not possible to give a complete analysis of Eqs. (6.57) for general reference signals; approximations must be made even in a simple case like this. We now investigate the adaptive system when the reference signal is sinusoidal. The equilibrium points are first explored, and the behavior in their neighborhood is then investigated by averaging and linearization.

Equilibrium Values for the Parameters

It follows from Eqs. (6.57) that the parameters $\hat{\theta}_1$ and $\hat{\theta}_2$ are constant when the error e is zero. The conditions for e to be zero will now be investigated. The signal transmission from the command signal u_c to the output y is described by the transfer function

$$G_c = \frac{\hat{\theta}_1 G}{1 + \hat{\theta}_2 G}$$

and the control error becomes

$$e(t) = y(t) - y_m(t) = (G_c(p) - G_m(p))\, u_c(t)$$

Let the reference signal be $u_c = u_0 \sin \omega t$. The error e is then zero if

$$G_c(i\omega) = G_m(i\omega)$$

or

$$\hat{\theta}_1^0 G(i\omega) = \hat{\theta}_2^0 G_m(i\omega) G(i\omega) + G_m(i\omega) \tag{6.58}$$

This equation can be solved for $\hat{\theta}_1^0$ and $\hat{\theta}_2^0$ by equating the real and imaginary parts. There is a unique solution if $\operatorname{Im}\{G(i\omega)\} \neq 0$. The solutions are easily obtained by dividing Eq. (6.58) by $G_m G$ and G, respectively, and taking

imaginary parts. This gives

$$
\begin{aligned}
\hat{\theta}_1^0 &= \frac{\operatorname{Im}\{1/G(i\omega)\}}{\operatorname{Im}\{1/G_m(i\omega)\}} \\
\hat{\theta}_2^0 &= -\frac{\operatorname{Im}\{G_m(i\omega)/G(i\omega)\}}{\operatorname{Im} G_m(i\omega)}
\end{aligned}
\tag{6.59}
$$

In the nominal case we get $\hat{\theta}_1^0 = b_m/b$ and $\hat{\theta}_2^0 = (a_m - a)/b$. These equilibrium values do not depend on the frequency of the command signal. They also correspond to the desired feedback gains.

Averaging

The command signal u_c is the only external signal; hence $v = u_c$. Furthermore, $\varphi^T = \begin{pmatrix} u_c & y \end{pmatrix}$. To obtain the averaging equations, the transfer functions G_{ev} and $G_{\varphi v}$ are first calculated:

$$
\begin{aligned}
G_{ev} &= \frac{\hat{\theta}_1 G}{1 + \hat{\theta}_2 G} - G_m \\
G_{\varphi v}^T &= \begin{pmatrix} -1 & \dfrac{\hat{\theta}_1 G}{1 + \hat{\theta}_2 G} \end{pmatrix}
\end{aligned}
$$

By using Lemma 6.5 the averaged equations can now be written as

$$
\begin{aligned}
\frac{d\bar{\theta}_1}{dt} &= -\frac{\gamma u_0^2}{2} Re\left\{ \frac{\bar{\theta}_1 G(i\omega)}{1 + \bar{\theta}_2 G(i\omega)} - G_m(i\omega) \right\} \\
\frac{d\bar{\theta}_2}{dt} &= \frac{\gamma u_0^2}{2} Re\left\{ \left(\frac{\bar{\theta}_1 G(i\omega)}{1 + \bar{\theta}_2 G(i\omega)} - G_m(i\omega) \right) \frac{\bar{\theta}_1 G(-i\omega)}{1 + \bar{\theta}_2 G(-i\omega)} \right\}
\end{aligned}
\tag{6.60}
$$

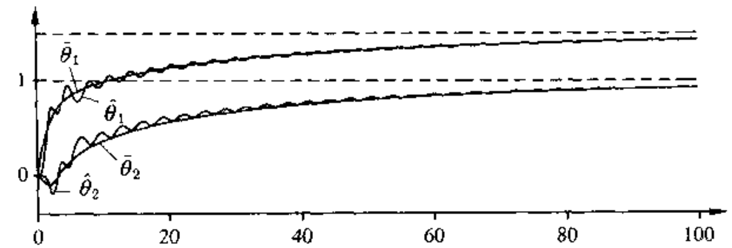

Figure 6.11 Parameter estimates and their approximation by the averaging method. The dashed lines show the equilibrium values of the gains.

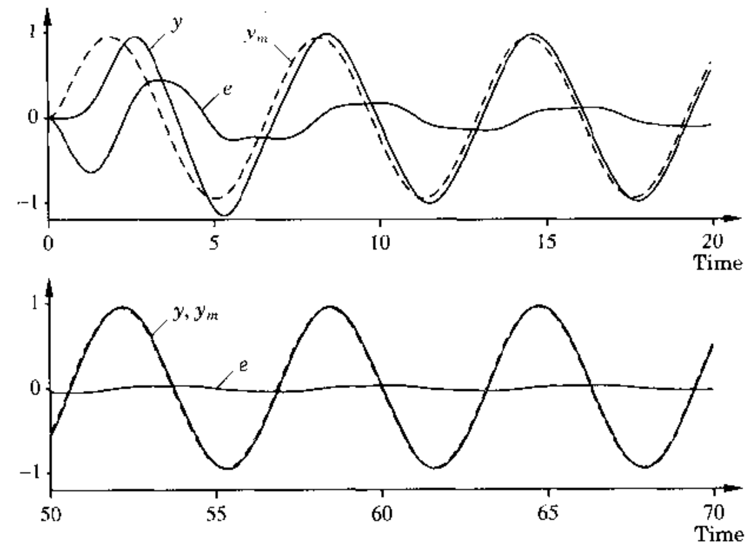

Figure 6.12 System output y (solid line) and the output of the reference model y_m (dashed line) and error e for Example 6.9 for $t=0$–20 and $t=50$–70.

Notice that these equations are valid also when G is a general transfer function, that is, G does not need to satisfy Eq. (6.56).

EXAMPLE 6.9 Accuracy of averaging

Consider the particular case of $a = 1$, $b = 2$, and $a_m = b_m = 3$. Let the adaptation gain γ be 1, and let the command signal be $u_0 \sin t$. The time histories of the parameter estimates $\hat{\theta}_1$, $\hat{\theta}_2$ and their approximations $\bar{\theta}_1$, $\bar{\theta}_2$ are shown in Fig. 6.11. The figure shows that the averaging gives a good approximation in this case. Notice that the approximation improves with time. The process output y and the output of the reference model y_m are shown in Fig. 6.12. Notice that the signals are already quite close after 10 s, although the parameters are quite far from their correct values at this time. The error $e = y - y_m$ thus appears to converge much faster than the parameters. This was seen for several different adaptive controllers in the previous chapters. Also notice that much faster convergence will be obtained with a recursive least-squares method. □

Local Stability

The stability of the equilibrium of the averaged equations (Eqs. 6.60) will now

be investigated. Straightforward but tedious calculations give the following linearized equation:

$$\frac{dx}{dt} = Ax \tag{6.61}$$

where x is a vector whose two components are the deviations of $\bar{\theta}_1$ and $\bar{\theta}_2$ from their equilibrium values and the matrix A is given by

$$A = \frac{\gamma u_0^2 |G_m|}{2\bar{\theta}_1^0} \begin{pmatrix} -\cos\theta_m & |G_m|\cos 2\theta_m \\ |G_m| & -|G_m|^2 \cos\theta_m \end{pmatrix} \tag{6.62}$$

where $\bar{\theta}_1^0$ is the equilibrium value of $\bar{\theta}_1$ and $\theta_m = \arctan(\omega/a_m)$. The matrix A has the characteristic equation

$$\lambda^2 + \alpha\lambda(1 + \cos^2\theta_m) + \alpha^2 \sin^2\theta_m = 0$$

where

$$\alpha = \frac{\gamma u_0^2 a_m b}{2\,(a_m^2 + \omega^2)}$$

The characteristic equation has its zeros in the left half-plane if $\omega \neq 0$. The equilibrium of the linearized equation (Eq. 6.61) is thus stable for all $\omega \neq 0$. The investigated MRAS has been designed by using Lyapunov theory. In the idealized case the transfer function (6.56) is SPR, and it is expected that the MRAS should have good performance.

Unmodeled Dynamics

The consequences of unmodeled dynamics are now investigated for the MRAS shown in Fig. 5.11. This system was designed on the basis of the assumption that the transfer function of the process has the form (6.56). We now investigate what happens if the process actually has a pole excess larger than 1. Before we go into details, a specific example is investigated.

EXAMPLE 6.10 Unmodeled dynamics

Assume that the nominal transfer function (6.56) has $a = 1$ and $b = 2$ but that the actual transfer function is

$$G(s) = \frac{458}{(s+1)(s^2 + 30s + 229)} \tag{6.63}$$

The dynamics correspond to the nominal plant $2/(s+1)$ cascaded with $229/(s^2 + 30s + 229)$. The process thus has two poles $s = -15 \pm 2i$, which were neglected in the model used to design the adaptive controller. Figure 6.13 shows the behavior of the controller parameters when the command signal is a step and

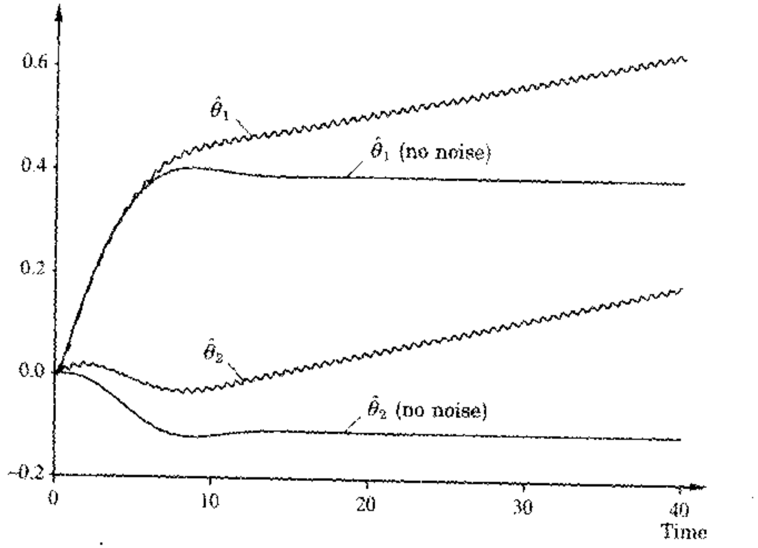

Figure 6.13 Controller parameters $\hat{\theta}_1$ and $\hat{\theta}_2$ when the adaptive control law of Eqs. (6.57) is applied to the process of Eq. (6.63). The command signal is a step, and there is sinusoidal measurement noise. The smooth curves show the behavior when there is no measurement noise.

there is a sinusoidal measurement error. Figure 6.14 shows the behavior of the parameters when the command signal is sinusoidal with different frequencies. □

Example 6.10 shows that the presence of unmodeled dynamics will drastically change the behavior of the adaptive system. Figure 6.14 shows that the equilibrium depends on the frequency of the command signal and that it may be unstable for certain frequencies. We now attempt to understand the mechanisms that change the behavior of the system so drastically and to find suitable remedies.

Step Commands

First, the behavior illustrated in Fig. 6.13 is analyzed. The case of step commands is first investigated when there is no measurement noise. When $\omega = 0$, the equilibrium condition of Eq. (6.58) reduces to

$$\hat{\theta}_2 = \frac{1}{G_m(0)}\hat{\theta}_1 - \frac{1}{G(0)} \tag{6.64}$$

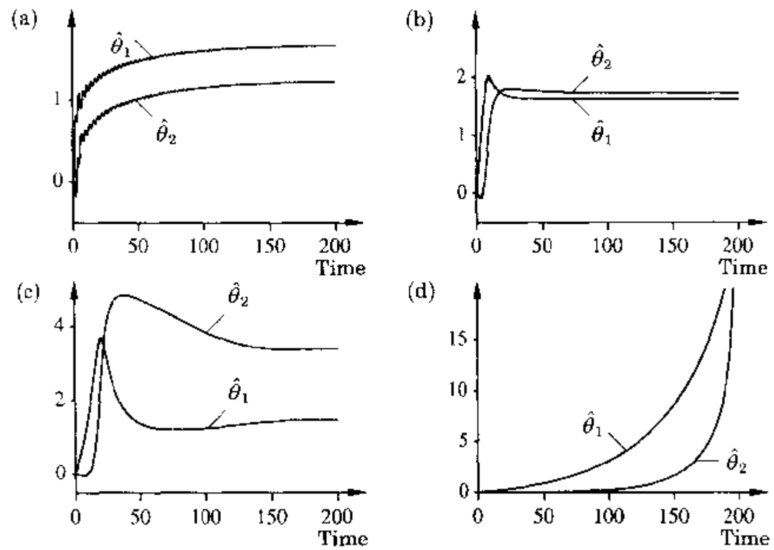

Figure 6.14 Controller parameters $\hat{\theta}_1$ and $\hat{\theta}_2$ when the adaptive control law of Eqs. (6.57) is applied to the process of Eq. (6.63) when the command signal is $u_c = \sin \omega t$ with (a) $\omega = 1$; (b) $\omega = 3$; (c) $\omega = 6$; (d) $\omega = 20$.

The equilibrium set is thus a straight line in the parameter space. The line is uniquely determined by the steady-state gains $G(0)$ and $G_m(0)$. Notice in particular that the equilibrium set is not a point. This is easily understood from the viewpoint of system identification. We wish to determine two parameters, $\hat{\theta}_1$ and $\hat{\theta}_2$. However, the excitation used is a step that is persistently exciting of first order and thus admits determination of only one parameter. (See Example 2.5.)

Averaging is now applied to obtain further insight into the behavior of the system. The averaging analysis applies to the set of parameter values such that the closed-loop system is stable for fixed parameters. To find this set, notice that the closed-loop system is a linear time-invariant system when parameters $\hat{\theta}_1$ and $\hat{\theta}_2$ are constant. The closed-loop eigenvalues are the zeros of the equation

$$1 + \hat{\theta}_2 G(s) = 0$$

A necessary condition for stability is that $1 + \hat{\theta}_2 G(s)$ has its roots in the left half-plane. This condition is also sufficient in the nominal case because the transfer function $G(s)$ is then SPR, and arbitrarily large feedback gains can be used. When there are unmodeled dynamics, the transfer function $G(s)$ is usually not SPR and the closed-loop system typically becomes unstable when $\hat{\theta}_2$ is sufficiently large.

EXAMPLE 6.11 Step commands

With the transfer function of Eq. (6.63) used in Example 6.10, the closed-loop characteristic equation is given by

$$(s+1)(s^2+30s+229)+458\theta_2 = 0$$

or

$$s^3+31s^2+259s+229+458\theta_2 = 0$$

This equation has all roots in the left half-plane if

$$-0.5 < \hat{\theta}_2 < 17.03 = \theta_2^{stab}$$

The averaged equations for the parameter estimates are obtained by setting $\omega = 0$ in Eqs. (6.60). If it is assumed that $G_m(0) = 1$, the equations become

$$\begin{aligned}\frac{d\bar{\theta}_1}{dt} &= -\frac{\gamma u_0^2}{2}\left(\frac{\bar{\theta}_1 G(0)}{1+\bar{\theta}_2 G(0)} - 1\right)\\ \frac{d\bar{\theta}_2}{dt} &= \frac{\gamma u_0^2}{2}\,\frac{\bar{\theta}_1 G(0)}{1+\bar{\theta}_2 G(0)}\left(\frac{\bar{\theta}_1 G(0)}{1+\bar{\theta}_2 G(0)} - 1\right)\end{aligned} \tag{6.65}$$

These differential equations have the equilibrium set of Eq. (6.64).

Close to the equilibrium set, the equations are described by the following linearized equation:

$$\frac{dx}{dt} = \frac{\gamma u_0^2}{2\theta_1^0}\begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} x \tag{6.66}$$

where $x_1 = \bar{\theta}_1 - \theta_1^0$ and $x_2 = \bar{\theta}_2 - \theta_2^0$. Consider a point away from the equilibrium line, that is, $x_2 = x_1 + \delta$ or $\bar{\theta}_2 = \bar{\theta}_1 - 1/G(0) + \delta$. The velocity of the state vector at that point is $\dot{x}_1 = \gamma u_0^2 \delta/\theta_1^0$, $\dot{x}_2 = -\gamma u_0^2\delta/\theta_1^0$. The vector field of the linearized equation is thus as shown in Fig. 6.15. The vector field thus

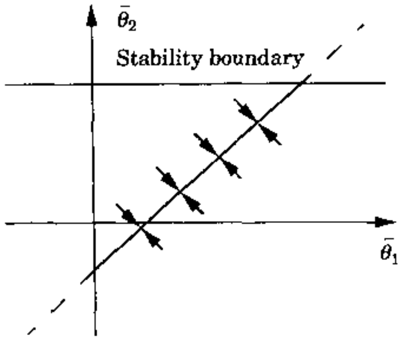

Figure 6.15 Equilibrium set and local behavior of the averaged equations.

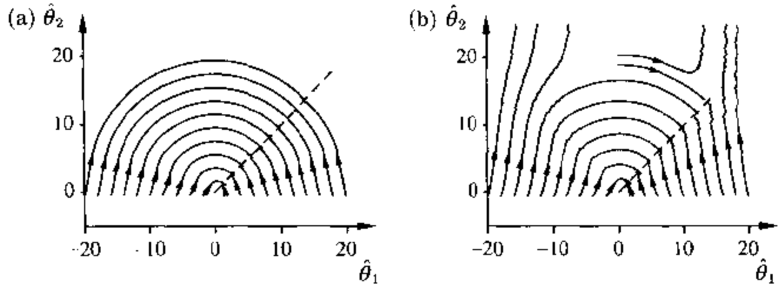

Figure 6.16 Phase plane of the controller parameters (a) in the nominal case of $G(s) = 2/(s+1)$ and (b) in the case of unmodeled dynamics Eq. (6.63). The dashed lines are the equilibrium sets of the parameters in the nominal case.

pushes the parameter toward the equilibrium for $\bar{\theta}_1 > 0$ and away from the equilibrium for $\bar{\theta}_1 < 0$. Notice that the system is not structurally stable because one eigenvalue of the linearized equation is zero. This means that we can expect drastically different properties when the system is perturbed.

It is usually difficult to go beyond the local analysis. However, in this particular case it is possible to obtain the global properties of the averaged equation. Outside the equilibrium set of Eq. (6.64), the averaged equations (Eqs. 6.65) can be divided to give

$$\frac{d\bar{\theta}_2}{d\bar{\theta}_1} = -\frac{G(0)\bar{\theta}_1}{1+\bar{\theta}_2 G(0)}$$

This differential equation has the solution

$$\bar{\theta}_2^2 + \frac{2}{G(0)}\bar{\theta}_2 + \bar{\theta}_1^2 = \text{const}$$

The parameters of the averaged equations will thus move along circular paths with the center at $(0, -1/G(0))$. The motion is clockwise for $\bar{\theta}_2 > \bar{\theta}_1 - 1/G(0)$ and counterclockwise for $\bar{\theta}_2 < \bar{\theta}_1 - 1/G(0)$. The motion slows down and stops when the parameters reach the equilibrium set

$$\left\{\bar{\theta}_1,\ \bar{\theta}_2 \middle| \bar{\theta}_1 > 0,\ \bar{\theta}_2 = \bar{\theta}_1 - 1/G(0)\right\}$$

The averaged equation approximates the nonlinear equations for the parameters only for parameters such that the closed-loop system is stable. In the nominal case, when the transfer function of the plant is $G(s) = 2/(s+1)$, the stability region is $-1/G(0) < \hat{\theta}_2$. In the case of unmodeled dynamics the stability region is defined by $-1/G(0) < \hat{\theta}_2 < \hat{\theta}_2^{stab}$. This means that trajectories that start far away from the origin will escape from the stability region. Figure 6.16 shows the actual parameter paths in the nominal case and for the unmodeled dynamics given by the transfer function of Eq. (6.63) in Example 6.10. With unmodeled dynamics the trajectories will diverge if the initial values are too large. The deviation from circular arcs is due to the initial transient when $y(t)$ is different from the equilibrium value. The adaptation gain used in the example is quite large ($\gamma = 1$). The trajectories will be arbitrarily close to circles by choosing γ sufficiently small. The "jitter" in the trajectories in Fig. 6.16(b) is caused by oscillations in the parameters, not numerical errors. □

The analysis and the simulations show that the adaptive system can be unstable if the input signal is a step and if there are unmodeled dynamics.

Measurement Noise

We now investigate the effects of measurement noise. The simulation shown in Fig. 6.13 indicates that measurement noise may cause the parameters to drift. Figure 6.17 shows parameter $\hat{\theta}_2$ as a function of parameter $\hat{\theta}_1$ with and without measurement noise. The simulation indicates that the equilibrium is lost in the presence of measurement noise. The parameters move toward a set close to the equilibrium set, oscillate rapidly in the neighborhood of this set, and drift along the set. The analysis tools developed will now be used to explain the behavior of the system. Assume that the command signal is a step with amplitude u_0 and that the measurement noise can be modeled as an additive zero mean signal n at the process output. It follows from Eqs. (6.57) that

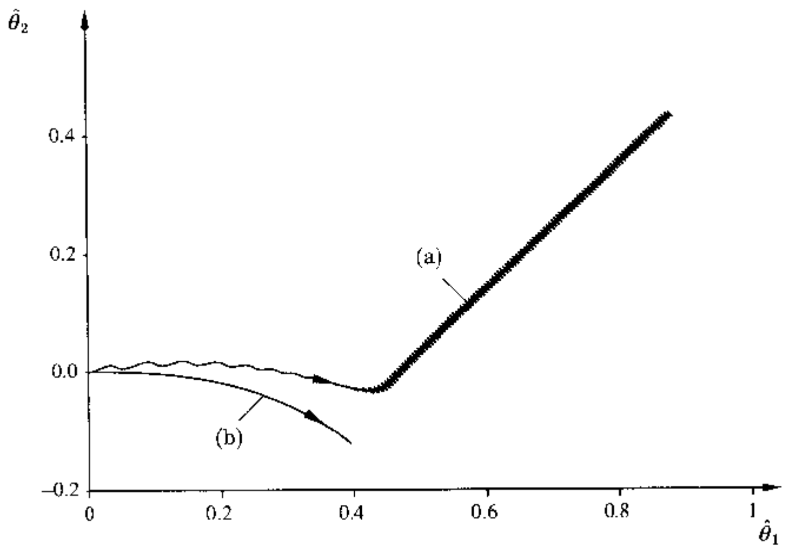

Figure 6.17 Phase plane of the controller parameters (a) with and (b) without measurement noise.

the error cannot be made identically zero by proper choice of the parameters. Hence no true equilibrium exists such that the parameters are constant. The phenomenon is a typical behavior of a system that lacks structural stability. Intuitively, the results can be explained as follows: A step input is persistently exciting of order 1 only, which means that it admits consistent estimation of one parameter only. When two parameters are adjusted, the equilibrium values of the parameters make a submanifold, not a point. Measurement errors and other disturbances may cause the parameters to drift along the equilibrium set. In the presence of unmodeled dynamics, the feedback gain may then become so large that the closed-loop system becomes unstable. By using averaging, the equilibrium set and the drift rate along the set can be determined.

The parameter values will drift also in the nominal case. However, the closed-loop system is stable for all parameter values.

Sinusoidal Command Signals

Several of the difficulties encountered with step commands are due to the fact that a step is persistently exciting of first order only. This means that the equilibrium set is a manifold and only a linear combination of the parameters can be determined. With a sinusoidal command signal that is persistently exciting of second order, two parameters can be determined consistently. It may therefore be expected that some of the difficulties will disappear. However, the simulation shown in Fig. 6.14 indicates that there are some problems with sinusoidal command signals in combination with unmodeled dynamics.

As before, it is assumed that the adaptive controller is designed as if the process were described by the transfer function

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

Since the character of the unmodeled dynamics is important, it is assumed that the actual plant is described by the frequency function

$$G(i\omega) = \frac{b}{a+i\omega}\, r(\omega)e^{-i\phi(\omega)} \tag{6.67}$$

The functions r and ϕ represent the distortions of amplitude and phase due to unmodeled dynamics. It is assumed that the transfer function corresponding to r and ϕ has no poles in the right half-plane.

The unmodeled dynamics may change the properties of the system drastically. For example, the nominal system will be stable for all values of the feedback gain, since it is SPR. If the unmodeled dynamics are such that the additional phase lag can be large, the system with unmodeled dynamics will be unstable for sufficiently large feedback gains. The critical gain can be determined as follows. The phase lag of the plant is $\phi(\omega) + \arctan(\omega/a)$. This lag is π if

$$\frac{\omega}{a} = \tan(\pi - \phi(\omega)) = -\tan\phi(\omega)$$

or

$$\omega \cos\phi(\omega) + a \sin\phi(\omega) = 0 \tag{6.68}$$

The process gain of this frequency is

$$|G(i\omega)| = \frac{b\,r(\omega)}{\sqrt{a^2+\omega^2}}$$

The system thus becomes unstable for the gain

$$\bar\theta_2 = \bar\theta_2^0 = \frac{\sqrt{a^2+\omega^2}}{b\,r(\omega)} \tag{6.69}$$

where ω is the smallest value that satisfies Eq. (6.68).

Equilibrium Analysis

The possible equilibria of the parameters will first be determined. Introducing the transfer function of Eq. (6.67) into Eq. (6.59) gives (after straightforward but tedious calculations)

$$\begin{aligned}\hat\theta_1 &= \frac{b_m}{b}\cdot\frac{(a\sin\phi(\omega)+\omega\cos\phi(\omega))}{\omega r(\omega)}\\ \hat\theta_2 &= \frac{\omega(a_m-a)\cos\phi(\omega)+(\omega^2+aa_m)\sin\phi(\omega)}{\omega b r(\omega)}\\ &= \frac{1}{br(\omega)}\left((\omega\sin\phi(\omega)-a\cos\phi(\omega))+\frac{a_m}{\omega}(a\sin\phi(\omega)+\omega\cos\phi(\omega))\right)\end{aligned} \tag{6.70}$$

A comparison with the nominal case shows that the equilibrium will be shifted because of the unmodeled dynamics. The shift in the equilibrium depends on the frequency of the input signal as well as on the unmodeled dynamics.

It is of particular interest to determine whether there are conditions that may lead to difficulties. The feedforward gain vanishes for frequencies such that Eq. (6.68) is satisfied. This is precisely the frequency at which the process has a phase lag of 180°. The feedback gain for this frequency is

$$\hat\theta_2 = \frac{1}{br(\omega)}(\omega\sin\phi - a\cos\phi) = \frac{\sqrt{a^2+\omega^2}}{br(\omega)}$$

This implies that $\hat\theta_2|G(i\omega)| = 1$, that is, that the loop gain then becomes unity.

We thus find that the equilibrium values of the parameters for sinusoidal input signals will depend on the unmodeled dynamics and the frequency of the sinusoidal command signal. When the frequency is such that the plant has a phase shift of 180°, the feedforward gain is zero and the feedback gain is such that the closed-loop system is unstable. This observation is illustrated by an example.

EXAMPLE 6.12 Sinusoidal command signal

Consider the system in Example 6.10. The transfer function with the unmodeled dynamics is

$$G(s) = \frac{458}{(s+1)(s^2+30s+229)} = \frac{458}{s^3+31s^2+259s+229}$$

The equilibrium values of the controller gains are

$$\bar\theta_1 = \frac{3(259-\omega^2)}{458}$$

$$\bar\theta_2 = \frac{2(137+7\omega^2)}{229}$$

when $a_m = b_m = 3$. The transfer function G has a phase shift of 180° at $\omega = \sqrt{259} = 16.09$. At this frequency the equilibrium values of the controller gains are $\hat\theta_1 = 0$ and $\hat\theta_2 = 3900/229 = 17.03$. The closed-loop system is unstable for this feedback gain. This explains the results shown in Fig. 6.14. □

Summary of the MRAS Examples

The investigation of the first-order MRAS is summarized in the following table:

Inputs	**Exact Model Structure**	**Unmodeled Dynamics**
Step command	Equilibrium set is a half-line.	Equilibrium set is a line segment. Stability is lost for some initial values.
Step command + measurement noise	Solution will move toward a line and then drift along the line.	Solution will move toward a line and drift along the line until stability is lost.
Sinusoidal	Equilibrium set is a point that is independent of the frequency.	Equilibrium set is a point that depends on the frequency. The equilibrium is unstable for sufficiently high frequencies.

Several interesting conclusions can be drawn from the examples. When the input signal is not sufficiently exciting, the equilibrium is a manifold independently of the presence of unmodeled dynamics or disturbances. When there are disturbances, the estimates will drift along the manifold. In the case of unmodeled dynamics the closed-loop system may eventually become unstable. From a methodological point of view the examples give insights that can be derived from equilibrium analysis, which can be carried out with a moderate effort in many cases. We can find out if an equilibrium exists in the sense that the parameters remain constant. Notice that the averaged equations may have an equilibrium even if the exact equations do not. However, it is rarely the case that global analysis can be carried out.

6.8 AVERAGING IN STOCHASTIC SYSTEMS

The importance of averaging was illustrated in the previous sections. However, the excitation has been restricted to constant or sinusoidal inputs. In this section averaging is used on discrete-time systems with stochastic inputs. Assume that the system is described by

$$A^*(q^{-1})y(t) = B^*(q^{-1})u(t-d) + C^*(q^{-1})e(t) \tag{6.71}$$

where $e(t)$ is a zero-mean Gaussian stochastic process. Depending on the specifications, different self-tuning regulators can be used to control the system (compare Chapter 4). For simplicity it is assumed that the basic direct self-tuning algorithm (Algorithm 4.1) is used. The controller parameters are then estimated from a model of the form

$$y(t) = R^*(q^{-1})u(t-d) + S^*(q^{-1})y(t-d) \tag{6.72}$$

or

$$y(t) = \varphi(t-d)^T\theta \tag{6.73}$$

The parameters θ are estimated by using the recursive least-squares method. In applying averaging, it is appropriate to use the form

$$\begin{aligned}\hat{\theta}(t) &= \hat{\theta}(t-1) + \gamma(t)R(t)^{-1}\varphi(t-d)\left(y(t) - \varphi^T(t-d)\hat{\theta}(t-1)\right)\\ R(t) &= R(t-1) + \gamma(t)\left(\varphi(t-d)\varphi^T(t-d) - R(t-1)\right)\end{aligned} \tag{6.74}$$

where the covariance matrix $P(t)$ is related to $R(t)$ through

$$P(t) = \gamma(t)R(t)^{-1}$$

and $\gamma(t) = 1/t$. In some cases it is convenient to replace the matrix $R(t)$ by a scalar $r(t)$. This gives shorter computation times and requires less storage, but it gives slower convergence. For stochastic approximation we obtain

$$r(t) = r(t-1) + \gamma(t)\left(\varphi(t-d)^T\varphi(t-d) - r(t-1)\right) \tag{6.75}$$

The controller is

$$u(t) = -\frac{\hat{S}^*(q^{-1})}{\hat{R}^*(q^{-1})}y(t) \tag{6.76}$$

or

$$\varphi(t)^T\hat{\theta}(t) = 0$$

The self-tuning regulator is described by Eqs. (6.73) and (6.74). The control law of Eq. (6.76) is then used on the system of Eq. (6.71). The resulting closed-loop system is a set of nonlinear, stochastic difference equations, which can be very difficult to analyze. The difficulty arises mainly from the interplay between the estimated parameters as well as the fact that these parameters are used in the controller. By using the averaging idea it is possible to derive *associated deterministic differential equations*. The convergence properties of the algorithm can then be determined by using these equations. The method was suggested by Ljung in 1977 and is sometimes called the *ODE (ordinary differential equation) approach*. Only a heuristic derivation and motivation are given here; further details can be found in the references at the end of this chapter.

A Heuristic Derivation

For sufficiently large t the step size $\gamma(t)$ in Eqs. (6.74) is small, and the correction in $\hat{\theta}(t)$ is small. As in Section 6.6, we can separate the states from the parameters and assume that the parameters are constant in evaluating the behavior of the closed-loop system. Both $R(t)$ and $\varphi(t)$ depend on the parameter estimates. Since $\hat{\theta}$ is assumed to change slowly, the behavior of the model can be approximated by

$$y(t) = \varphi^T(t-d,\bar{\theta})\bar{\theta}$$

where $\bar{\theta}$ is the averaged value of the estimates. Also, φ depends on the estimated variables through the feedback. The updating equation for R can be approximated by

$$\bar{R}(t) = \bar{R}(t-1) + \gamma(t)\left(G(\bar{\theta}) - \bar{R}(t-1)\right) \tag{6.77}$$

where

$$G(\bar{\theta}) = E\left\{\varphi(t-d,\bar{\theta})\varphi^T(t-d,\bar{\theta})\right\} \tag{6.78}$$

The expectation is taken with respect to the underlying stochastic process in Eq. (6.71) and evaluated for the fixed value of the parameters $\bar{\theta}$. In the same way the parameter update is approximated by

$$\bar{\theta}(t) = \bar{\theta}(t-1) + \gamma(t)\bar{R}(t)^{-1}f(\bar{\theta}) \tag{6.79}$$

where

$$f(\bar{\theta}) = E\left\{\varphi(t-d,\bar{\theta})\left(y(t) - \varphi^T(t-d,\bar{\theta})\bar{\theta}\right)\right\} \tag{6.80}$$

Equations (6.79) and (6.77) are the averaged difference equations describing the estimator. Now let $\Delta\tau$ be a small number, and let t' be defined by

$$\Delta\tau = \sum_{k=t}^{t'}\gamma(k)$$

Then

$$\bar\theta(t') = \bar\theta(t) + \Delta\tau \bar R(t)^{-1} f(\bar\theta(t))$$
$$\bar R(t') = \bar R(t) + \Delta\tau \left(G(\bar\theta(t)) - \bar R(t)\right)$$

With a change of time scale such that $t = \tau$ and $t' = t + \Delta\tau$, these equations can be seen as a difference approximation of the ordinary differential equations

$$\frac{d\bar\theta}{d\tau} = \bar R(\tau)^{-1} f(\bar\theta(\tau)) \tag{6.81}$$

$$\frac{d\bar R}{d\tau} = G(\bar\theta(\tau)) - \bar R(\tau) \tag{6.82}$$

If stochastic approximation is used, Eq. (6.82) is replaced by

$$\frac{d\bar r}{d\tau} = g(\bar\theta(\tau)) - \bar r(\tau)$$

where

$$g(\bar\theta) = E\left\{\varphi^T(t-d)\varphi(t-d)\right\}$$

and $\bar R$ is replaced by $\bar r$ in Eq. (6.81). These equations are called the *associated ordinary differential equations* to Eqs. (6.74) and (6.75). They are a special kind of averaged equations. First, the difference equations are replaced by differential equations; second, there is a time scaling compared with the original system. The time scaling can be interpreted as a logarithmic compression of the original time. That is, more and more steps of length $\gamma(t)$ are needed to get the step $\Delta\tau$ as the time progresses.

The arguments leading to Eqs. (6.81) and (6.82) have been heuristic. However, it can be rigorously shown that, provided that the estimates $\hat\theta(t)$ are "sufficiently often" in the domain of attraction of the associated differential equations, then

- Only stable stationary points of Eqs. (6.81) and (6.82) are possible convergence points for the estimates.
- The trajectories $\bar\theta(\tau)$ are the "asymptotic paths" of the estimates $\hat\theta(t)$.

The associated ODE can be used to find possible convergence points of an adaptive algorithm, $\bar\theta^0$ and $\bar R^0$. The equations can then be linearized around these stationary points. It is easily seen that the linearized equations are

$$\frac{d}{dt}\begin{pmatrix}\bar\theta - \bar\theta^0 \\ \bar R - \bar R^0\end{pmatrix} = \begin{pmatrix} G(\bar\theta)^{-1}\dfrac{\partial f(\bar\theta)}{\partial\bar\theta} & 0 \\ X & -I\end{pmatrix}_{\bar\theta = \bar\theta^0}\begin{pmatrix}\bar\theta - \bar\theta^0 \\ \bar R - \bar R^0\end{pmatrix}$$

where the element X is not important for the local stability. The stationary point is thus stable if the matrix

$$K = G(\bar\theta)^{-1}\frac{\partial f(\bar\theta)}{\partial\bar\theta}\bigg|_{\bar\theta = \bar\theta^0} \tag{6.83}$$

has all its eigenvalues in the left half-plane. The associated ODEs can thus be used in the following way:

1. Compute the expressions for $\varphi(t)$ and $\varepsilon(t) = y(t) - \varphi(t-d)^T\bar\theta$ for a fixed value of $\bar\theta$.
2. Compute the expected values $G(\bar\theta)$ and $f(\bar\theta)$.
3. Determine possible convergence points for Eqs. (6.81) and (6.82), and determine the local stability properties by using Eq. (6.83).
4. Simulate the equations.

Even if Eqs. (6.81) and (6.82) can be quite difficult to analyze in detail, it is usually easy to determine the possible stationary points. The equations can also be simulated to obtain a feel for the behavior of the convergence properties. The change in the time scale makes it more favorable to simulate the ODEs than the averaged difference equations.

Stability of Stochastic Self-Tuners

Averaging methods can be used for stability analysis of stochastic self-tuning regulators. Consider a simple self-tuner based on least-squares estimation and minimum-variance control (Algorithm 4.1 with $Q^* = P^* = 1$). Let the algorithm be applied to a system described by Eq. (6.71). The self-tuner is assumed to be compatible with the model in the sense that the time delay and the model orders are the same. The closed-loop system is globally stable if the pulse transfer function

$$G(z) = \frac{1}{C(z)} - \frac{1}{2}$$

is SPR (see Ljung (1977b)). The filter P^* that is used to filter the regressors can be interpreted as an estimate of the observer polynomial C^*. The condition for global stability is then that the transfer function

$$G(z) = \frac{P(z)}{C(z)} - \frac{1}{2}$$

is SPR. The local stability condition is that the real part of polynomial $C(z)$ is positive at all zeros of the polynomial $B(z)$ (see Holst (1979)). The method with stochastic averaging is illustrated with three examples.

EXAMPLE 6.13 Stochastic averaging

Consider the system

$$y(t) + ay(t-1) = u(t-1) + bu(t-2) + e(t) + ce(t-1)$$

with $a = -0.99$, $b = 0.5$, and $c = -0.7$. Let the estimated model be

$$y(t) = u(t-1) + r_1 u(t-2) + s_0 y(t-1)$$

and use the controller

$$u(t) = -s_0 y(t) - r_1 u(t-1)$$