To

Bia, Emma, Ida, Johanna, Kalle, Kalle, Karin, and Karin
Adaptive control is a fascinating field for research. It is also of increasing practical importance, since adaptive techniques are being used more and more in industrial control systems. However, the field is not mature; there are still many unsolved theoretical and practical issues. In spite of this, we believe that the book will serve a purpose. Since knowledge about adaptive techniques is widely scattered in the literature, it is difficult for a newcomer to get a good grasp of the field. This book has evolved from many years of research and teaching in adaptive control.

Our intent has been to write a book that can serve as an introduction. In the presentation it is assumed that the reader already has good knowledge in automatic control and a basic knowledge in sampled data systems. At our university the course can be taken after an introductory course in feedback control and a course in digital control. The intent is also that the book should be useful for an industrial audience.
The book is organized in the following way. The first two chapters give a broad presentation of adaptive control and background for its use. Real-time estimation, which is an essential part of adaptive control, is introduced in Chapter 3. Both discrete-time and continuous-time estimation are covered. Chapters 4 and 5 give two basic developments of adaptive control: model-reference adaptive systems (MRAS) and self-tuning regulators (STR). Today we do not make a distinction between these two approaches, since they are actually equivalent. We have tried to follow the historical development by treating MRAS in continuous time and STR in discrete time. By doing so it is possible to cover many aspects of adaptive regulators. These chapters mainly cover the ideas and basic properties of the controllers. They also serve as a source of algorithms for adaptive control.

Chapter 6 gives deeper coverage of the theory of adaptive control. Questions such as stability, convergence, and robustness are discussed. Stochastic adaptive control is treated in Chapter 7. Depending on the background of the students, some of the material in Chapters 6 and 7 can be omitted in an introductory course. Automatic tuning of regulators, which is rapidly gaining industrial acceptance, is presented in Chapter 8.

Even though adaptive controllers are very useful tools, they are not the only ways to deal with systems that have varying parameters. Since we believe that it is useful for an engineer to have several ways of solving a problem, two chapters with alternatives to adaptive control are also included. Gain scheduling is discussed in Chapter 9, and robust high-gain control and self-oscillating controllers are presented in Chapter 10.

Chapter 11 gives suggestions for the implementation of adaptive controllers. The guidelines are based on practical experience in using adaptive controllers on real processes. Chapter 12 is a summary of applications and description of some commercial adaptive controllers. The applications show that adaptive control can be used in many different types of processes, but also that all applications have special features that must be considered to obtain a good control system. Finally, Chapter 13 contains a brief review of some areas closely related to adaptive control that we have not been able to cover in the book. Connections to adaptive signal processing, expert systems, and neural networks are given. Many examples and simulations are given throughout the book to illustrate ideas and theory.

The book can be used in many different ways. An introductory course in adaptive control could cover Chapters 1, 2, 3, 4, 5, 8, 11, 12, and 13. A more advanced course might include all chapters in the book. A course for an industrial audience could contain Chapters 1 and 2, parts of Chapters 3, 4, and 5, and Chapters 8, 9, 11, and 12. To get the full benefit of a course, it is important to supplement lectures with problem-solving sessions and laboratory experiments. A good simulation package is also indispensable. All the simulations in the book are done using the
interactive simulation package Simnon, which has been developed at Lund Institute of Technology. It is available for IBM-PC compatible computers and also for several mainframe computers. Further information can be obtained from the authors at the address given below.

As teachers and researchers in automatic control, we know the importance of feedback. We therefore encourage all readers to write to us about errors, misunderstandings, suggestions for improvements, and also about what may be valuable in the material we have presented.

Acknowledgments

During the years we have done research in adaptive control and written the book, we have had the pleasure and privilege of interacting with many colleagues throughout the world. Consciously and subconsciously, we have picked up material from the knowledge base called adaptive control. It is impossible to mention everyone who has contributed ideas, suggestions, concepts, and examples, but we owe you all our deepest thanks. The long-term support of our research on adaptive control by the Swedish Board of Technical Development (STU) is greatly acknowledged.

We want to express our gratitude to those who have read different versions of the manuscript and thereby removed many of the errors and given us good suggestions for improvements: Gunnar Bengtsson, Bo Bernhardsson, Hang Chang Chien, Guy Dumont, Bo Egardt, Gene Franklin, David Hill, Tore Hägglund, Rick Johnson, Bob Kosut, Panganamala Kumar, Bengt Lennartson, Bill Levine, Michael Lundh, Charles Neuman, John MacGregor, Per-Olof Olsson, Per Persson, Lars Rundqvist, Shankar Sastry, Jan Sternby, Torsten Söderström, Lena Valavani, and Vincent Wertz.

Finally, we want to thank some people who, more than others, have made it possible for us to write this book. Leif Andersson has been our \TeX{}Xpert. He and Eva Dagnegård have been invaluable when solving many of the \TeX{} problems. Eva Dagnegård and Agneta Tuszyński have done an excellent job of typing many versions of the manuscript. Most of the illustrations have been done by Britt-Marie Carlsson and Doris Nilsson. Without all their patience and understanding for our whims, there would never have been a final book. We also want to thank Tom Robbins, Karen Myer, and their colleagues at Addison-Wesley for their support and professionalism in bookmaking, which we have come to appreciate more and more in our attempts to make a final \TeX{} manuscript of the book.

Department of Automatic Control
Lund Institute of Technology
Box 118, S-221 00 Lund, Sweden

Karl Johan Åström
Björn Wittenmark
# CONTENTS

1. **What Is Adaptive Control?** ........................................... 1  
   1.1 Introduction .................................................... 1  
   1.2 Adaptive Schemes ............................................. 3  
   1.3 Adaptive Control Theory .................................... 14  
   1.4 Applications .................................................. 21  
   1.5 Conclusions ................................................... 25  
   Problems ........................................................... 26  
   References ........................................................ 26

2. **Why Adaptive Control?** ........................................... 29  
   2.1 Introduction .................................................... 29  
   2.2 When Is Constant-gain Feedback Insufficient? .......... 30  
   2.3 Robust Control ............................................... 42  
   2.4 The Adaptive Control Problem ............................ 49  
   2.5 Conclusions ................................................... 52
## Contents

Problems ............................................. 52  
References .......................................... 56  

### 3. Real-time Parameter Estimation ............................................. 58  
3.1 Introduction ........................................ 58  
3.2 Least Squares and Regression Models ......................... 59  
3.3 Estimating Parameters in Dynamical Systems ................. 72  
3.4 Experimental Conditions ...................................... 79  
3.5 Properties of Recursive Estimators ......................... 85  
3.6 Implementation Issues .................................... 89  
3.7 Conclusions .......................................... 99  
Problems ............................................. 99  
References .......................................... 103  

### 4. Model-reference Adaptive Systems .................................... 105  
4.1 Introduction .......................................... 105  
4.2 The MRAS Problem ....................................... 107  
4.3 The Gradient Approach .................................... 110  
4.4 MRAS Based on Stability Theory ............................ 124  
4.5 Direct MRAS for General Linear Systems ..................... 139  
4.6 MRAS for Partially Known Systems .......................... 151  
4.7 Conclusions .......................................... 155  
Problems ............................................. 156  
References .......................................... 159  

### 5. Self-tuning Regulators ............................................. 163  
5.1 The Basic Idea .......................................... 163  
5.2 Indirect Self-tuning Regulators .............................. 166  
5.3 Direct Self-tuning Regulators ................................ 173  
5.4 Unification of Direct Self-tuning Regulators ................. 186  
5.5 Linear Quadratic STRs .................................... 194  
5.6 Adaptive Predictive Control ................................ 198  
5.7 A Priori Knowledge in STR ................................ 208  
5.8 Conclusions .......................................... 211  
Problems ............................................. 211  
References .......................................... 213  

### 6. Stability, Convergence, and Robustness ......................... 219  
6.1 Introduction .......................................... 219  
6.2 Global Stability ....................................... 220  
6.3 Convergence .......................................... 229  
6.4 Averaging ........................................... 235  
6.5 An Example of Averaging Analysis ............................ 239  
6.6 Robustness ........................................... 247  
6.7 Stochastic Averaging .................................... 267
11. Practical Aspects and Implementation ........................................... 402
  11.1 Introduction ................................................................. 402
  11.2 Estimator Implementation ................................................ 403
  11.3 Controller Implementation ................................................ 415
  11.4 Estimation and Control Interaction ....................................... 423
  11.5 Prototype Algorithms ...................................................... 429
  11.6 Conclusions ................................................................. 435
Problems ....................................................................................... 435
References .................................................................................... 437

12. Applications ............................................................................ 439
  12.1 Introduction ................................................................. 439
  12.2 Status of Applications ....................................................... 440
  12.3 Industrial Adaptive Controllers ............................................ 443
  12.4 Some Novatune Applications ............................................... 458
  12.5 A Ship Steering Autopilot .................................................... 463
  12.6 Ultrafiltration ................................................................. 468
  12.7 Conclusions ................................................................. 475
References .................................................................................... 476

13. Perspectives on Adaptive Control ............................................. 478
  13.1 Introduction ................................................................. 478
  13.2 Adaptive Signal Processing .................................................. 479
  13.3 Extremum Control ............................................................ 484
  13.4 Expert Control ............................................................... 487
  13.5 Learning Systems ............................................................. 490
  13.6 Future Trends ................................................................. 493
  13.7 Conclusions ................................................................. 495
References .................................................................................... 496

A. Regulator Design ....................................................................... 499
  Pole Placement Design ............................................................ 499
  The Diophantine Equation .......................................................... 508
References .................................................................................... 511

B. Positive Real Transfer Functions ............................................ 513
  References .................................................................................. 516

Index ............................................................................................. 517
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 regulator is a regulator that can modify its behavior in response to changes in the dynamics of the process and the disturbances. Since ordinary feedback has been introduced for the same purpose, 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. At an early symposium in 1961 a long discussion ended with the following suggested definition: “An adaptive system is any physical system that has been designed with an adaptive viewpoint.” A new 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. These efforts, however, were not widely accepted. A meaningful definition of adaptive control, which would make it possible to look at a regulator hardware and software and decide if it is adaptive or not, is still lacking. There appears, however, to be a consensus that a constant-gain feedback is not an adaptive system.

In this book we will take the pragmatic attitude that adaptive control is a special type of nonlinear feedback control in which the states of the process can be separated into two categories, which change at different rates. The slowly changing states are viewed as parameters. This introduces the idea of two time scales: a fast time scale for the ordinary feedback and a slower one for updating the regulator parameters. This also implies that linear constant parameter regulators are not adaptive. In an adaptive controller we also assume that there is some kind of feedback from the performance of the closed-loop system. This implies that gain scheduling should not be regarded as an adaptive controller, since the parameters are determined by a schedule, without any feedback from the performance.

The reason why a control engineer should know about adaptive systems is that they have useful capabilities and interesting properties, which can be profitably incorporated in the design of new control systems.

**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. 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 that changed operating conditions led to difficulties. A more sophisticated regulator, which could work well over a wide range of operating conditions, was therefore needed. Interest in the subject diminished due to lack of insight and a disaster in a flight test.

In the 1960s many contributions to control theory were important for 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 as recursive equations of a special type. There were also major developments in system identification and in parameter estimation. There was a renaissance of adaptive control 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 correct proofs for stability of adaptive
systems appeared, albeit under very restrictive assumptions. Investigation of the necessity of those assumptions has sparked new and interesting research into the robustness of adaptive control, as well as into controllers that are universally stabilizing.

Rapid and revolutionary progress in microelectronics has made it possible to implement adaptive regulators simply and cheaply. Vigorous development of the field is now taking place, both at universities and in industry. Several commercial adaptive regulators based on different ideas are appearing on the market, and the industrial use of adaptive control is growing slowly but surely. Additional historical remarks are given in Section 1.2.

Relations to Other Areas of Automatic Control

Adaptive control is by no means a mature field. Many of the algorithms and approaches used are of an ad hoc nature; the tools are gathered from a wide range of fields; and good systematic approaches are still lacking. Yet the algorithms and systems have found good uses, and there are adaptive systems that clearly outperform conventional feedback systems.

To pursue adaptive control one must have a background in conventional feedback control and also sampled data systems. The reason for the latter is that virtually all adaptive systems are implemented using digital computers. Adaptive control has links in many directions, some of which are illustrated in Fig. 1.1. There are strong ties to nonlinear system theory, because adaptive systems are inherently nonlinear. Stability theory is a key element. Adaptive control also has connections to singular perturbations and averaging theory, because of the separation of time scales in adaptive systems. There are also links to stochastic control and parameter estimation, because one way to look at adaptive systems is to view them as a combination of parameter estimation and control.

Organization of This Chapter

Some adaptive systems are presented in Section 1.2 to give a feel for what they look like. Section 1.3 treats key theoretical issues, and some uses of adaptation are presented in Section 1.4. The idea of this chapter is to give a quick overview; the detailed discussions of these issues are given in later chapters.

1.2 Adaptive Schemes

A constant-gain robust regulator is used as a point of departure for the discussion in this section. This system includes a model of the desired performance. We then treat different heuristic adaptive systems, such as
the self-oscillating adaptive controller, gain scheduling, model-reference control, and self-tuning regulators. Finally, we discuss the systems that arise from stochastic control theory. Such systems are of interest even if they cannot be easily realized. They allow fast adaptation and indicate the need for the new functions of caution and probing, which are not present in the previously discussed approaches based on heuristic ideas.

**Robust High-gain Control**

We will start by discussing a robust high-gain regulator, which is a constant-gain regulator designed to cope with parameter variations. A block diagram of such a regulator is shown in Fig. 1.2. There is a high-gain feedback around the plant, making the output  follow the reference signal  over a wide bandwidth . The bandwidth will change with the process dynamics. There is also a model that gives the desired per-
formance in the feedforward path. If the bandwidth of the model is lower than $\omega_B$, the output $y$ will respond to the command signal $u_c$, as specified by the model, even if the process dynamics change. The key problem is to design the feedback loop and the feedforward model so that stability and performance are maintained in spite of process variations. The structure in Fig. 1.2 is called a two-degree-of-freedom system. This type of design is useful when the uncertainties are unstructured and known in advance. Robust high-gain control is discussed in Chapters 2 and 10.

**Self-oscillating Adaptive Systems (SOASs)**

The self-oscillating adaptive system has the same structure as the high-gain system. The bandwidth of the feedback loop is, however, automatically adjusted to be as high as possible. A block diagram of such a system is shown in Fig. 1.3. The high loop gain is maintained by introducing a relay in the feedback loop. This creates a limit cycle oscillation. It can be shown that, for signals whose frequencies are much lower than the limit cycle oscillation, the amplitude margin is approximately equal to 2. The system with relay feedback thus automatically adjusts itself to give a reasonable amplitude margin.

Notice that the system will always be excited because of the limit cycle oscillation. The frequency of this oscillation can be influenced by the lead-lag filter shown in Fig. 1.3. The amplitude of the oscillation can be adjusted by changing the relay amplitude. The limit cycle oscillation is
sometimes acceptable, but for piloted aircraft it has always been subject to much discussion. Experiments have indicated that the pilots will always notice the oscillation.

There are many variations of the basic self-oscillating adaptive control system. Some have attempted to adjust the limit cycle amplitude by feedback. However, if the relay amplitude is too small, the response to command signals will be too slow. There have also been attempts to quench the relay oscillations by a dither signal. SOASs are discussed in Chapter 10.

Gain Scheduling

In some systems there are auxiliary variables that relate well to the characteristics of the process dynamics. If these variables can be measured, they can be used to change the regulator parameters. This approach is called gain scheduling because the scheme was originally used to accommodate changes in process gain. A block diagram of a system with gain scheduling is shown in Fig. 1.4.

Gain scheduling is an open-loop compensation and can be viewed as a system with feedback control in which the feedback gains are adjusted by feedforward compensation. There is no feedback from the performance of the closed-loop system, which compensates for an incorrect schedule. The concept of gain scheduling originated in connection with the development of flight control systems. In this application the Mach number and the dynamic pressure are measured by air data sensors and used as scheduling variables. 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. With regard to nomenclature, it is controversial whether gain scheduling should be considered as an adaptive system or not, because the parameters are changed in open loop. Nevertheless, gain scheduling is a very useful technique for reducing the
effects of parameter variations. Details and applications of gain scheduling are given in Chapter 9.

Auto-tuning

Simple PID controllers are adequate in many applications. Such controllers are traditionally tuned using simple experiments and simple empirical rules. Many adaptive techniques can be applied to tune PID controllers, and special methods for automatic tuning of such regulators have also been proposed. This is discussed in Chapter 8.

Tuning is usually based on an experimental phase in which test signals such as steps or pulses are injected. Natural disturbances can also be used. The regulator parameters can be determined from the experiments using standard rules for tuning PID controllers.

A special class of auto-tuners is obtained by using relay feedback in the experiment. Information about the process dynamics is then derived from the limit cycle oscillation obtained. This type of auto-tuner is thus related to the SOAS.

One advantage with auto-tuners is that the tuning experiment is initiated and can be supervised by an operator. The “safety nets” required can then be simpler than when the parameters are adapted continuously.

Model-reference Adaptive Systems (MRAS)

The model-reference adaptive system was originally proposed to solve a problem in which the specifications are given in terms of a reference model that tells how the process output ideally should respond to the command signal. A block diagram of the system is shown in Fig. 1.5. In this case the
reference model is in parallel with the system rather than in series, as for the SOAS. The regulator can be thought of as consisting of two loops: an inner loop, which is an ordinary feedback loop composed of the process, and the regulator. The parameters of the regulator are adjusted by the outer loop in such a way that the error $e$ between the process output $y$ and the model output $y_m$ becomes small. The outer loop is thus also a regulator loop. The key problem 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:

$$\frac{d\theta}{dt} = -\gamma e \frac{\partial e}{\partial \theta}$$  \hspace{1cm} (1.1)

In this equation $e$ denotes the model error. The components of the vector $\frac{\partial e}{\partial \theta}$ are the sensitivity derivatives of the error with respect to the adjustable parameters $\theta$. Approximations of the sensitivity derivatives can be generated as outputs of a linear system driven by process inputs and outputs. The parameter $\gamma$ determines the adaptation rate.

The MIT rule can be explained as follows. Assume that the parameters $\theta$ change much slower than the other system variables. To make the square of the error small, it seems reasonable to change the parameters in the direction of the negative gradient of $e^2$.

The parameter adjustment mechanism described by Eq. (1.1) can be regarded as composed of a linear filter for computing the sensitivity derivatives from process inputs and outputs, a multiplier, and an integrator. The parameters are then introduced in the control law using a second multiplier. This is a generic part of many adaptive schemes. In Fig. 1.6 a redrawn block diagram emphasizes this structure, which is called the error model. Notice that the MRAS attempts to adjust the parameters so that the correlation between the error $e$ and the sensitivity derivatives becomes zero. A simple example illustrates model-reference adaptive control.

**Example 1.1—Adaptation of a feedforward gain**

Consider the problem of adjusting a feedforward gain. Let the model have the transfer function $G_m(s) = \theta^0 G(s)$ where $\theta^0$ is a known constant.
The process is assumed to have the transfer function $G(s)$, which is also assumed to be known. The error is

$$e = y - y_m = G(p)\theta u_c - G_m(p)u_c = G(p)(\theta - \theta^0)u_c$$

where $u_c$ is the command signal, $y_m$ the model output, $y$ the process output, $\theta$ the adjustable parameter, and $p = d/dt$ the differentiation operator. The sensitivity derivative is

$$\frac{\partial e}{\partial \theta} = G(p)u_c = y_m/\theta^0$$

The MIT rule (Eq. 1.1) gives

$$\frac{d\theta}{dt} = -\gamma ey_m$$

where $\theta^0$ has been included in $\gamma$. The rate of change of the parameter should thus be made proportional to the product of the error and the model output. A block diagram of the model-reference adaptive system is shown in Fig. 1.7.

Notice that no approximations were needed in Example 1.1. When the MIT rule is applied to more complicated problems, it is necessary to use approximations to obtain the sensitivity derivatives. A detailed description and analysis of MRAS are given in Chapter 4.
Real-time Estimation

A real-time estimator is a central part in most adaptive controllers. A very simple estimator is given in Eq. (1.1), where one parameter is continuously adjusted following the negative gradient of $e^2$.

There are many alternative ways to make real-time estimation, both in continuous and discrete time. Different approaches and their properties are discussed in Chapter 3. The least-squares estimation method and its variants play a fundamental role in the construction of adaptive controllers. The method may be illustrated by a very simple example.

Example 1.2—Least-squares estimation

Assume that a process is described by the difference equation

$$y(t + 1) = \theta^0 y(t) + u(t)$$  \hspace{1cm} (1.2)

where $\theta^0$ is an unknown parameter. Consider the model

$$\hat{y}(t + 1) = \hat{\theta} y(t) + u(t)$$  \hspace{1cm} (1.3)

where $\hat{\theta}$ is the estimate of $\theta^0$ and $\hat{y}(t + 1)$ is the prediction or assumed value of the output at time $t + 1$ based on the estimate $\hat{\theta}$. Define the least-squares loss function

$$V(t) = \frac{1}{2} \sum_{k=0}^{t} e(k)^2$$  \hspace{1cm} (1.4)

with

$$e(t) = y(t) - \hat{y}(t)$$
$$= \theta^0 y(t - 1) - \hat{\theta} y(t - 1)$$
$$= y(t) - u(t - 1) - \hat{\theta} y(t - 1)$$

where Eq. (1.2) has been used to eliminate $\theta^0$. Differentiation of Eq. (1.4) with respect to $\hat{\theta}$ gives the least-squares estimate

$$\hat{\theta}(t) = \frac{\sum_{k=0}^{t-1} y(k)(y(k + 1) - u(k))}{\sum_{k=0}^{t-1} y^2(k)}$$  \hspace{1cm} (1.5)

The time index is introduced to indicate that the estimate is based on data up to and including time $t$. The estimate of Eq. (1.5) minimizes Eq. (1.4), under the assumption that the process is described by Eq. (1.2).
1.2 Adaptive Schemes

Self-tuning Regulators (STRs)

The schemes discussed so far are called direct methods, because the adjustment rules tell directly how the regulator parameters should be updated. A different scheme is obtained if the process parameters are updated and the regulator parameters are obtained from the solution of a design problem. A block diagram of such a system is shown in Fig. 1.8. The adaptive regulator can be thought of as composed of two loops. The inner loop consists of the process and an ordinary linear feedback regulator. The parameters of the regulator are adjusted by the outer loop, which is composed of a recursive parameter estimator and a design calculation. To obtain good estimates, it may also be necessary to introduce perturbation signals. This function is not shown in Fig. 1.8, so as to keep the figure simple. 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 Chapter 5.

The block labeled “Design” in Fig. 1.8 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 explored. The regulator parameters are updated indirectly via the design calculations in the self-tuner shown in Fig. 1.8. It is sometimes possible to reparameterize the process so that the model can be expressed in terms of the regulator parameters. This gives a significant simplification of the algorithm, because the design calculations are eliminated. In terms of Fig. 1.8 the block labeled “Design” disappears, and the regulator parameters are updated directly. An example illustrates the idea.

**Example 1.3—Simple direct self-tuning regulator**
Consider the discrete time system described by

\[ y(t + 1) + ay(t) = bu(t) + e(t + 1) + ce(t) \]  

(1.6)

where \( \{e(t)\} \) is a sequence of zero-mean uncorrelated random variables. If the parameters \( a, b, \) and \( c \) are known, the proportional feedback

\[ u(t) = -\theta y(t) = -\frac{c-a}{b} y(t) \]  

(1.7)

minimizes the variance of the output. The output then becomes

\[ y(t) = e(t) \]  

(1.8)

This can be concluded from the following argument. Consider the situation at time \( t \). The random variable \( e(t + 1) \) is independent of \( y(t), u(t), \) and \( e(t) \). The output \( y(t) \) is known, and the signal \( u(t) \) is at our disposal. The variable \( e(t) \) can be computed from past inputs and outputs. Choosing the variable \( u(t) \) such that the terms underlined in Eq. (1.6) vanish makes the variance of \( y(t + 1) \) as small as possible. This gives Eq. (1.8) and Eq. (1.7).

Since the process in Eq. (1.6) is characterized by three parameters, a straightforward application of a self-tuner would require estimation of three parameters. Notice, however, that the feedback law is characterized by one parameter only, i.e., \( \theta = (c-a)/b \). A self-tuner that estimates this parameter can be obtained based on the model

\[ y(t + 1) = \theta y(t) + u(t) \]  

(1.9)

The least-squares estimate of the parameter \( \theta \) in this model is given by Eq. (1.5), and the control law is then given by

\[ u(t) = -\hat{\theta}(t)y(t) \]  

(1.10)

The self-tuning regulator given by Eq. (1.5) and Eq. (1.10) has some remarkable properties, which can be seen heuristically as follows. Equation
(1.5) can be written as

\[
\frac{1}{t} \sum_{k=0}^{t-1} y(k + 1)y(k) = \frac{1}{t} \sum_{k=0}^{t-1} \left( \hat{\theta}(t)y^2(k) + u(k)y(k) \right)
\]

\[
= \frac{1}{t} \sum_{k=0}^{t-1} \left( \hat{\theta}(t) - \hat{\theta}(k) \right)y^2(k)
\]

Assuming that \( y \) is mean square bounded and that the estimate \( \hat{\theta}(t) \) converges as \( t \to \infty \), we get

\[
\hat{r}_y(1) = \lim_{t \to \infty} \frac{1}{t} \sum_{k=0}^{t-1} y(k + 1)y(k) = 0 \quad (1.11)
\]

The adaptive algorithm Eqs. (1.5) and (1.10) thus attempts to adjust the parameter \( \hat{\theta} \) so that the correlation of the output at lag one, \( \hat{r}_y(1) \), is zero. If the system to be controlled is actually governed by Eq. (1.6), the estimate will converge to the minimum-variance control law under the given assumption. This is somewhat surprising, because the structure of Eq. (1.9), which was the basis of the adaptive regulator, is not compatible with the true system in Eq. (1.6).

### Relations between MRAS and STR

The MRAS originated from a deterministic servo problem and the STR from a stochastic regulation problem. In spite of their different origins it is clear from Fig. 1.5 and Fig. 1.8 that they are closely related. Both systems have two feedback loops. The inner loop is an ordinary feedback loop with a process and a regulator. The regulator has adjustable parameters, which are set by the outer loop. The adjustments are based on feedback from the process inputs and outputs. However, the methods for design of the inner loop and the techniques used to adjust the parameters in the outer loop are different. The regulator parameters are updated directly in the MRAS in Fig. 1.5. In the STR in Fig. 1.8 they are updated indirectly via parameter estimation and design calculations. This difference is, however, not fundamental, because the STR may be modified so that the regulator parameters are updated directly, as was shown in Example 1.3.

### Direct and Indirect Adaptive Controllers

In the discussion of model-reference adaptive systems and self-tuning regulators we have pointed out that there is a choice of what to estimate. In
direct algorithms the controller parameters are updated directly. If the controller parameters are obtained indirectly via a design procedure (compare Fig. 1.8), we use the term indirect algorithms. It is, however, sometimes possible to reparameterize the process model such that it is possible to use either a direct or an indirect controller (see Example 1.3). The advantages and disadvantages of the two approaches are discussed, in connection with self-tuning regulators, for instance, in Chapter 5.

There has been some confusion in the nomenclature. In the self-tuning context the indirect methods have sometimes been called explicit self-tuning control, since the process parameters have been estimated. Direct updating of the regulator parameters has been called implicit self-tuning control. In this book we will use only the terms direct and indirect methods. (This is a penance, since the authors have in earlier papers contributed to the confusion.)

The Certainty Equivalence Principle

In MRAS and 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 one (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 to modify the controller. Different ways to do this are discussed in Section 1.3 and Chapter 7.

1.3 Adaptive Control Theory

Adaptive systems are inherently nonlinear. Their behavior is therefore quite complex, which makes them difficult to analyze. Progress in theory has been slow, and much work remains before a reasonably complete, coherent theory is available. Because of the complex behavior of adaptive systems, it is necessary to consider them from several points of view. Theories of nonlinear systems, stability, system identification, recursive parameter estimation, optimal control, and stochastic control all contribute towards the understanding of adaptive systems. This section gives an overview of some key issues and some tools that are useful for analysis and design of adaptive systems. The theoretical analysis of adaptive controllers is given in Chapter 6.

Why Are Adaptive Systems Nonlinear?

It is of interest to find the simplest possible parameter adjustment rules that can be used in an adaptive system. It is natural to ask if a linear
error feedback can be used in the outer loop. A simple counterexample shows that this is not possible.

**Example 1.4—Adaptive systems are nonlinear**
Consider the simple MRAS in Fig. 1.7, where the problem is to adjust the gain only. Assume that $G(s) = 1$. The error is then given by

$$e = \theta u_c - \theta^0 u_c = u_c(\theta - \theta^0)$$

If the rate of change of the parameter $\theta$ is made proportional to the error, we get

$$\frac{d\theta}{dt} = -\gamma u_c (\theta - \theta^0)$$

This differential equation has the unique equilibrium solution $\theta = \theta^0$. However, this solution is stable only if $\gamma \cdot u_c > 0$. In order to have a stable equilibrium, $\gamma$ must have the same sign as $u_c$. It is thus not possible to obtain an adjustment rule that gives a stable solution with a linear error feedback having constant gain. The adjustment rules

$$\frac{d\theta}{dt} = -\gamma e u_c$$

$$\frac{d\theta}{dt} = -\gamma e \text{ sign } u_c$$

$$\frac{d\theta}{dt} = -\gamma \frac{e}{u_c}$$

will all give a stable equilibrium. These rules are, however, all nonlinear.

The fact that adaptive systems are nonlinear has some far-reaching consequences. For example, the stability concepts for nonlinear differential equations refer to stability of a particular solution. It can thus happen that one solution is stable and another one unstable. Hence it is only in exceptional cases that we can speak of a stable system. Since it is often difficult to obtain global results for nonlinear problems, we often have to be satisfied with local results. This means that the equilibrium solutions are first determined, then the local behavior is obtained by linearization.

**Key Issues**
Adaptive system theory has several goals. It is desirable to have tools to analyze a given system as well as tools for design. A typical problem is to design a parameter adjustment rule that is guaranteed to result in a stable closed loop system. As was mentioned in the introduction, many
theoretical tools can be used for these purposes. Some ideas that have proven useful are drawn from the fields of stability, parameter estimation, and nonlinear systems.

**Adaptation and Tuning**

It is customary to separate the tuning and the adaptation problems. In the tuning problem it is assumed that the process to be controlled has constant but unknown parameters; in the adaptation problem it is assumed that the parameters are changing. Many issues are much easier to handle in the tuning problem. The convergence problem is to investigate whether the parameters converge to their true values. The corresponding problem is much more difficult in the adaptive case, because the true values are changing. The estimation algorithms are similar for tuning and adaptation. A typical algorithm is

\[ \hat{\theta}(t+1) = \hat{\theta}(t) + P(t)\varphi(t) \left( y(t+1) - \varphi^T(t)\hat{\theta}(t) \right) \]  

where \( \hat{\theta} \) is the estimate of the parameter vector, \( \varphi \) a vector of regressors (which are functions of measured signals in the system), \( y \) is the measurement signal, and \( P \) a gain matrix (which is also governed by a difference equation). The gain matrix \( P \) behaves very differently in tuning and adaptation. It goes to zero in the tuning case as \( t \) increases, but it is not allowed to converge to zero in the adaptation case.

**Stability**

Stability is a basic requirement in a control system. Much effort has been devoted to analysis of stability of adaptive systems. Stability theory has been the major source of inspiration for the development of model-reference adaptive systems. It has, however, not been applied to systems with gain scheduling. This is surprising, since such systems are much simpler than MRAS. Much of the work on stability has centered around the error model shown in Fig. 1.6. A system that can be represented in this form can be regarded as composed of a linear system with a nonlinear feedback. This is a classical configuration, for which stability results are available. A key result is that the closed-loop system is stable if the linear part is strictly positive real (SPR) and the nonlinear part is passive.

The error model can be used both for analysis and design. If the transfer function \( G(s) \) of the linear part is not SPR, we can attempt to filter the error \( e \) with a linear system \( G_c \) such that the combination \( GG_c \) is SPR. Several improvements of the model-reference adaptive system were derived from this type of argument. To obtain the error model shown in Fig. 1.6 it is necessary to parameterize the model so that it is linear in the
parameters. The model should thus be of the form

\[ y(t) = \varphi^T(t)\theta \]

where \( \theta \) is a parameter vector and \( \varphi \) is a vector of signals that can be generated from measured system variables. This requirement strongly limits the algorithms that can be considered.

Conditions for global asymptotic stability have been derived based on the error model. An interesting feature of these results is that they apply for arbitrary values of the adaptation gain, such as \( \gamma \) in Eq. (1.1), i.e., for arbitrarily fast adaptation. However, very strong assumptions are required for the results to hold; minor changes in the process may invalidate them.

Stability theory gives only a crude way of estimating the effects of disturbances, i.e., that a system that is uniformly asymptotically stable can withstand disturbances.

### Parameter Convergence

The theory of parameter estimation gives another viewpoint on adaptive systems. This approach is natural for self-tuning regulators. The approach taken is then to explore the conditions under which the recursive parameter estimator built into the adaptive system will work. This approach leads naturally to conditions (persistent excitation or sufficient richness) that ensure that the input signal changes sufficiently. This is easy to understand intuitively, because it is impossible to determine process dynamics from an experiment in which the input to the process is zero. An example illustrates what may happen.

#### Example 1.5—Parameter convergence

Consider the simple adaptive system in Example 1.1. Let \( G(s) = 1 \). The parameter adjustment law then becomes

\[ \frac{d\theta}{dt} = -\gamma u_c^2(\theta - \theta^0) \]

This equation has the solution

\[ \theta(t) = \theta^0 + (\theta(0) - \theta^0)e^{-\gamma I_t} \]

where

\[ I_t = \int_0^t u_c^2(\tau) \, d\tau \]

How the estimate \( \theta \) behaves for large \( t \) depends critically on the integral \( I_t \). The parameter converges to the correct value \( \theta = \theta^0 \) if the integral diverges. If the integral converges to \( I_{\infty} \), the parameter will instead converge
to the value
\[ \theta(\infty) = \theta^0 + (\theta(0) - \theta_0) e^{-\gamma T}\]

Notice that the limit value depends on the initial condition and the input \( u_c \) and that it is different from the desired equilibrium value \( \theta = \theta^0 \).

The reason why the parameter in Example 1.5 does not converge to the desired equilibrium point is that the input signal \( u_c \) does not excite the system sufficiently. Conditions that guarantee that there is enough variation in the input to determine the parameters are thus important. The notions of sufficient richness and persistent excitation have been developed to cover this.

The parameter estimation approach also leads to questions concerning identifiability of systems under closed loop conditions.

**Slow Adaptation and Averaging Methods**

Many adaptive algorithms are based on the assumption that the parameters change more slowly than the state variables of the system. The rate of change of the parameters can be controlled by the designer's choice of the adaptation rate, e.g., the parameter \( \gamma \) in Eq. (1.1). It is natural to try to use this property to simplify the analysis by treating the parameters and the states separately. This means that the parameters are viewed as constants when investigating the behavior of the states. When analyzing the evolution of the estimated parameters in Eq. (1.12), for example, the term \( P \varphi(y - \varphi^T \theta) \) is simply replaced by its mean value. This type of approximation, which is called *averaging theory*, is a standard tool of applied mathematics. Formal justification of the averaging techniques has been established when the rapidly varying signals are periodic or almost periodic. The averaging methods will work better the smaller the adaptation gain is. In many cases it can be shown that the difference between the true and the averaged equations is proportional to the adaptation gain. Averaging methods have recently received much attention and may lead to a unification of analysis of adaptive systems.

**Adaptive Schemes Derived from Stochastic Control Theory**

It would be appealing to obtain adaptive systems from a unified theoretical framework. This can be done using nonlinear stochastic control theory, in which system and its environment are described by a stochastic model. The parameters are introduced as state variables, and the parameter uncertainty is modeled by stochastic models. An unknown constant is thus modeled by the differential equation,

\[ \frac{d\theta}{dt} = 0 \]
or the difference equation,

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

with an initial distribution that reflects the parameter uncertainty. Parameter drift can be described by adding random variables to the right-hand sides of the equations above. A criterion is formulated, so as to minimize the expected value of a loss function, which is a scalar function of states and controls.

The problem of finding a control that minimizes the expected loss function is difficult. Under the assumption that a solution exists, a functional equation for the optimal loss function can be derived using dynamic programming. This requires the solution of a functional equation, called the Bellman equation. The structure of the optimal regulator obtained is shown in Fig. 1.9. The controller can be regarded as composed of two parts: a nonlinear estimator and a feedback regulator. The estimator generates the conditional probability distribution of the state from the measurements. This distribution is called the hyperstate of the problem. The feedback regulator is a nonlinear function that maps the hyperstate into the space of control variables. This function can 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. Notice that there is no distinction between the parameters and the other state variables in Fig. 1.9. This means that the regulator can handle very rapid parameter variations.

The optimal control law has interesting properties, which have been found by solving a number of specific problems. The control 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 controls. 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 regulator in Fig. 1.9 with the self-
tuning regulator in Fig. 1.8. 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. In the opti-
mal stochastic regulator there is no such distinction. There is no feedback 
from the variance of the estimate in the STR, although this information 
is available in the estimator. The design calculations in the STR are made 
in the same way as if the parameters were known exactly. In the optimal 
stochastic regulator there is feedback from the conditional distribution 
of parameters and states. Finally, there are no attempts in the STR to 
 improve the estimates when they are uncertain. In the optimal stochastic 
regulator the control law is calculated based on the hyperstate, which takes 
full account of uncertainties. This also introduces perturbations when the 
estimates are poor. The comparison indicates that it may be useful to add 
parameter uncertainties and probing to the STR.

Dual control is discussed in more detail in Chapter 7. A simple exam-
ple illustrates the dual control law and some approximations.

**Example 1.6—Dual control**
Consider a discrete time system described by

\[ y(t + 1) = y(t) + bu(t) + e(t + 1) \]

where \( u \) is the control, \( y \) the output, and \( e \) normal \((0, \sigma_e)\) white noise. Let 
the criterion be to minimize the loss function

\[ V = E \left( \frac{1}{N} \sum_{k=t+1}^{t+N} y^2(k) \right) \]

When the process is known, the problem is a special case of the system 
in Example 1.3, with \( a = -1 \) and \( c = 0 \). The optimal control law is then 
given by Eq. (1.7), i.e.,

\[ u(t) = -y(t)/b \]

The parameter \( b \) is now assumed to be a random variable with a Gaussian 
prior distribution. The conditional distribution of \( b \), given inputs and 
outputs up to time \( t \), is Gaussian with mean \( \hat{b}(t) \) and standard deviation 
\( \sigma(t) \). The hyperstate can then be characterized by the triple \((y(t), \hat{b}(t), \sigma(t))\). The equations for updating the hyperstate in this case are the 
ordinary Kalman filtering equations.
The following control law,

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

which is obtained simply by taking the control law of Eq. (1.7) for known parameters and replacing the parameters with their estimates, is called the certainty equivalence controller. The self-tuning regulator can be interpreted as a certainty equivalence controller.

The optimal control problem can be solved analytically for \( N = 1 \). The control law then becomes

\[ u(t) = -\frac{1}{\hat{b}(t)} \cdot \frac{\hat{b}^2(t)}{\hat{b}^2(t) + \sigma^2(t)} y(t) \]

This control law is called one-step control or myopic control, because the loss function \( V \) only looks one step ahead. It is also called cautious control because, in comparison with the certainty equivalence control, it hedges by decreasing the gain when the estimate of \( \hat{b} \) is uncertain.

For \( N > 1 \) the optimization can no longer be made analytically. The control law can be computed numerically. This results in a control law of the form

\[ u = F(y, \hat{b}, \sigma) \]

where \( F \) is a complicated function. It can be shown that the control law is close to the cautious control for large values of the ratio \( \hat{b}/\sigma \). However, for small ratios the optimal control law is quite different both from cautious control and certainty equivalence control, because it exhibits probing.

\[ \square \]

1.4 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 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. Because of this, adaptive regulators are also entering the marketplace even in single-loop controllers.

A great number of industrial control loops are today under adaptive control. These include a wide range of applications in aerospace, process control, ship steering, robotics, and other industrial control systems. The applications have shown that there are many cases in which adaptive control is very useful and others in which the benefits are marginal. A survey of some applications is found in Chapter 12.
Figure 1.10 The figure shows the variations in heading and the corresponding rudder motions. (a) Adaptive autopilot and (b) Conventional autopilot based on a PID-like algorithm.

Industrial Products

Adaptive techniques are being used in a number of products. Gain scheduling is the standard method for design of flight control systems for high-performance aircraft. It is also used in the process industries. Self-oscillating adaptive systems have been used in missile control systems for a long time. Several commercial autopilots for ship steering are based on adaptive control. There are adaptive motor drives and adaptive systems for industrial robots, and adaptive control systems for biomedical products. For process control applications there are general-purpose adaptive systems as well as dedicated, special-purpose systems.

Based on the products that are now appearing it is clear that adaptive techniques can be used in many different ways. The commercial adaptive regulators are illustrated by two examples.

Example 1.7—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 is easier to operate because it requires fewer adjustments, and it can have 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. A conventional autopilot has three dials, which have to be adjusted over a continuous scale. 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. 1.10. 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 for the conventional autopilot, but that the rudder motions are of the same magnitude. The reason why the adaptive autopilot is better is that 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 regulator 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 regulator. The adaptive autopilot is discussed in more detail in Chapter 12.

The next example illustrates a general-purpose adaptive system.

Example 1.8—Novatune
The first general-purpose adaptive system was announced by the Swedish company Asea Brown Boveri 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 simple means of introducing adaptation. See Chapter 12.

Auto-tuning
Although work on auto-tuning started later than work on adaptive control, it is now rapidly gaining industrial acceptance. One reason for this is that auto-tuners are useful; another is that they are simple to implement. Auto-tuning is applied to single-loop PID controllers as well as to PID controllers in DDC packages. An example of an industrial PID regulator with auto-tuning is shown in Fig. 1.11. An ordinary regulator may be regarded as automation of the actions of a process operator; auto-tuning may be viewed as the next level of automation, in which the actions of an instrument engineer are automated. Although auto-tuning is current being widely applied to simple regulators, it is also useful for more complicated regulators. It is in fact a prerequisite for the widespread use of more advanced control algorithms. An auto-tuning mechanism is often necessary to get the correct timescale and to find a starting value for a more complex
Figure 1.11  A commercial PID regulator with automatic tuning (SattControl Instruments ECA40). Tuning is performed on operator demand when the tune button is pushed. (By courtesy of SattControl Instruments.)

The main advantage of using an auto-tuner is that it simplifies tuning drastically and thus contributes to improved control quality.

Automatic Construction of Gain Schedules

Auto-tuning or adaptive algorithms may be used to build gain schedules. A scheduling variable is first determined. The parameters obtained when the system is running in one operating condition are then stored in a table, together with the scheduling variable. The gain schedule is obtained when the process has operated at a range of operating conditions that covers the operating range.

It is easy to install gain scheduling into a computer-controlled system. The only facilities required are a table for storing and recalling regulator parameters and appropriate commands. Such systems have the advantage that they can follow rapid changes in the operating conditions.

Adaptive Control

The adaptive techniques may, of course, also be used for genuine adaptive control of systems with time-varying parameters. There are many ways to do this. The operator interface is important, since adaptive regulators may also have parameters, which must be chosen. It has been our experience that regulators without any externally adjusted parameters can be designed for specific applications, where 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 regulator 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 regulators can be designed using this concept. For example, it is possible to have a regulator with one dial, labeled with the desired closed-loop bandwidth. Another possibility would be to have a regulator 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.

Uses and Abuses of Adaptive Control

An adaptive regulator, being inherently nonlinear, is more complicated than a fixed-gain regulator. 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 where constant-gain feedback can do as well as an adaptive regulator. This is one reason it is appropriate to include alternatives to adaptive control in this book.

Notice that it is not possible to judge the need for adaptive control from the variations of the open-loop dynamics over the operating range. There are many cases in which a constant-gain feedback can cope well with considerable variations in system dynamics. However, for large classes of problems it often requires only a modest effort to get a general-purpose adaptive regulator to work well.

1.5 Conclusions

The purpose of this chapter has been to give a quick introduction to adaptive control. The topics discussed will be dealt with in much more detail in later chapters, and a perspective on the field is provided in the last chapter of the book. A short assessment of the field can be formulated as follows. Adaptive techniques are emerging after a long period of research and experimentation. Important theoretical results on stability have been established, although much theoretical work still remains. The field is currently in a state of rapid development, in which the advent of microprocessors has been a strong driving force for the applications. Laboratory experiments and industrial feasibility studies have contributed to a better understanding of the practical aspects of adaptive control. A number of adaptive regulators are now appearing on the market, and several thou-
sand loops are now under adaptive control. Although the field is scattered, it is clear that adaptive techniques offer new insights into control system design.

We will end this chapter by giving a few reasons for learning about adaptive control. To the student we emphasize that the field is at the crossroads of many central themes in automatic control. (See Fig. 1.1.) There are also many good research problems in the field. The engineer working in industry should note that adaptive control involves such system features as nonlinear compensation, use of tables, mechanisms for parameter adjustments, and automatic tuning. Armed with the techniques of adaptive control, an engineer is presumably better equipped to design a good control system.

Problems

1.1 Look up the definition of "adaptive" in a good dictionary. How would you define adaptive control?

1.2 Find descriptions of adaptive regulators from some manufacturers and browse through them. What do the manufacturers mean by adaptive control?

1.3 Which processes do you consider to be good candidates for adaptive control?

References

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


Reprints of 44 fundamental papers in adaptive control are found in:

Later development in adaptive control is also treated in:


The following are some fundamental references on the different adaptive schemes in Section 1.2. Robust high-gain control is presented in:


Self-oscillating adaptive systems are discussed in Gregory (1959) and Mishkin and Braun (1961), referenced above. The model-reference approach based on the MIT rule is given in:


One important paper, treating the stability problems in MRAS is:


The combination of recursive estimation and control design is first proposed in:


The self-tuning idea and its asymptotic properties were first derived in:


A unified approach to continuous-time self-tuning regulators is found in:


Convergence and stability properties of adaptive controllers are thoroughly discussed in:


The application of averaging theory to adaptive control is found in:


Recursive estimation is an important part of many adaptive controllers. A unified approach to this scattered field is presented in:


Adaptive control theory are treated in Tsypkin (1971) and Goodwin and Sin (1984), referenced above. Interesting aspects and connections to optimization theory are given in:


Surveys of applications of adaptive control are given in:


Chapter 2

WHY ADAPTIVE CONTROL?

2.1 Introduction

The purpose of this introductory chapter is to explore under what circumstances constant-gain linear feedback is insufficient and to provide some perspective on the need for adaptive control. One of the goals of adaptive control is to compensate for parameter variations. Section 2.2 gives a collection of examples in which parameter variations cause difficulties when a fixed-gain regulator is used. Some generic examples of mechanisms that change the dynamics are discussed. Parameters may vary due to nonlinear actuators, changes in the operating conditions of the process, and nonstationary disturbances acting on the process. The examples in Section 2.2 give background as to why adaptive control may be of interest. Before different adaptive control methods are presented in later chapters, some alternatives will be discussed. The most common regulator is a feedback
controller with fixed parameters. Through feedback it is possible to decrease the sensitivity to parameter variations by increasing the loop gain of the system. High-gain controllers are briefly discussed in Section 2.3. The main drawbacks of high-gain controllers are the magnitude of the control signal and the problem of stability of the closed-loop system. If there are bounds on the uncertainties of the process parameters, it is possible to design robust controllers by increasing the complexity of the controller. To use this approach it is necessary to know the structure of the process fairly accurately and to have bounds on the variations of the parameters. Robust control is also discussed in Section 2.3. The need for adaptation is discussed in Section 2.4, together with means of formulating the adaptive control problem.

2.2 When Is Constant-gain Feedback Insufficient?

Conventional control theory deals predominantly with linear systems having constant parameters. This is often a good approximation for systems that are regulated at fixed operating points. With moderate disturbances and a good control system the deviations will be so small that the linear approximation is sufficiently good. However, the linear constant coefficient approximation will not always be satisfactory when the operating conditions change. In this section we will give examples of such control problems.

*Hint for the reader:* The first-time reader is advised to scan this section and to concentrate on one or two examples of special interest in order to get a feel for the nature of changes in process dynamics.

**Nonlinear Actuators**

Many actuators have a nonlinear characteristic, which creates difficulties unless special precautions are taken. This is illustrated by an example.

**Example 2.1—Nonlinear valve**

A simple feedback loop with a nonlinear valve is shown in Fig. 2.1. Let the static valve characteristic be

\[ v = f(u) = u^4 \quad u \geq 0 \]

Linearizing the system around a steady-state operating point shows that the loop gain is proportional to \( f'(u) \). It then follows that the system can perform well at one operating point and poorly at another. This
is illustrated by the step responses in Fig. 2.2. One way to handle this type of problem is to feed the control signal $u$ through an inverse of the nonlinearity. 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.

Concentration Control

Systems with flow through pipes and tanks are common in process control. The flows are often closely related to the production rate. The process dynamics thus change when the production rate changes, and a regulator

![Figure 2.2 Step responses for PI control of the simple flow loop in Fig. 2.1 at different operating conditions. The parameters of the PI controller are $K = 0.15$, $T_i = 1$. Further, $f(u) = u^4$ and $G_0(s) = 1/(s + 1)^3$.](image)
that is well tuned for one production rate will not necessarily work well for other rates. A simple control example illustrates what may happen.

Example 2.2—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. 2.3. The concentration $c_{in}$ at the inlet is regulated by mixing two flows in a small tank whose dynamics can be neglected. Let the pipe volume be $V_d$ and the tank volume $V_m$. Furthermore, let the flow be $q$ and the concentration in the tank and at the outlet be $c$.
2.2 When Is Constant-gain Feedback Insufficient?

\[ (a) \]

\[ \begin{align*}
q &= 0.5 \\
q &= 0.9 \\
q &= 1.1 \\
q &= 2.0 \\
\end{align*} \]

\[ (b) \]

\[ \begin{align*}
1.0 & 0.9 & 0.8 \\
1.0 & 0.9 & 0.8 \\
\end{align*} \]

**Figure 2.5** Output \( c \) and reference \( c_r \) concentration for the system in Example 2.2 for different flows. (a) Change in reference value \( c_r = 1 \). (b) Load disturbance \( c_r = 0 \).

A mass balance gives

\[ V_m \frac{dc(t)}{dt} = q(t) (c_{in}(t - \tau) - c(t)) \]  \hspace{1cm} (2.1)

where

\[ \tau = V_d / q(t) \]

Introduce

\[ T = V_m / q(t) \]  \hspace{1cm} (2.2)

For a fixed flow, the process has the transfer function

\[ G_0(s) = \frac{e^{-s\tau}}{1 + sT} \]  \hspace{1cm} (2.3)

The dynamics is characterized by a time delay and first-order dynamics. Both the time constant \( T \) and the time delay \( \tau \) are inversely proportional to the flow \( q \). The consequences of these variations for the control system are illustrated in Fig. 2.4 and Fig. 2.5. Figure 2.4 shows the response to step changes and load disturbances with conventional PI control under nominal operating conditions. The regulator has the gain \( K = 0.5 \) and...
the integration time $T_i = 1.1$. Figure 2.5 shows what happens when the same regulator is used and the flow changes. It is seen that the response to changes in the command signal deteriorates even for small changes of the flow. Notice the drastic change in performance when the flow changes by a factor of 2. However, the response to load changes is not much influenced by a flow change of $\pm 10\%$. If the flow changes by a factor of 2 or more, it is necessary to retune the regulator in order to get reasonable performance. Note that it is safest to tune the regulator at the smallest flow.

\[ \square \]

**Robotics**

Industrial robots are based on high-precision servo technology. A representative robot is shown in Fig. 2.6. The tip of the robot arm is positioned by servos in the joints. In many applications there is no position feedback from measurements of the tip of the arm; precision is achieved by a combination of mechanical stiffness and good servos. The robot is a nonlinear system because of the nonlinear mechanical coupling between the different motions, the variations in moment of inertia, and compliance with geometry. An example illustrates the range of variation in the moment of inertia.

**Example 2.3—Variations in the moment of inertia**

A simple robot arm with two axes is shown in Fig. 2.7. Consider the control problem for the vertical axis. A torque balance for this axis gives

\[
\frac{d}{dt} \left( J(\theta) \frac{d\varphi}{dt} \right) = J \frac{d^2 \varphi}{dt^2} + \frac{\partial J}{\partial \theta} \frac{d\theta}{dt} \frac{d\varphi}{dt} = T_f + T_e \tag{2.4}
\]
2.2 When Is Constant-gain Feedback Insufficient?

where $\varphi$ is the angle around the vertical axis, $\theta$ is the arm angle, $J$ is the total moment of inertia of motor and load, $T_f$ is the friction torque, and $T_e$ is the torque generated by the motor. The moment of inertia depends on the angle $\theta$ and on the load $m_L$. This dependence is of the form

$$J(\theta, m_L) = \alpha + \beta m_L + (\gamma + \delta m_L) \sin^2 \theta$$

With reasonable values of the constants the moment of inertia may change in the ratio 1:50 from the upright to the outstretched position. Notice that part of the change in inertia can be calculated from the geometry. The other part depends on the load $m_L$, which may be unknown. The variations are smaller if a geared motor is used, because the moment of inertia at the motor axis becomes

$$J_{ma} = J_m + \frac{1}{N^2} J$$

where $J_m$ denotes the moment of inertia of the motor rotor and $N$ is the gear ratio. Many factors determine the gear ratio: for instance, desired translation acceleration and speed, time to move small distances, and available motors. Variations in the total moment of inertia of the order 1:5 are common in current industrial robots. \hfill \Box

The consequences of variations in the moment of inertia for the servo design are illustrated by an example.

Example 2.4—Velocity servo for a robot arm

Consider a system for controlling the angular velocity of the robot arm in Example 2.3 (see Fig. 2.8). Assuming that $\theta$ is constant, the moment of inertia $J$ is then also constant, and Eq. (2.4) becomes

$$J \frac{d\omega}{dt} = T_e$$

where $\omega$ is the angular velocity and the friction torque $T_f$ is neglected.
It is assumed that the motor is a DC motor with current feedback. The torque from the motor is

$$T_e = k_m I$$  \hspace{1cm} (2.7)

where $I$ is the current through the rotor. Assume that the control system is a PI regulator

$$I = K \left( \alpha \omega_{ref} - \omega + \frac{1}{T_i} \int_{t}^{t} (\omega_{ref} - \omega) d\tau \right)$$  \hspace{1cm} (2.8)

Elimination of $T_e$ and $I$ between Eqs. (2.6), (2.7), and (2.8) gives

$$J \frac{d^2 \omega}{dt^2} + k_m K \frac{d\omega}{dt} + \frac{k_m K}{T_i} \omega = \alpha k_m K \frac{d\omega_{ref}}{dt} + \frac{k_m K}{T_i} \omega_{ref}$$  \hspace{1cm} (2.9)

If the regulator parameters are chosen as

$$K = \frac{2\zeta_0 \omega_0 J}{k_m}$$

$$T_i = \frac{2\zeta_0}{\omega_0}$$ \hspace{1cm} (2.10)

then the transfer function, $G_c(s)$, relating the velocity $\omega$ to the command velocity, becomes

$$G_c(s) = \frac{2\alpha \zeta_0 \omega_0 s + \omega_0^2}{s^2 + 2\zeta_0 \omega_0 s + \omega_0^2}$$ \hspace{1cm} (2.11)

The parameter $\alpha$ may be used to adjust overshoot of the step response. It follows from Eq. (2.10) that the gain should be proportional to the moment of inertia. If the regulator is designed for a nominal moment of inertia $J_{\text{nom}}$ and applied to a robot with the moment of inertia $J$, the characteristic equation becomes

$$s^2 + 2\zeta_0 \frac{J_{\text{nom}}}{J} \omega_0 s + \frac{J_{\text{nom}}}{J} \omega_0^2 = 0$$ \hspace{1cm} (2.12)
2.2 When Is Constant-gain Feedback Insufficient?  

![Figure 2.9](image)

Figure 2.9 Step response of a robot with a regulator having fixed parameters and different moments of inertia, $J_{\text{nom}}/J = 0.25$, 1, and 4.

The damping becomes

$$\zeta = \zeta_0 \sqrt{\frac{J_{\text{nom}}}{J}} \quad (2.13)$$

The relative damping is thus inversely proportional to the square root of the moment of inertia. If the nominal value of the relative damping is $\zeta_0 = 0.707$, it will increase to 1 if the moment of inertia is reduced by half, and it will decrease to 0.5 if the moment of inertia doubles. The effect of changes in the movement is illustrated in Fig. 2.9. It follows from Eq. (2.13) and Fig. 2.9 that a conventional PI regulator gives reasonable responses for moments of inertia varying in the range $0.4$. If the moment of inertia varies more, it is clear that a regulator of PI type with constant parameters is inadequate.  

Variation in the moment of inertia is not the only problem in the design of servos for industrial robots; compliance in the arms is another source of difficulties. This gives oscillatory dynamics with low damping. There are two options for design of control for such systems: to design the control system so that it does not excite the poorly damped modes, and to design a control system that actively damps the oscillatory modes. Active damping of the flexure modes is not used in current industrial robots. One reason for this is that the control system becomes complicated, since the frequencies of the oscillatory modes vary with orientation and load.

Industrial robots are much more complex than the simple prototype used in the examples. There are many joints; the dependence of the moments of inertia on the geometry is more complicated; and there are also several modes of oscillation. A more realistic model is described in Section 4.6. The examples indicate, however, that the parameter variations depend on the process design. Since robots with high performance must be light, it is clear that the effects of compliance will be more significant in robots designed in the future.
Airplane Control

Control of high-performance airplanes was a strong driving force for the development of adaptive control in the 1960s. The dynamics of an airplane depend on speed, altitude, angle of attack, etc. Also, the flexible structure of the airplane will cause difficulties when designing the control system. The following example adopted from Ackermann (1983) shows the variation of the short-period dynamics.

Example 2.5—Short-period airplane dynamics

Figure 2.10 gives the notations for the airplane. The rigid body or the so-called short-period dynamics for an experimental F4-E with canards will be given. The canards make it easier to maneuver the airplane, at the cost of decreased stability. The dynamics can be linearized around stationary flight conditions (i.e., constant speed and altitude and small angle of attack $\alpha$). Let the state variables be normal acceleration $N_z$ and pitch rate $q = \dot{\theta}$. It is furthermore assumed that the dynamics of the servo of the elevons and the canards have the transfer function

$$\frac{a}{s + a}$$

with $a = 14$. Let $\delta_c$ be the corresponding state. The model of the airplane then has three states: $N_z$, $q$, and $\delta_c$, and the linearized equations have the form

$$\dot{x} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & -a \end{pmatrix} x + \begin{pmatrix} b_1 \\ 0 \\ a \end{pmatrix} u \quad (2.14)$$

The bending modes are not included in the model. The first bending mode has the frequency 85 rad/s. Figure 2.11 shows the flight envelope and four different flight conditions. Table 2.1 shows the parameters for the flight conditions indicated in Fig. 2.11. The eigenvalues of the short-period dynamics, $\lambda_1$ and $\lambda_2$, are also given. It is seen that the airplane is unstable for
subsonic speeds and poorly damped for supersonic speeds. The autopilot is designed for different flight conditions, and the parameters are changed using gain scheduling. (Compare Section 9.5). Air data (i.e., dynamic pressure and Mach number) are used as the scheduling variables. Further pre-filters are used on the command signals from the pilot. It is very important that the controller not excite the bending modes of the airplane.

Rotary Dryer

Drying of materials is a common unit process in many industries. The material to be dried may be paper, cement, grain, food, etc. A typical

Table 2.1 Parameters of the airplane state model of Eq. (2.14) for different flight conditions.

<table>
<thead>
<tr>
<th></th>
<th>FC 1</th>
<th>FC 2</th>
<th>FC 3</th>
<th>FC 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mach</td>
<td>0.5</td>
<td>0.85</td>
<td>0.9</td>
<td>1.5</td>
</tr>
<tr>
<td>Altitude (feet)</td>
<td>5000</td>
<td>5000</td>
<td>35000</td>
<td>35000</td>
</tr>
<tr>
<td>$a_{11}$</td>
<td>-0.9896</td>
<td>-1.702</td>
<td>-0.667</td>
<td>-0.5162</td>
</tr>
<tr>
<td>$a_{12}$</td>
<td>17.41</td>
<td>50.72</td>
<td>18.11</td>
<td>26.96</td>
</tr>
<tr>
<td>$a_{13}$</td>
<td>96.15</td>
<td>263.5</td>
<td>84.34</td>
<td>178.9</td>
</tr>
<tr>
<td>$a_{21}$</td>
<td>0.2648</td>
<td>0.2201</td>
<td>0.08201</td>
<td>-0.6896</td>
</tr>
<tr>
<td>$a_{22}$</td>
<td>-0.8512</td>
<td>-1.418</td>
<td>-0.6587</td>
<td>-1.225</td>
</tr>
<tr>
<td>$a_{23}$</td>
<td>-11.39</td>
<td>-31.99</td>
<td>-10.81</td>
<td>-30.38</td>
</tr>
<tr>
<td>$b_1$</td>
<td>-97.78</td>
<td>-272.2</td>
<td>-85.09</td>
<td>-175.6</td>
</tr>
<tr>
<td>$\lambda_1$</td>
<td>-3.07</td>
<td>-4.90</td>
<td>-1.87</td>
<td>-0.87 ± 4.3i</td>
</tr>
<tr>
<td>$\lambda_2$</td>
<td>1.23</td>
<td>1.78</td>
<td>0.56</td>
<td></td>
</tr>
</tbody>
</table>
process scheme is given in Fig. 2.12. It can be very difficult to derive a good process model, since the dynamics depend on many variables. Also, the dynamics will change with the speed of the material through the dryer and the moisture content of the incoming material. For instance, the dead time will vary with production rate. This implies that a constant-gain controller has to be conservatively tuned to the longest time delay, and the control becomes sluggish for shorter delays.

In drying it is important to combine feedback and feedforward. For instance, the moisture content in the incoming product or the production rate can be used as feedforward signals. To achieve good feedforward control it is essential to have good knowledge of the dynamical influence of the measurable disturbances.

The rotary dryer is a typical industrial process, in which the physics is not fully understood. This makes the process difficult to control using conventional controllers. An adaptive controller can be very useful, since it can adapt to the changing dynamics and to the variations in the disturbances.

Nonstationary Disturbance Patterns

Disturbances are an important aspect of feedback control. Without disturbances there is little need for feedback. It is straightforward to compensate for disturbances with given characteristics. Low-frequency disturbances, like infrequent step changes of a load, can be effectively reduced by a controller having integral action. Narrow-band disturbances with a known center frequency can be reduced by a regulator having a resonant mode corresponding to the frequency of the disturbance. There are many processes in which the characteristic disturbances acting on the system vary with time. One example is paper machine control, in which the quality
of the pulp influences the formation of the sheet of the paper. Another example is ore crushing, in which the size and hardness of the ore varies over a wide range. In these and similar examples it may be necessary to retune the controller in order to maintain good stationary control.

**Example 2.6—Dynamic positioning**

Positioning of ships and platforms is very important in the offshore industry. Disturbances on the position are generated by wind, waves, and current, and can be separated into low-frequency and high-frequency components. The low-frequency component is due to steady wind and current; the high-frequency component is due to waves. The dynamics of the vessel are such that it is not possible to compensate for the high-frequency disturbances. To obtain good control it is necessary to filter the measurements so that the waves do not introduce a modulation in the thrusters. The dominating wave frequency parameter may change by a factor of 3 when the weather conditions vary from light breeze to fresh gale. Typical measurements of waves and spectra for two weather conditions are shown in Fig. 2.13. By adaptation the regulators can be adjusted to the changing characteristics of the disturbances.
Summary

The examples above illustrate mechanisms that can create parameter variations such that the processes may be candidates for adaptive control. The examples are, however, oversimplified. In practice there are many different sources of parameter variations, and there is usually a mixture of different phenomena. The underlying reasons for the variations are in most cases not fully understood, as in the case of the rotary dryer. When the physics is well known (as for airplanes), it can be sufficient to use gain scheduling. In fact, this is the common way to construct autopilots for airplanes. Most industrial processes are, however, very complex; it is not possible or economical to make a thorough investigation of the causes of the parameter variations. Adaptive or self-tuning 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, where the geometry, motors, and gear-boxes do not change, but the load does change. In such cases it is of great importance to utilize the available \textit{a priori} knowledge and only estimate and adapt to the unknown part of the process. How to use \textit{a priori} knowledge will be discussed in Chapter 4.

2.3 Robust Control

Feedback by itself has the ability to cope with parameter changes. The search for ways to design a system insensitive to parameter variations in fact led to the invention of the electronic feedback amplifier. Analysis of feedback amplifiers by Nyquist and Bode is one of the cornerstones of feedback theory. Further refinements of the theory led to design methods that explicitly take process uncertainty into account. Before starting with adaptive control we will investigate how linear fixed-gain controllers can deal with parameter variations. We start by investigating a simple example.

Example 2.7—An electronic feedback amplifier

Consider the electronic amplifier shown in Fig. 2.14. The active component is an operational amplifier whose gain can change between $10^4$ and $10^5$, depending on individual variations and the operating conditions. In spite of this the amplifier shown in the figure will have an input-output characteristic that is almost constant. The dynamics of the operational amplifier can be approximately described by the transfer function

$$G_0(s) = \frac{k}{(1 + sT)(1 + sT_1)^2}$$

where the parameters $T$ and $T_1$ have the nominal values $10^{-2}$ and $10^{-6}$,
Figure 2.14  An electronic amplifier.

respectively. Neglecting the current that flows into the operational amplifier, we get

\[
\frac{V_1 - V_0}{R_1} = \frac{V_0 - V_2}{R_2}
\]

Furthermore,

\[V_2 = -G_0(s)V_0\]

Simple calculations give the following input-output relation from \(V_1\) to \(V_2\):

\[
G_e = \frac{V_2}{V_1} = -\frac{R_2}{R_1} \cdot \frac{G_0}{1 + G_0 + \frac{R_2}{R_1}} = -\frac{R_2}{R_1} \cdot \frac{1}{1 + \frac{R_1 + R_2}{R_1 G_0}}
\]

If \(R_2/R_1=100\), the amplifier has a gain of 100 with a precision of 1% as long as the amplifier gain remains larger than 10^4.

The stability of the closed-loop system is determined by the characteristic equation

\[
s^3TT_1^2 + s^2(T_1^2 + 2TT_1) + s(T + 2T_1) + 1 + \frac{k}{s + a} = 0
\]

To simplify the writing we have introduced

\[a = \frac{R_2}{R_1}\]

The stability condition is

\[(T_1 + 2T)(T + 2T_1) > TT_1 \frac{1 + a + k}{1 + a}\]

or approximately, for \(k \gg a \gg 1\) and \(T \gg T_1\),

\[2aT > kT_1\]
This condition is satisfied as long as the amplifier gain is less than $10^6$. Hence, with the given design, the DC amplifier gain remains within 1% of the nominal value, in spite of the fact that the open-loop gain varies by an order of magnitude.

The example shows that a closed-loop system will be insensitive to parameter variations as long as the loop gain is high. This observation can be generalized as follows. Consider a simple closed-loop system with unity feedback, as shown in Fig. 2.15(a). The normalized relative sensitivity of the closed-loop transfer function with respect to changes in the process transfer function is given by

$$\frac{dG_c}{G_c} \cdot \left( \frac{dG_0}{G_0} \right)^{-1} = \frac{G_0}{G_c} \cdot \frac{dG_c}{dG_0} = \frac{1}{1 + G_0}$$

This expression clearly shows that the sensitivity is low as long as the loop gain is high. However, when the gain is increased, there is a risk that the closed-loop system will become unstable. In the case of Example 2.7 we get

$$\frac{dG_c}{G_c} \cdot \frac{k}{dk} = \frac{R_1 + R_2}{R_1 + R_2 + R_1 G_0}$$

For low frequencies and $k \gg a \gg 1$ this is approximately $R_2/(R_1 k) < 0.01$, which, of course, agrees with the results of the detailed calculation in Example 2.7.

**Unstructured Perturbations**

We will now develop a condition that guarantees the stability of the closed loop system under general perturbations in the process model. For this purpose consider the simple feedback system shown in Fig. 2.15. Let $G_0$ be the nominal loop transfer function. Assume that the true loop transfer function is $G = G_0(1 + L)$, due to model uncertainties. Notice that $1 + L$ is the ratio between the true and nominal transfer functions. It is assumed
that the perturbations are such that \( G \) and \( G_0 \) have the same number of poles in the open right half-plane.

The effect of uncertainties on the stability of the closed-loop system will first be discussed.

**Theorem 2.1—Perturbation robustness**

Assume that the closed-loop system obtained with unity feedback around \( G_0 \) is stable. Consider a perturbation \( L \), with no poles in the right half-plane, which satisfies

\[
|L(s)| < \left| \frac{1 + G_0(s)}{G_0(s)} \right| \tag{2.15}
\]

on a contour that encloses the right half-plane. The closed-loop system obtained with unity feedback around \( G = G_0(1 + L) \) is then also stable.

**Proof:** The closed-loop poles of the perturbed system are the zeros of the equation

\[
1 + G_0(s) + G_0(s)L(s) = 0
\]

The theorem follows by applying Rouché’s theorem to the functions \( 1 + G_0 \) and \( G_0L \).

If the inequality of Eq. (2.15) is violated, there exists a perturbation \( L \) that gives an unstable closed-loop system. A perturbation \( L \) such that

\[
L(\pm i\omega_0) = -\frac{1 + G_0(\pm i\omega_0)}{G_0(\pm i\omega_0)} \tag{2.16}
\]

gives a closed-loop system with a pole at \( s = \pm i\omega_0 \). A perturbation \( L \) with no additional restrictions is called an *unstructured model uncertainty*. In many practical cases the perturbations have a very special form, because they may originate from variations of physical parameters. Such perturbations, which are called *structured model uncertainties*, may give stable closed-loop systems even if they violate the inequality of Eq. (2.15).

**Robustness and Performance**

The tradeoff between robustness and performance is a key element of control design. To describe this tradeoff, introduce the closed-loop transfer function

\[
G_c = \frac{G_0}{1 + G_0}
\]

Equation (2.15) can then be written as

\[
|L| |G_c| < 1
\]
To obtain a large degree of robustness (i.e., insensitivity to modeling errors) it is thus desirable that $|G_c|$ be small.

The closed-loop transfer function can also be written as

$$G_c = \frac{G_0}{1 + G_0} = 1 - \frac{1}{1 + G_0} = 1 - S$$  \hspace{1cm} (2.17)

where $S$ is the sensitivity function. To obtain a good control performance it is thus desirable to have a small $S$. It follows from Eq. (2.17) that

$$S + G_c = 1$$  \hspace{1cm} (2.18)

This implies that $S$ and $G_c$ cannot be made small simultaneously. If we want good performance, the robustness must be sacrificed, and vice versa.

The key element in the design of fixed-gain control systems is to determine the frequency ranges in which performance and robustness should dominate. The value of adaptation is that model uncertainty can be reduced. This makes it possible to get good performance over a wider range.

**Structured Perturbations**

The results on unstructured perturbations are conservative; in practice there are often restrictions on the uncertainty because of their physical origin. Considerable insight into the effects of structured perturbations can be obtained by analyzing the simple feedback system shown in Fig. 2.15. The nominal loop transfer function $G_0$ is the product of the transfer function $G_p$ of the process and the transfer function $G_r$ of the regulator. The closed-loop transfer function is given by Eq. (2.17). The key robustness problem is to understand how changes in $G_p$ will influence the closed-loop transfer function $G_c$. Considerable insight into this problem can be derived from the Nyquist diagram (see Fig. 2.16). This diagram also shows the loci where the closed-loop system $G_c$ has constant magnitude $M$ and constant phase $\phi$. These curves are circles, which are called $M$-circles and $\phi$-circles.

A perturbation of the plant dynamics implies that the Nyquist curve changes. Consider a particular point on the Nyquist curve. A change of gain means that the point moves radially; a change of phase means that the point moves on a circle centered at the origin. It follows from the diagram that the process dynamics can change significantly far from the origin without changing the closed-loop dynamics much. The distance between neighboring $M$-circles and $\phi$-circles will increase with the distance from the origin. This key observation for high-gain control can also be made analytically from Eq. (2.17), because it follows that $G_c$ is close to 1 when $G_0$ is large. The closed-loop system is, however, quite sensitive to changes
in the open-loop dynamics for those frequencies at which the loop gain is close to 1.

The changes in the Nyquist curve must not be so large that the closed-loop system becomes unstable. For example, changes corresponding to a sign reversal cannot be tolerated, because this would correspond to the Nyquist curve being mirror-imaged in the real and imaginary axes. A simple example illustrates this.

**Example 2.8—Integrator with unknown sign**

Let the process be an integrator

\[ G_0(s) = \frac{k_p}{s} \]

If the sign is not known, the process cannot be controlled using high-gain robust control. This can be seen either from the Nyquist plot or from the characteristic equation

\[ s + k k_p = 0 \]

where \( k \) is the gain of a proportional controller. This equation will clearly be unstable if the process gain changes sign.

It is also essential to avoid perturbations that change the number of right half-plane poles, because this will change the encirclement condition
for the Nyquist theorem. This is illustrated by an example.

**Example 2.9—Right half-plane poles**

Consider a system with the loop transfer function

\[ G_0(s) = \frac{1}{s(s + 1)} \]

The corresponding unity feedback closed-loop system is clearly stable. Now consider the system with the loop transfer function

\[ G(s) = \left( 1 + \frac{\varepsilon}{s - a} \right) G_0(s) = \frac{s - a + \varepsilon}{s - a} G_0(s) \]

where \( a > 0 \). The transfer functions \( G_0 \) and \( G \) can be made arbitrarily close by choosing \( \varepsilon \) sufficiently small. The characteristic equation of the nominal closed loop system is

\[ s^2 + s + 1 = 0 \]

The perturbed system has the characteristic equation

\[ (s - a)(s^2 + s + 1) + \varepsilon = 0 \]

The closed loop system corresponding to the perturbed system is unstable for sufficiently small \( \varepsilon \). \( \square \)

The example shows that in order to guarantee stability it is not enough to know the Nyquist curve. It is also necessary to know if there are any unstable right half-plane poles or zeros that are close to cancellation. This is also clear from the Nyquist stability criterion, which states that the closed-loop system is stable if the number of encirclements of the standard contour equals the number of singularities of the loop transfer function in the right half-plane.

The above discussion shows that it is possible to control many processes using linear feedback control even if the process is not fully known. Special design techniques that deal explicitly with processes having uncertainty will be discussed in Chapter 10. Use of adaptation especially combined with _a priori_ knowledge can, however, be a good alternative to linear feedback control. Feedforward control is very different from feedback control, since it generally requires more accurate process models. Use of adaptation is therefore of even greater importance for feedforward control.
2.4 The Adaptive Control Problem

The examples in Section 2.2 and the discussion of robust control in Section 2.3 show why adaptive controllers are needed. It should be emphasized that typical industrial processes are so complex that the parameter variations cannot be determined from first principles. It can therefore be advantageous to trade engineering efforts against more "intelligence" in the controller. A more complex controller may then 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.

From the presentation of model-reference adaptive controllers and self-tuning controllers in Chapter 1, we find that an adaptive controller will contain

- Control law with adjustable parameters
- Characterization of the closed-loop response (reference model or specifications for the design)
- Design procedure
- Parameter updating based on measurements
- Implementation of the control law.

These parts are somewhat different for different adaptive schemes but have many common features.

Design and analysis of adaptive controllers can be achieved using either continuous-time (Laplace transform) or discrete-time (z-transform) models. Historically, MRAEs have been designed for continuous time while STRs have been designed for discrete time models. We will use the same distinctions in the presentations in Chapters 4 and 5. By doing so it is possible to cover specific properties and difficulties of the two approaches. It should, however, be pointed out that MRAEs can be implemented using discrete-time models and that it is possible to make continuous-time STRs. Some researchers have also advocated a hybrid approach in which the design and analysis are done in continuous time, while the implementation is done in discrete time form.

There is today a gap between theory and practice in adaptive control. In theory it is possible to handle only idealized situations. For instance, it is often assumed that the order of the process is known. In practice quite complex algorithms have been used; ad hoc fixes have been introduced to handle possible difficulties found through analysis or experienced in applications. The treatment of the basic algorithms in this book will be for
idealized situations. Possible extensions are pointed out using the theory in Chapter 6, and implementation aspects are given in Chapter 11.

Judging the Need for Adaptation

The fact that there are significant variations in open-loop step responses does not necessarily mean that adaptive control is needed. This is illustrated by the following examples.

Example 2.10—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 illustrated in Fig. 2.17(a). The closed-loop systems obtained by introducing the feedback \( u = u_c - y \) give the step responses shown in Fig. 2.17(b).

The reason why the closed-loop responses are so close in Example 2.10 is that the transfer functions of the systems are very close at the crossover frequency, although their low-frequency behaviors are different.
2.4 The Adaptive Control Problem

![Graph](image)

Figure 2.18  (a) Open-loop step responses for \( G_0(s) = \frac{20(1-sT)}{(s+1)(s+20)(1+sT)} \) for \( T = 0 \), 0.015, and 0.03. (b) Closed-loop step responses for the same system, with the feedback \( u = 20(u_c - y) \).

The next example illustrates the converse: some systems behave very differently in closed loop even though their open-loop step responses are very close.

**Example 2.11—Similar open-loop responses**
Consider systems with the open-loop transfer functions

\[
G_0(s) = \frac{20(1-sT)}{(s+1)(s+20)(1+sT)}
\]

with \( T = 0 \), 0.015, and 0.03. The open-loop step responses are shown in Fig. 2.18(a) and Fig. 2.18(b) shows the closed-loop step responses for the systems obtained with the feedback \( u = 20(u_c - y) \).

The systems in Example 2.11 have the property that their low-frequency behaviors are similar but the transfer functions differ considerably at high frequencies.

The lesson to be learned from these examples is that it is essential to know the frequency response at the desired crossover frequency in order to judge whether parameter variations will have any effect on the closed-loop systems properties. Step responses are poor guides; it is much better to look at the frequency responses.
2.5 Conclusions

This chapter has given some reasons for using adaptive control. The key factors are

- Variations in process dynamics
- Variations in the character of the disturbances
- Engineering efficiency.

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.

Adaptive control is not the only way to deal with parameter variations. Robust control and gain scheduling are competitive techniques. To have a balanced view of adaptive techniques it is therefore necessary to know these methods as well (see Chapters 9 and 10). However, for feedforward control of processes with varying dynamics there are few alternatives to adaptation. This is also the case when the character of the disturbances changes.

Engineering efficiency is an often overlooked argument. This is one reason for the success of auto-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. When toolboxes for adaptive control (such as Novatune) are available, 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 will be given in the chapters that follow.

Problems

2.1 The system in Fig. 2.1 has the following characteristics

\[ G_0(s) = \frac{1}{(s + 1)^3} \]

\[ f(u) = u^4 \]

The regulator has the gain \( K = 0.15 \) and the reset time \( T_i = 1 \). Determine a reference value such that the linearized equations are just becoming unstable. Linearize the equations when the reference values are \( u_c = 0.3, 1.1, \) and \( 5.1 \). Determine the poles of the characteristic equation in the different cases.
2.2 Consider the concentration control system in Example 2.2. Assume that $V_d = V_m = 1$ and that the nominal flow is $q = 1$. Determine PI controllers that give good closed-loop performance for the flows $q = 0.5$, 1, and 2. Test the controllers for the nominal flow.

2.3 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 static 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 $k_2$.

2.4 A block diagram of a system used for metal cutting on a numerically controlled machine is shown in Fig. 2.19. The machine is equipped with a force sensor, which measures the cutting force. A regulator 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, \( \alpha \) is a parameter in the range \( 0.5 < \alpha < 1 \), and \( k \) is a positive parameter. The static gain from feed rate to force is

\[
K = k \alpha \omega v^{\alpha-1} N^{-\alpha}
\]

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 \( T \) 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) The same as in (a) except that \( a \) varies between 0.5 and 4 and \( N = 1 \).

2.5 Consider the system in Fig. 2.20. Compute the closed-loop transfer function \( G_c \) from \( u_c \) to \( y \) and determine the normalized relative sensitivity of \( G_c \) with respect to \( G \), \( H \), and \( F \). What are the consequences?

2.6 Compute the normalized relative sensitivity of the closed-loop transfer function for the systems in Examples 2.10 and 2.11. Plot the Bode diagrams of the sensitivities. The controllers are of the form

\[
u = K(u_c - y)\]

How will the gain in the controller, \( K \), influence the sensitivity?

2.7 Consider the system in Fig. 2.21. Let the process be

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

where

\[
K = K_0 + \Delta K \quad K_0 = 1
\]
\[
a = a_0 + \Delta a \quad a_0 = 1
\]
and
\[-0.5 \leq \Delta K \leq 2.0\]
\[-2.0 \leq \Delta a \leq 2.0\]

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 (b) and investigate the sensitivity to parameter changes.
(d) Use the controller from (b). Investigate the sensitivity to the disturbance $d(t)$ when
\[d(t) = \begin{cases} 
-1 & 0 \leq t < 6 \\
2 & 6 \leq t < 15 \\
1 & 15 \leq t
\end{cases}\]
(e) Use the controller from (b) and investigate the influence of measurement noise, $e(t)$. Let $e(t)$ be zero mean white noise.

2.8 Make the same investigation as in Problem 2.7 when the process is
\[G_0(s) = \frac{K}{s^2 + a_1 s + a_2}\]

where
\[K = K_0 + \Delta K\quad K_0 = 1\]
\[a_1 = a_{10} + \Delta a_1\quad a_{10} = 1.4\]
\[a_2 = a_{20} + \Delta a_2\quad a_{20} = 1.4\]
and
\[ -0.5 \leq \Delta K \leq 2.0 \]
\[ -2.0 \leq \Delta a_1 \leq 2.0 \]
\[ -3.0 \leq \Delta a_2 \leq 3.0 \]

Let the desired closed-loop response be given by
\[ Y_m(s) = \frac{1}{s^2 + 1.4s + 1} U_c(s) \]

References

Although the content of this chapter is very important, it is seldom dealt with in the literature. Discussions of when adaptive control may be necessary can be found in:


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


A discussion of adaptive flight control is found in:


The airplane in Example 2.5 is taken from:


Robust high-gain control is thoroughly discussed in:

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:


Unstructured perturbations are discussed in:


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


Other references on robustness and sensitivity are:


Chapter 3

REAL-TIME
PARAMETER ESTIMATION

3.1 Introduction

On-line determination of process parameters is a key element in adaptive control. It is an important part of a self-tuning regulator (see, e.g., Fig. 1.8). Parameter estimation also occurs implicitly in a model-reference adaptive regulator. This chapter presents some methods for real-time parameter estimation. In adaptive systems, parameter estimation is used in the larger 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 basic issues. Simple transfer function models will be used in this chapter. The identification problems are simplified signifi-
cantly if the model is 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. This is straightforward for open-loop problems but poses some difficulties in adaptive control when the input signal to the plant is generated by feedback. In certain cases this does not permit the parameters to be determined uniquely, which 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. The purpose of this chapter is to develop such methods. In solving identification problems it is very important to validate the results. This is even more important for adaptive systems, in which identification is done automatically. Some validation techniques will therefore also be discussed.

Least squares 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. The method of least squares is developed in Section 3.2, and a probabilistic interpretation is given. This chapter shows how the estimate can be calculated recursively, presents some simplified recursive algorithms, and shows how the method can be applied to continuous time observations.

### 3.2 Least Squares and Regression Models

Gauss formulated the principle of least squares at the end of the eighteenth century and used it to determine the orbits of planets. 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.

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

$$y(t) = \varphi_1(t)\theta_1 + \varphi_2(t)\theta_2 + \cdots + \varphi_n(t)\theta_n = \varphi(t)^T\theta$$ \hspace{1cm} (3.1)$$

where \( y \) is the observed variable, \( \theta_1, \theta_2, \ldots, \theta_n \) are unknown parameters, and \( \varphi_1, \varphi_2, \ldots, \varphi_n \) are known functions that may depend on other known variables. The model is indexed by the variable \( t \), which often denotes time. It will be assumed initially that the index set is a discrete set. The
variables $\varphi_i$ are called the regression variables or the regressors, and the model in Eq. (3.1) is also called a regression model. The vectors

$$\varphi^T(t) = [\varphi_1(t) \ \varphi_2(t) \ldots \varphi_n(t)]$$
$$\theta^T = [\theta_1 \ \theta_2 \ldots \theta_n]$$

have also been introduced. 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. (3.1) agree as closely as possible with the measured variables $y(i)$ in the sense of least squares. Since measured variable $y$ is linear in parameters $\theta$ and the least-squares criterion is quadratic, the problem admits an analytical solution. Introduce the notation

$$Y(t) = [y(1) \ y(2) \ldots y(t)]^T$$
$$E(t) = [\varepsilon(1) \ \varepsilon(2) \ldots \varepsilon(t)]^T$$

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

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

where residuals $\varepsilon(i)$, defined by

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

have also been introduced. The least-squares error can then be written as

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

$$= \frac{1}{2} E^T E = \frac{1}{2} \|E\|^2$$

where

$$E = Y - \hat{Y} = Y - \Phi\theta$$

The solution to the least-squares problem is given by the following theorem.
3.2 Least Squares and Regression Models

Theorem 3.1—Least-squares estimation
The function of Eq. (3.2) is minimal for parameters $\hat{\theta}$ such that

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

(3.4)

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

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

(3.5)

Proof: The loss function of Eq. (3.2) can be written as

$$2V(\theta, t) = E^T E = (Y - \Phi \theta)^T (Y - \Phi \theta)$$

$$= Y^T Y - Y^T \Phi \theta - \theta^T \Phi^T Y + \theta^T \Phi^T \Phi \theta$$

Since the matrix $\Phi^T \Phi$ is always nonnegative definite, the function $V$ has a minimum. The loss function is quadratic in $\theta$. By completing the square, it is possible to find the minimum.

$$2V(\theta, t) = Y^T Y - Y^T \Phi \theta - \theta^T \Phi^T Y + \theta^T \Phi^T \Phi \theta$$

$$+ Y^T \Phi (\Phi^T \Phi)^{-1} \Phi^T Y - Y^T \Phi (\Phi^T \Phi)^{-1} \Phi^T Y$$

$$= Y^T (I - \Phi (\Phi^T \Phi)^{-1} \Phi^T) Y$$

$$+ (\theta - (\Phi^T \Phi)^{-1} \Phi^T Y)^T \Phi^T \Phi (\theta - (\Phi^T \Phi)^{-1} \Phi^T Y)$$

(3.6)

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

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

and the theorem is proven.

Remark 1. Equation (3.4) is called the normal equation. Equation (3.5) can be written as

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

(3.7)

Remark 2. The condition that the matrix $\Phi^T \Phi$ is invertible is called an excitation condition. □
Figure 3.1 Geometric interpretation of the least-squares estimate.

Geometric Interpretation

The least-squares problem can be interpreted as a geometric problem in $R^t$, where $t$ is the number of observations. Notice that Eq. (3.3) 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

\[E = Y - \varphi^1 \theta_1 - \varphi^2 \theta_2 - \cdots - \varphi^n \theta_n\]

where $\varphi^i$ are the columns of the matrix $\Phi$. The least-squares problem can thus be interpreted as the problem of finding constants $\theta_1, \ldots, \theta_n$ such that 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 $E = Y - \hat{Y}$. See Fig. 3.1. The vector $E$ is smallest when it is orthogonal to all vectors $\varphi^i$. This gives

\[(\varphi^i)^T (y - \theta_1 \varphi^1 - \theta_2 \varphi^2 - \cdots - \theta_n \varphi^n) = 0 \quad i = 1, \ldots, n\]

which is identical to the normal equations (Eq. 3.4). The vector $\theta$ is unique if the vectors $\varphi^1, \varphi^2, \ldots, \varphi^n$ are linearly independent.
3.2 Least Squares and Regression Models

Statistical Interpretation

The least-squares method can be interpreted in statistical terms. Some assumptions are required for this. Assume that the data has been generated by

\[ y(t) = \varphi^T(t)\theta^0 + e(t) \tag{3.8} \]

where \( \theta^0 \) is the vector of “true” parameters and \( \{e(t), t = 1, 2, \ldots\} \) is a sequence of independent, equally distributed random variables with zero mean values. The following theorem is given without proof.

Theorem 3.2—Statistical properties of least-squares estimation

Consider the estimate in Eq. (3.5) and assume that the data is generated from Eq. (3.8), where \( \{e(t), t = 1, 2, \ldots\} \) is a sequence of independent random variables with zero mean and variance \( \sigma^2 \). If \( \Phi^T\Phi \) is nonsingular, then

(i) \[ E\hat{\theta} = \theta^0 \]

(ii) \[ \text{cov } \hat{\theta} = \sigma^2(\Phi^T\Phi)^{-1} \]

(iii) \[ s^2 = 2V(\hat{\theta}, t)/(t - n) \]

is an unbiased estimate of \( \sigma^2 \)

where \( n \) is the number of parameters in \( \theta^0 \) and \( \hat{\theta} \) and \( t \) is the number of data.

\[ \square \]

Notice that the least-squares estimate corresponds to the case when all measurements have the same precision.

A desirable property of an estimate is that it converges to the true parameter value as the number of observations increases towards 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 3.1—Decrease of variance

Consider the case when the model in Eq. (3.8) only has one parameter. Let \( t \) be the number of observations. It follows from (ii) of Theorem 3.2 that the variance of the estimate is given by

\[ \text{Var } \hat{\theta} = \frac{\sigma^2}{t} \sum_{k=1}^{t} \varphi^2(k) \]

Several different cases can now be considered, depending on the asymptotic behavior of \( \varphi(t) \) for large \( t \).
(a) Assume that \( \varphi(t) \sim e^{-\alpha t}, \alpha > 0 \). The sum then converges and the variance goes to a constant.

(b) Assume that \( \varphi(t) \sim t^{-a}, a > 0 \). Hence

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

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

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

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

The example shows clearly how the precision of the estimate depends on the rate of growth of the regression vector. The variance does not decrease with increasing number of observations if the regression variable increases more slowly than \( t^{-0.5} \). 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 faster.

In the case of several parameters the convergence rates may be different for different parameter combinations.

**Recursive Computations**

In adaptive controllers the observations are obtained sequentially in real time. It is desirable to make the computations recursively in order to save computation time. The computations can be arranged in such a way that the results obtained at time \( t - 1 \) can be used in order to get the estimates at time \( t \). An analogous problem occurs when the number of parameters is not known \textit{a priori}. The least-squares estimate can then be needed for a different number of parameters.

The solution in Eq. (3.5) to the least-squares problem will now be rewritten into a recursive form. Let \( \hat{\theta}(t - 1) \) denote the least-squares estimate based on \( t - 1 \) measurements. It is assumed that the matrix \( \Phi^T \Phi \) is regular for all \( t \). It follows from the definition of \( P(t) \) that

\[
P(t)^{-1} = P(t - 1)^{-1} + \varphi(t)\varphi^T(t) \quad (3.9)
\]
The least-squares estimate $\hat{\theta}(t)$ is then given by the normal equation (Eq. 3.7)

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

Using Eq. (3.7) and Eq. (3.9) gives

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

The estimate at time $t$ can now be written as

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

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

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

where

$$K(t) = P(t)\varphi(t)$$

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

The residual $\varepsilon(t)$ can be interpreted as the prediction error (one step ahead) of $y(t)$ 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. (3.9). The following lemma is useful:

**Lemma 3.1—Matrix inversion lemma**

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

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

**Proof:** By direct substitution we find that

$$(A + BCD) \left( A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \right)$$

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

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

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

$$= I + BCD A^{-1} - BCD A^{-1}$$

$$= I$$

$\Box$
Applying Lemma 3.1 to $P(t)$ and using Eq. (3.9) gives

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

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

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

This implies that

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

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

The calculations are summarized in the following theorem:

**Theorem 3.3—Recursive least-squares estimation (RLS)**

Assume that the matrix $\Phi(t)$ has full rank for all $t > t_0$. The least-squares estimate $\hat{\theta}$ then satisfies the recursive equations

$$
\hat{\theta}(t) = \hat{\theta}(t - 1) + K(t)(y(t) - \varphi^T(t)\hat{\theta}(t - 1)) \quad (3.10)
$$

$$
K(t) = P(t)\varphi(t) = P(t - 1)\varphi(t)(I + \varphi^T(t)P(t - 1)\varphi(t))^{-1} \quad (3.11)
$$

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

$$
= (I - K(t)\varphi^T(t))P(t - 1) \quad (3.12)
$$

**Remark 1.** Equation (3.10) 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. (3.1). 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 estimates of the parameters. 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

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

$$
y(t) = \varphi^T(t)\theta(t) + \epsilon(t) \quad (3.13)
$$
Remark 3. The recursive equations can also be derived by starting with the loss function of Eq. (3.2). Using Eqs. (3.6) and (3.5) gives

\[
2V(\theta, t) = 2V(\theta, t - 1) + \varepsilon^2(\theta, t)
= Y^T \left( I - \Phi(t - 1)(\Phi^T(t - 1)\Phi(t - 1))^{-1}\Phi(t - 1) \right) Y
+ (\theta - \hat{\theta}(t - 1))^T \Phi^T(t - 1)\Phi(t - 1)(\theta - \hat{\theta}(t - 1))
+ (y(t) - \varphi^T(t)\theta)^T (y(t) - \varphi^T(t)\theta)
\] (3.14)

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

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 \( t \) is sufficiently small. In order 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 and determine

\[
P(t_0) = (\Phi^T(t_0)\Phi(t_0))^{-1}
\]
\[
\hat{\theta}(t_0) = P(t_0)\Phi^T(t_0)Y(t_0)
\]

The recursive equations can then be used from \( t > t_0 \). 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

\[
P(t) = (P_0^{-1} + \Phi^T(t)\Phi(t))^{-1}
\]

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

Using the statistical interpretation of the least-squares method, it may be seen that this way of starting the recursion corresponds to the situation when the parameters have an initial covariance proportional to \( P_0 \).
Recursion in the Number of Parameters

When extra parameters are introduced, the vector $\theta$ will have more components, and there will be additional columns in the matrix $\Phi$. The calculations can be arranged such that it is possible to make a recursion in the number of parameters in the model. The recursion involves an inversion of a matrix of the same dimension as the number of added parameters.

Time-varying Parameters

In the least-squares problem (Eq. 3.1) parameters $\theta_i$ 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 situations can be covered by simple extensions of the least-squares method. In one such case parameters are changing abruptly but seldom; in the other case the parameters are changing slowly. The case of abrupt parameter changes can be covered by resetting. The matrix $P$ in the least-squares algorithm (Theorem 3.3) is then periodically reset to $\alpha I$, where $\alpha$ is a large number. A more sophisticated version is to run $n$ estimators in parallel. The estimators are reset, and the appropriate estimator is chosen by a 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. (3.2) with

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

(3.15)

where $\lambda$ is a parameter such that $0 < \lambda \leq 1$. The parameter $\lambda$ is called forgetting factor or discounting factor. The loss function of Eq. (3.15) 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 $\lambda^n$. The method is therefore called exponential forgetting or exponential discounting. Repeating the calculations leading to Theorem 3.3 for the loss function of Eq. (3.15), the following result is obtained.

**Theorem 3.4—RLS with exponential forgetting**

Assume that the matrix $\Phi(t)$ has full rank for $t \geq t_0$. The parameter $\theta$, which minimizes Eq. (3.15), is given recursively by

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

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

$$P(t) = (I - K(t)\varphi^T(t)) P(t-1) / \lambda$$

(3.16)
A disadvantage of exponential forgetting is that data is discounted even if there is no information in the new data (e.g., $P \varphi = 0$). In this case it follows from Eq. (3.16) that the matrix $P$ increases exponentially with rate $\lambda$. Several ways to avoid this are discussed in detail in Chapter 11.

An alternative method of dealing with time-varying parameters is explicitly to assume a mathematical model with such parameters. One possibility is to generalize the filtering interpretation of the least-squares problem given in Remark 2 of Theorem 3.3. Time-varying parameters can be obtained by replacing the first equation of Eq. (3.13) with the model

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

where $\Phi_v$ is a known matrix and $\{v(t)\}$ is discrete-time white noise. 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 3.3 has two state variables, $\hat{\theta}$ and $P$, which must be updated at each step. Updating the vector $\hat{\theta}$ requires $2n$ additions and $2n$ multiplications per step. Updating the matrix $P$ requires $1.5n(n + 1)$ additions, $1.5n(n + 1)$ multiplications, and $0.5n(n + 1)$ divisions per step. For large $n$ the $P$ update clearly 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 $\mathbb{R}^n$. One measurement

$$y(t) = \varphi^T(t) \theta$$  \hspace{1cm} (3.17)

determines the projection of the parameter vector $\theta$ on the vector $\varphi(t)$. From this interpretation it is immediately clear that $n$ measurements, where $\varphi(1), \ldots, \varphi(n)$ span $\mathbb{R}^n$, are required to determine the parameter vector $\theta$ uniquely. Assume that an estimate $\hat{\theta}(t - 1)$ is available and that a new measurement like Eq. (3.17) is obtained. Since the measurement $y(t)$ contains information only in the direction $\varphi(t)$ in parameter space, it is natural to update the estimate as

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

where parameter $\alpha$ is chosen so that

$$y(t) = \varphi^T(t) \hat{\theta}(t) = \varphi^T(t) \hat{\theta}(t - 1) + \alpha \varphi^T(t) \varphi(t)$$
This gives
\[ \alpha = \frac{1}{\varphi^T(t)\varphi(t)} \left( y(t) - \varphi^T(t)\hat{\theta}(t-1) \right) \]

The updating formula thus becomes
\[ \hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\varphi(t)}{\varphi^T(t)\varphi(t)} \left( y(t) - \varphi^T(t)\hat{\theta}(t-1) \right) \]

which is Kaczmarz’s algorithm. Assume that data has been generated by Eq. (3.17) with parameter \( \theta = \theta^0 \). It then follows from Eq. (3.18) that the parameter error
\[ \tilde{\theta} = \theta^0 - \hat{\theta} \]
satisfies the equation
\[ \tilde{\theta}(t) = A(t)\tilde{\theta}(t-1) \]
where
\[ A(t) = I - \frac{\varphi(t)\varphi^T(t)}{\varphi^T(t)\varphi(t)} \]

The matrix \( A(t) \) is a projection matrix. It has one eigenvalue \( \lambda = 0 \) corresponding to the eigenvector \( \varphi(t) \). The other eigenvalues are all equal to 1.

To avoid a potential problem that occurs when \( \varphi(t) = 0 \), the projection algorithm is in practice often modified as follows.

**Algorithm 3.1—Projection algorithm**
\[ \hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\gamma \varphi(t)}{\alpha + \varphi^T(t)\varphi(t)} \left( y(t) - \varphi^T(t)\hat{\theta}(t-1) \right) \]

where \( \alpha \geq 0 \) and \( 0 < \gamma < 2 \). \( \square \)

The projection algorithm assumes that the data is generated by Eq. (3.17) with no error. When the data is generated by Eq. (3.8) with additional random error, a simplified algorithm is given by
\[ \hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi(t) \left( y(t) - \varphi^T(t)\hat{\theta}(t-1) \right) \]

where
\[ P(t) = \left( \sum_{i=1}^{t} \varphi^T(i)\varphi(i) \right)^{-1} \]

This is the stochastic approximation algorithm.
Continuous-time Models

The observations have been indexed by parameter $t$, which belongs to a discrete set. 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. Consider the case with exponential forgetting. The parameter should then be determined such that the criterion

$$V(\theta) = \int_0^t e^{\alpha(t-s)} (y(s) - \varphi^T(s)\theta)^2\,ds$$  \hspace{1cm} (3.22)

is minimized. A straightforward calculation shows that the criterion is minimized if

$$\left( \int_0^t e^{-\alpha(t-s)} \varphi(s)\varphi^T(s)\,ds \right) \dot{\theta}(t) = \int_0^t e^{-\alpha(t-s)} \varphi(s)y(s)\,ds$$  \hspace{1cm} (3.23)

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

$$R(t) = \int_0^t e^{-\alpha(t-s)} \varphi(s)\varphi^T(s)\,ds$$  \hspace{1cm} (3.24)

is positive definite. It is also possible to obtain real-time equations for the estimate. They will be differential equations, since the parameter $t$ is continuous. The estimate is given by the following theorem.

**Theorem 3.5—Continuous-time least-squares estimation**

Consider $t$ such that the matrix $R(t)$ given by Eq. (3.24) is invertible. The estimate that minimizes Eq. (3.22) satisfies

$$\frac{d\hat{\theta}}{dt} = P(t)\varphi(t)e(t)$$  \hspace{1cm} (3.25)

$$e(t) = y(t) - \varphi^T(t)\dot{\theta}(t)$$  \hspace{1cm} (3.26)

$$\frac{dP(t)}{dt} = \alpha P(t) - P(t)\varphi(t)\varphi^T(t)P(t)$$  \hspace{1cm} (3.27)

**Proof:** The theorem is proven by differentiating Eq. (3.22). \hfill \Box

**Remark.** Notice that the matrix $R(t)$ satisfies

$$\frac{dR}{dt} = -\alpha R + \varphi\varphi^T$$
There are also continuous-time versions of the simplified algorithms. The projection algorithm is given by Eq. (3.25) with

\[ P(t) = \left( \int_0^t \varphi^T(s)\varphi(s) \, ds \right)^{-1} \]

### 3.3 Estimating Parameters in Dynamical Systems

It will now be shown 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. Some extensions of the least-squares method will also be given.

**Finite Impulse Response (FIR) Models**

A linear time-invariant dynamical system is uniquely characterized by its impulse response. The impulse response is, however, infinite dimensional even for discrete-time systems. For stable systems the impulse response will go to zero and may then be truncated. This results in so-called finite impulse response (FIR) models or transversal filter. Such a model can be described by the equation

\[ y(t) = b_1 u(t - 1) + b_2 u(t - 2) + \cdots + b_n u(t - n) \quad (3.28) \]

or

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

where

\[ \theta^T = [b_1 \ldots b_n] \]

\[ \varphi^T(t - 1) = [u(t - 1) \ldots u(t - n)] \]

This model is identical to the regression model of Eq. (3.1), 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. (3.28) clearly fits the least-squares formulation. Notice, however, that a large number of parameters may be required if the sampling interval is short compared to the slowest time constant of the system. Notice that the parameter estimator can be represented by the
block diagram in Fig. 3.2. 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, the recursive estimator can also be interpreted as an adaptive filter.

**Persistent Excitation**

It is obvious that the parameters of the model of Eq. (3.28) cannot be determined unless some conditions are imposed on the input signal. For example, consider the case when the input is zero. It follows from the condition for uniqueness of the least-squares estimate given by Theorem 3.1 that the minimum is unique if the matrix

\[
\Phi^T \Phi = \begin{pmatrix}
\sum_{n+1}^{t} u^2(k-1) & \sum_{n+1}^{t} u(k-1)u(k-2) & \cdots & \sum_{n+1}^{t} u(k-1)u(k-n) \\
\sum_{n+1}^{t} u(k-1)u(k-2) & \sum_{n+1}^{t} u^2(k-2) & \cdots & \sum_{n+1}^{t} u(k-2)u(k-n) \\
\vdots & \vdots & \ddots & \vdots \\
\sum_{n+1}^{t} u(k-1)u(k-n) & \sum_{n+1}^{t} u(k-n) & \cdots & \sum_{n+1}^{t} u^2(k-n)
\end{pmatrix}
\]

(3.29)

has full rank. This condition is called an excitation condition. For long data sets the end effects are negligible, and all sums in Eq. (3.29) can be
taken from 1 to \( t \). We then get

\[
C_n = \lim_{t \to \infty} \frac{1}{t} \Phi^T \Phi = \begin{pmatrix}
    c(0) & c(1) & \ldots & c(n-1) \\
    c(1) & c(0) & \ldots & c(n-2) \\
    \vdots & \vdots & \ddots & \vdots \\
    c(n-1) & c(n-2) & \ldots & c(0)
\end{pmatrix}
\] (3.30)

where \( c(k) \) are the empirical covariances of the input, i.e.,

\[
c(k) = \lim_{t \to \infty} \frac{1}{t} \sum_{i=1}^{t} u(i) u(i-k)
\]

For long data sets the condition for uniqueness can thus be expressed as the matrix in Eq. (3.30) being positive definite. This leads to the following definition.

**Definition 3.1—Persistent excitation**
A square summable signal \( u \) is called **persistently exciting** (PE) of order \( n \) if the matrix \( C_n \) given by Eq. (3.30) is positive definite.

The following result can thus be established.

**Theorem 3.6—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 3.1 and Theorem 3.2.

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

\[
A(q)y(t) = B(q)u(t)
\] (3.31)

or

\[
y(t) + a_1 y(t-1) + \cdots + a_n y(t-n) = b_1 u(t-1) + \cdots + b_n u(t-n)
\]

Assume that the polynomials \( A \) and \( B \) are of order \( n \) and \( n - 1 \), respectively. Further 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

\[
\theta^T = [a_1 \ldots a_n \ b_1 \ldots b_n]
\] (3.32)
and the regression vector

$$\varphi^T(t-1) = [-y(t-1) \ldots -y(t-n) \ u(t-1) \ldots u(t-n)]$$

Notice that the output signal appears delayed in the regression vector. The model is therefore called an auto-regression. The way the elements are ordered in matrix $\theta$ is, of course, arbitrary. Later, when dealing with adaptive control, it will be natural to have the term $b_1$ first in the $\theta$ vector, because it will be treated in a special way. The convention that the time index of the $\varphi$ vector will refer to the time at which 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 least squares. The matrix $\Phi$ is furthermore given by

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

Using the statistical interpretation of the least-squares estimate given by Theorem 3.2, 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. (3.31). Compare with Eq. (3.8). 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. The method obtained is then called an output error method. To describe such a method, let $u$ be the input and $x$ the output of a system with the input-output relation

$$x(t) + a_1 x(t-1) + \cdots + a_n x(t-n) = b_1 u(t-1) + \cdots + b_n u(t-n)$$

Determine the parameters that minimize the criterion

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

where $y(t) = x(t) + e(t)$. This problem is clearly a least-squares problem whose solution is given by

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi(t-1)e(t)$$
where
\[ \varphi^T(t - 1) = [-x(t - 1) \ldots - x(t - n) \ u(t - 1) \ldots u(t - n)] \]
\[ \varepsilon(t) = y(t) - \varphi^T(t - 1)\hat{\theta}(t - 1) \]

Compare with Theorem 3.1. The recursive estimator obtained can be represented by the block diagram in Fig. 3.3.

**Continuous-time Transfer Functions**

The least-squares model can also be applied to estimate parameters in continuous-time transfer functions. For example, consider a continuous-time model of the form

\[ \frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_n y = b_1 \frac{d^{n-1} u}{dt^{n-1}} + \cdots + b_n u \]

which can also be written as

\[ A(p)y(t) = B(p)u(t) \quad (3.33) \]

where \( A(p) \) and \( B(p) \) are polynomials in the differential operator \( p \). In most cases we cannot conveniently compute \( p^n y(t) \), because it would involve taking \( n \) derivatives of a signal. The model of Eq. (3.33) is therefore rewritten as

\[ A(p)y_f(t) = B(p)u_f(t) \quad (3.34) \]

where

\[ y_f(t) = F(p)y(t) \]
\[ u_f(t) = F(p)u(t) \]
and $F(p)$ is a transfer function with a pole excess at least equal to $n$. Introducing

$$
\theta = [a_1 \ldots a_n \ b_1 \ldots b_n]^T \\
\varphi_f(t) = [-p^{n-1}y_f \ldots -y_f \quad p^{n-1}u_f \ldots u_f] \\
= [-p^{n-1}F(p)y \ldots -F(p)y \quad p^{n-1}F(p)u \ldots F(p)u]
$$

the model Eq. (3.34) can be written as

$$y_f(t) = \varphi_f^T(t)\theta$$

Standard least squares can now be applied, since this is a regression model.

**Nonlinear Models**

Least squares can also be applied to certain nonlinear models. The essential factor is that the models are linear in the parameters so that they can be written as regression models. An example illustrates the idea.

**Example 3.2—Nonlinear system**

Consider the model

$$y(t) + ay(t - 1) = b_1u(t - 1) + b_2u^2(t - 1)$$

By introducing

$$\theta = [a \ b_1 \ b_2]^T$$

and

$$\varphi^T(t) = [-y(t) \ u(t) \ u^2(t)]$$

the model can be written as

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

**Stochastic Models**

The least-squares estimate is biased when applied to data generated by Eq. (3.13), where the errors are correlated. One way to avoid the difficulty is to sample the models at widely spaced time intervals. Another possibility is to model the correlation of the disturbances and to estimate the parameters describing the correlations. Consider the model

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

where $A(q)$, $B(q)$, and $C(q)$ are polynomials in the shift operator and \{e(t)\} is white noise. The parameters of the polynomial $C$ describe the
correlation of the disturbance. The model of Eq. (3.35) 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

\[
\theta = [a_1 \ldots a_n \ b_1 \ldots b_n \ c_1 \ldots c_n]
\]

\[
\varphi^T(t) = [-y(t) \ldots -y(t-n+1) \ u(t) \ldots u(t-n+1) \ \varepsilon(t) \ldots \varepsilon(t-n+1)]
\]

where

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

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

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

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

\[
\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi(t-1)e(t)
\]

\[
P^{-1}(t) = P^{-1}(t-1) + \varphi(t-1)\varphi^T(t-1)
\]

(3.36)

Compare with Theorem 3.3. Another method of estimating the parameters in Eq. (3.35) is to make a recursive approximation of the maximum likelihood estimate or to use a prediction error method. The estimate is then given by Eq. (3.36) with residual

\[
\hat{C}(q)e(t) = \hat{A}(q)y(t) - \hat{B}(q)u(t)
\]

(3.37)

and regression vector \(\varphi\) replaced by \(\varphi_f\), where

\[
\hat{C}(q)\varphi_f(t) = \varphi(t)
\]

(3.38)

The most recent estimates should be used in these updates. The method obtained is then not truly recursive, since Eqs. (3.38) and (3.37) have to be solved from \(t = 1\) for each measurement. The following approximations can be made:

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

The algorithm obtained is then called the recursive maximum likelihood method (RML).
3.4 Experimental Conditions

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

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

It is sometimes advantageous to model a stochastic system with

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

instead of Eq. (3.35). Recursive parameters for this model can be derived in the same way as for Eq. (3.35).

The asymptotic properties of the extended least-squares method and the recursive maximum likelihood method are established in the literature on system identification.

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 $\theta$, $\varphi$, and $\varepsilon$ are different for the different methods.

3.4 Experimental Conditions

Certain experimental conditions are required in order to perform system identification. When executing 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. For example, consider the model

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

It is intuitively clear that no information about the polynomials $A$ and $B$ can be obtained when $u = 0$; similarly, no information about $C$ and $D$ can be obtained unless $e \neq 0$. The notion of persistent excitation was introduced in the previous section as a condition on system inputs. This notion will be explored more in this section. In adaptive systems identification is often carried out in closed loop. This may also cause special difficulties.
Persistent Excitation

The following result gives additional insight into the notion of persistent excitation.

**Theorem 3.7—Persistently exciting signals**

A square summable signal \( u \) is persistently exciting of order \( n \) if and only if

\[
\lim_{t \to \infty} \frac{1}{t} \left( \sum_{k=1}^{t} A(q)u(k) \right)^2 > 0
\]  

(3.39)

for all 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

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

where \( C_n \) is the matrix given by Eq. (3.30). 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. \( \square \)

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

**Example 3.3—Pulse**

It follows from Eq. (3.39) that \( C_n \to 0 \) for all \( n \) if \( u \) is a pulse. A pulse thus is not PE for any \( n \). \( \square \)

**Example 3.4—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 3.5—Sinusoid
Let \( u(t) = \sin \omega t \). It follows that
\[
(q^2 - 2q \cos \omega + 1) u(t) = 0
\]
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 3.6—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 3.7—Random signals
Consider a mean square ergodic stochastic process with nonvanishing prediction error. Since the signal can not be predicted, it follows that Eq. (3.39) holds. The signal is thus PE of any order.

Example 3.8—Frequency domain characterization
Consider a quasi-stationary signal \( u(t) \) with spectrum \( \Phi_u(\omega) \). It follows from Parseval’s theorem that
\[
\lim_{t \to \infty} \frac{1}{t} \left( \sum_{k=1}^{t} A(q) u(k) \right)^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |A(e^{i\omega})|^2 \Phi_u(\omega) \, d\omega \quad (3.40)
\]
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. (3.40) will thus be positive if \( \Phi_u(\omega) \neq 0 \) for at least \( n \) points in the interval \(-\pi \leq \omega \leq \pi\). 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 3.8—PE of filtered signals

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

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

is persistently exciting of order $n - m$. Assuming that $A$ is stable, the signal $v$ defined by

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

is persistently exciting of order $n$. $\square$

Identification in Closed Loop

In adaptive control systems 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. (3.31). The matrix $\Phi$ is then

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

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

Example 3.9—Loss of identifiability due to feedback

Consider a system described by

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

(3.41)

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

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

(3.42)

Multiplying Eq. (3.42) by $\alpha$ and adding to Eq. (3.41) gives

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

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 with slope $1/k$ in parameter space (see Fig. 3.4). The loss-function has the same value for all parameters on this line.

The problem with lack of identifiability due to feedback will disappear if a linear feedback of sufficiently high order is used or if the feedback gain is time variable. For example, in Example 3.9 it is sufficient to have a feedback of the form

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

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 3.9 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 will decrease as $1/t$, but more complex behavior is also possible. The following example shows what can happen.

**Example 3.10—Convergence rate**
Consider data generated by Eq. (3.41) with a feedback of the form

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

(3.43)
where \( \{v(t)\} \) is a sequence of independent random variables that are also independent of \( \{e(t)\} \). With the feedback law of Eq. (3.43) 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 information matrix \( \Phi^T \Phi \) of the estimation problem is

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

It follows that

\[
\sum_{j=1}^{t} y(j)u(j) = -k \sum_{j=1}^{t} y_j^2(j) - k \sum_{j=1}^{t} \frac{v(j)y_j^2(j)}{\sqrt{j}} \approx -k \sum_{j=1}^{t} y_j^2(j) \approx -k \sigma_y^2
\]

\[
\sum_{j=1}^{t} u_j^2(j) = k^2 \left( \sum_{j=1}^{t} y_j^2(j) + 2 \sum_{j=1}^{t} \frac{v(j)y_j^2(j)}{\sqrt{j}} + \sum_{j=1}^{t} \frac{v_j^2(j)y_j^2(j)}{j} \right)
\]

\[
\approx k^2 \left( \sum_{j=1}^{t} y_j^2(j) + \sum_{j=1}^{t} \frac{v_j^2(j)y_j^2(j)}{j} \right) \approx k^2 \sigma_y^2 (t + \sigma_v^2 \log t)
\]

Hence for large \( t \)

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

The covariance matrix of the estimate is thus

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

It now follows that

\[
\text{Var} \left( \hat{a} - kb \right) = \sigma_e^2 (1 - k) (\Phi^T \Phi)^{-1} (1 - k)^T \approx \frac{\sigma_e^2}{t \sigma_y^2}
\]

\[
\text{Var} \left( k\hat{a} + \hat{b} \right) = \sigma_e^2 (k) (\Phi^T \Phi)^{-1} (k \ 1)^T \approx \frac{\sigma_e^2}{\sigma_y^2 \sigma_v^2 \log t}
\]
The estimate will thus approach the line \( b = b^0 + (a - a^0)/k \) at the rate \( t^{-1} \). The estimate will then converge towards the correct values at the rate \( (\log t)^{-1} \). The convergence along the line is much slower than towards the line.

3.5 Properties of Recursive Estimators

The key problem in the analysis of recursive estimators is to find the properties of the estimators. The result depends strongly on how the data supplied to the estimator is generated—for example, whether the data is deterministic or stochastic. The properties of the input signal are particularly important, as shown in the previous section. In adaptive systems in which data is generated by feedback there may be complex interactions between the feedback law and the properties of the estimator.

The analysis of the estimator can be carried out completely for finite impulse response models. However, only asymptotic results for large data sets are available when estimating parameters of transfer function models.

The deterministic case, in which data is generated from a model that is compatible with the model used to derive the estimator, is particularly simple. In this case it is possible to derive general properties of the estimators.

A Projection Algorithm

The properties of the projection algorithm of Eq. (3.19) will now be investigated in the ideal case where data is generated by the model

\[
y(t) = \varphi^T(t) \theta^0
\]

(3.44)

We have the following result.

**Theorem 3.9—Projection algorithm properties**

Let the estimator

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

(3.45)

with \( \alpha \geq 0 \) and \( 0 < \gamma < 2 \), be applied to data generated by Eq. (3.44). It then follows that

(i) \( \| \hat{\theta}(t) - \theta^0 \| \leq \| \hat{\theta}(t - 1) - \theta^0 \| \leq \| \hat{\theta}(0) - \theta^0 \|, \quad t \geq 1 \)

(ii) \( \lim_{t \to \infty} \frac{e(t)}{\sqrt{\alpha + \varphi^T(t) \varphi(t)}} = 0 \) where \( e(t) = \varphi^T(t) \left( \hat{\theta}(t - 1) - \theta^0 \right) \)

(iii) \( \lim_{t \to \infty} \| \hat{\theta}(t) - \hat{\theta}(t - k) \| = 0 \) 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 \( \theta^0 \) from both sides of Eq. (3.45) and taking the norm gives

\[
V(t) - V(t - 1) = 2 \frac{\gamma\varphi^T(t)\tilde{\theta}(t - 1)e(t)}{\alpha + \varphi^T(t)\varphi(t)} + \frac{\gamma^2\varphi^T(t)\varphi(t)e^2(t)}{(\alpha + \varphi^T(t)\varphi(t))^2}
\]

\[
= \chi(t)\frac{\gamma e^2(t)}{\alpha + \varphi^T(t)\varphi(t)}
\]

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 above equation that

\[
V(t) = V(0) + \sum_{k=1}^{t} \chi(k)\frac{\gamma e^2(k)}{\alpha + \varphi^T(k)\varphi(k)}
\]

Hence

\[
\sum_{k=1}^{t} \frac{\gamma e^2(k)}{\alpha + \varphi^T(k)\varphi(k)} \leq \frac{1}{\delta} [V(0) - V(t)]
\]

Since \( 0 \leq V(t) \leq V(0) \), property (ii) now follows, because the normalized error

\[
e(t) \over \sqrt{\alpha + \varphi^T(t)\varphi(t)}
\]

is in \( l_2 \). It follows from Eq. (3.45) that

\[
\|\hat{\theta}(t) - \hat{\theta}(t - 1)\|^2 = \frac{\gamma^2\varphi^T(t)\varphi(t)e^2(t)}{(\alpha + \varphi^T(t)\varphi(t))^2}
\]

\[
= \frac{\gamma^2 e^2(t)}{\alpha + \varphi^T(t)\varphi(t)} \left( 1 - \frac{\alpha}{\alpha + \varphi^T(t)\varphi(t)} \right)
\]

It follows from (ii) that the right-hand side of the above equation goes to
zero as $t \to \infty$ if $\alpha > 0$. Hence

$$
\| \hat{\theta}(t) - \hat{\theta}(t-k) \|^2 = \left\| \sum_{i=1}^{k} \hat{\theta}(t-i+1) - \hat{\theta}(t-i) \right\|^2
\leq \sum_{i=1}^{k} \| \hat{\theta}(t-i+1) - \hat{\theta}(t-1) \|^2
$$

where the right-hand side goes to zero as $t \to \infty$ for finite $k$.

**Remark 1.** Notice that the result holds for all input sequences $\{ u(t) \}$. It applies to open-loop control as well as any closed-loop control.

**Remark 2.** Notice that the result does not imply that the estimates $\hat{\theta}(t)$ converge.

If the input and output signals of the system can be shown to be bounded, then $\varphi$ is bounded. If $\varphi(t)$ is bounded for all $t$, it follows from property (ii) of Theorem 3.9 that the prediction error $e(t)$ goes to zero as $t \to \infty$. The result implies that the projection algorithm gives estimates that converge to the true parameter as the number of observations increases if $\varphi^T \varphi > 0$.

**Recursive Least Squares**

Results similar to Theorem 3.9 can also be established for the least-squares algorithm and several of its variants. The key is to replace function $V(t)$ in Theorem 3.9 by

$$
V(t) = \hat{\theta}^T(t) P^{-1}(t) \hat{\theta}(t)
$$

For recursive least squares we can also make use of the observation that the estimate minimizes the loss function

$$
V_1(\theta, t) = \sum_{k=1}^{t} \left( \varphi^T(k)(\theta - \theta_0) \right)^2 + (\theta - \theta(0))^T P^{-1}(0)(\theta - \theta(0))
$$

Straightforward calculations show that the minimum is given by

$$
\left( P^{-1}(0) + \sum_{k=1}^{t} \varphi(k)\varphi^T(k) \right) \hat{\theta}(t)
= \sum_{k=1}^{t} \varphi(k)\varphi^T(k)\theta_0 + P_0^{-1}(0)\theta(0) \tag{3.46}
$$

From this analysis we obtain the following result.
Theorem 3.10—Property of RLS
Let the recursive least squares be applied to data generated by Eq. (3.44). Let $P(0)$ be positive definite and $\theta(0)$ bounded. Assume that

$$
\sum_{k=1}^{t} \varphi(k) \varphi^T(k) \geq \alpha(t) I
$$

where $\alpha(t)$ goes to infinity. Then the estimate converges to $\theta^0$. □

Transfer Function Models
The properties of estimates of parameters of 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

$$
A^0(q)y(t) = B^0(q)u(t) + e(t + n)
$$

(3.47)

where $A^0$ and $B^0$ are relative prime. If $e = 0$ and $\deg A > \deg A^0$ and $\deg B > \deg B^0$, it follows from Theorem 3.1 that the estimate is not unique, because the columns of the matrix $\Phi$ are linearly dependent. We have, however, the following result.

Theorem 3.11—Transfer function estimation
Consider data generated by the model of Eq. (3.47), with $A^0$ stable and $e = 0$. Let the parameters of the model be fitted by least squares. Assume that the input $u$ is persistently exciting of order $\deg A + \deg B + 1$ and that $\deg A = \deg A^0$, then $\lim_{t \to \infty} \Phi^T \Phi / t$ is positive definite.

Proof: Consider

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

Introduce

$$
v(t) = \varphi^T(t + n - 1) \theta = B(q)u(t) - (A(q) - q^n) y(t)
$$

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

$$
= \left\{ A^0(q)B(q) - (A(q) - q^n)B^0(q) \right\} \frac{1}{A^0(q)} u(t)
$$

Since $A^0$ is stable, it follows from Theorem 3.8 that the signal $A^0(q)^{-1} u(t)$ is persistently exciting of order $\deg A + \deg B + 1$. Since the polynomial in curly brackets has a degree lower than $\deg A + \deg B$, 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 proven.

Remark 1. Notice that $\deg A + \deg B + 1$ is equal to the number of parameters in the model of Eq. (3.47). The order of PE required is thus equal to the number of estimated parameters.

Remark 2. If the data is generated by Eq. (3.47), 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 3.1 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 only be established asymptotically for large number of observations. The following result can be established.

**Theorem 3.12—Convergence of RLS**
Let the least-squares method for estimation parameters of a transfer function be applied to data generated by the model of Eq. (3.47) where $\{e(t)\}$ is a sequence of uncorrelated random variables with zero mean and variance $\sigma^2$. Assume that the input signal is persistently exciting of order $\deg A + \deg B + 1$. Then

(i) \[ \hat{\theta}(t) \to \theta^0 \text{ as } t \to \infty \]

(ii) \[ \text{Var} (\hat{\theta} - \theta^0) \approx \frac{\sigma^2}{t} \left( \lim_{t \to \infty} \frac{1}{t} \phi^T \phi \right)^{-1} \]

Remark. The estimate will not converge to the true parameters if $e(t)$ is correlated with $e(s)$ for $t \neq s$.

**3.6 Implementation Issues**

Several issues must be considered when implementing recursive estimation algorithms. These will be dealt with in detail in Chapter 11, but some
Input \( u, y: \text{real} \)
Parameter \( \lambda: \text{real} \)
State \( \varphi, \theta: \text{vector} \)
\( P: \text{symmetric matrix} \)

Local variables \( w: \text{vector}, \text{den}: \text{real} \)

"Compute residual 
\( e = y - \varphi^T \theta \)
"Update estimate 
\( w = P \varphi \)
\( \text{den} = w^T \varphi + \lambda \)
\( \theta = \theta - w e / \text{den} \)
"Update covariance matrix 
\( P = [P - w w^T / \text{den}] / \lambda \)
"Update regression vectors 
\( \varphi = \text{shift}(\varphi) \)
\( \varphi(1) = -y \)
\( \varphi(n+1) = u \)

**Listing 3.1** Basic recursive least-squares algorithm with exponential forgetting.

understanding of these problems is needed to experiment with adaptive control. We will therefore discuss some problems in this section.

**Basic Code**

The recursive least-squares estimate with exponential forgetting is given by Theorem 3.4. The state of the estimator is the vector \( \hat{\theta} \) and the matrix \( P \). When applied to estimation of parameters of a transfer function model it is also natural to view the regression vector as a state variable for storing past values of the input and the output. The code given in Listing 3.1 is a straightforward implementation of the algorithm. The listing assumes that vector matrix and shift operations are available. If the vector \( \varphi \) is 
\[
\varphi = [\varphi_1 \, \varphi_2 \ldots \varphi_k]
\]
then shift \((\varphi)\) is defined as 
\[
\text{shift}(\varphi) = [0 \, \varphi_1 \ldots \varphi_{k-1}]
\]
It is straightforward to rewrite the algorithm in any standard computer language.

**Robustness**

In practice the algorithm given in Listing 3.1 has several drawbacks. It is a direct consequence of the least-squares formulation that a single large
error will have a drastic influence on the result. Another way to express this is that the probability of large errors is very small under Gaussian assumptions. Estimators with very different properties are obtained if it is assumed that the probability for large errors is not negligible. Without going into technicalities we remark that the estimators will be replaced by equations such as

\[
\theta(t) = \theta(t - 1) + P(t)\varphi(t - 1)f(\varepsilon(t))\]

\[
\frac{d\theta}{dt} = P\varphi f(\varepsilon)
\]

where the function \( f(\varepsilon) \) is linear for small \( \varepsilon \) but increases more slowly than linear for large \( \varepsilon \). A typical example is

\[
f(\varepsilon) = \frac{\varepsilon}{1 + a|\varepsilon|}
\]

The net effect is to decrease the consequences of large errors. The estimators are then called \textit{robust}.

\section*{Square Root Algorithms}

It is well known in numerical analysis that considerable accuracy may be lost when a least-squares problem is solved by forming and solving the normal equations. The reason is that the measured values are squared unnecessarily. The following procedure for solving the least-squares problem is much better conditioned numerically. Start with Eq. (3.3):

\[
E = Y - \Phi\theta
\]

An orthogonal transformation \( Q \) does not change the Euclidian norm of the error

\[
\tilde{E} = QE = QY - Q\Phi\theta
\]

Choose the transformation \( Q \) so that \( Q\Phi \) is upper triangular. The above equation then becomes

\[
\begin{pmatrix}
\tilde{e}^1 \\
\tilde{e}^2
\end{pmatrix} = 
\begin{pmatrix}
\tilde{y}^1 \\
\tilde{y}^2
\end{pmatrix} - 
\begin{pmatrix}
\tilde{\Phi}_1 \\
0
\end{pmatrix} \theta
\]

where \( \tilde{\Phi}_1 \) is upper triangular. It then follows that the least-squares estimate is given by

\[
\tilde{\Phi}_1 \theta = \tilde{y}^1
\]

and the error is \((\tilde{e}^2)^T\tilde{e}^2\). This way of computing the estimate is much more accurate than solving the normal equation, particularly if \( \|E\| \ll \|Y\| \). The method based on orthogonal transformation is called a \textit{square}
root method, because it works with \( \Phi \) or the square root of \( \Phi^T \Phi \). There are several numerical methods that can be used to find an orthogonal transformation \( Q \), e.g., Householder transformations or the QR method. These methods will not be discussed further, because we are primarily interested in recursive methods.

**Representation of Conditional Mean Values**

Recursive square root methods can naturally be explained using probabilistic arguments. Some preliminary results on conditional mean values for Gaussian random variables will first be developed. We have the following result.

**Theorem 3.13**—Conditional mean values and covariances

Let the vectors \( x \) and \( y \) be jointly Gaussian random variables with mean values

\[
E \left( \begin{array}{c} y \\ x \end{array} \right) = \left( \begin{array}{c} m_y \\ m_x \end{array} \right)
\]

and covariance

\[
\text{cov} \left( \begin{array}{c} y \\ x \end{array} \right) = \left( \begin{array}{cc} R_y & R_{yx} \\ R_{xy} & R_x \end{array} \right) = R
\]

where \( R_{xy} = R_{yx}^T \). The conditional mean value of \( x \) given \( y \) is Gaussian with mean

\[
E(x|y) = m_x + R_{xy} R_y^{-1} (y - m_y)
\]

and covariance

\[
\text{cov}(x|y) = R_{x|y} = R_x - R_{xy} R_y^{-1} R_{yx}
\]

A nonnegative matrix \( R \) can be decomposed as

\[
R = \rho \left( \begin{array}{cc} 1 & 0 \\ K & L_x \end{array} \right) \left( \begin{array}{cc} D_y & 0 \\ 0 & D_x \end{array} \right) \left( \begin{array}{cc} 1 & 0 \\ K & L_x \end{array} \right)^T
\]

where \( D_x \) and \( D_y \) are diagonal matrices and \( L_x \) is lower triangular. Then

\[
R_{xy} R_y^{-1} = K
\]

and

\[
R_{x|y} = \rho L_x D_x L_x^T
\]

**Proof:** It will first be shown that the vector \( z \) defined by

\[
z = x - m_x - R_{xy} R_y^{-1} (y - m_y)
\]
has zero mean, is independent of \( y \), and has the covariance

\[
R_z = R_x - R_{xy} R_y^{-1} R_{yx}
\]  

(3.56)

The mean value is zero. Furthermore,

\[
Ez(y - m_y)^T = E \{(x - m_x)(y - m_y)^T - R_{xy} R_y^{-1} (y - m_y)(y - m_y)^T\} = R_{xy} - R_{xy} R_y^{-1} R_y = 0
\]

The variables \( z \) and \( y \) are thus uncorrelated. Since they are Gaussian, they are also independent. It now follows that

\[
\begin{pmatrix}
  y - m_y \\
  x - m_x
\end{pmatrix} =
\begin{pmatrix}
  I & 0 \\
  R_{xy} R_y^{-1} & I
\end{pmatrix}
\begin{pmatrix}
  y - m_y \\
  z
\end{pmatrix}
\]

The joint density function of \( x \) and \( y \) is

\[
f(x, y) = (2\pi)^{-(n+p)/2}(\det R)^{-1/2} \exp\left\{-\frac{1}{2} (z^T R_z^{-1} z + (y - m_y)^T R_y^{-1} (y - m_y))\right\}
\]

The density function of \( y \) is

\[
f(y) = (2\pi)^{-p/2}(\det R_y)^{-1/2} \exp\left\{-\frac{1}{2} (y - m_y)^T R_y^{-1} (y - m_y)\right\}
\]

where \( p \) is the dimension of \( y \). The conditional density is then

\[
f(x|y) = \frac{f(x, y)}{f(y)} = (2\pi)^{-n/2}(\det R_y)^{1/2}(\det R)^{-1/2} \exp\left\{-\frac{1}{2} z^T R_z^{-1} z\right\}
\]

where \( n \) is the dimension of \( x \). But

\[
\det R = \det \begin{pmatrix}
  R_y & R_{yx} \\
  R_{xy} & R_x
\end{pmatrix} = \det \begin{pmatrix}
  R_y & R_{yx} \\
  0 & R_x - R_{xy} R_y^{-1} R_{yx}
\end{pmatrix}
\]

\[
= \det R_y \cdot \det (R_x - R_{xy} R_y^{-1} R_{yx}) = \det R_y \cdot \det R_z
\]

Hence

\[
f(x|y) = (2\pi)^{-n/2}(\det R_z)^{-1/2} e^{-\frac{1}{2} z^T R_z^{-1} z}
\]

where \( z \) is given by Eq. (3.55) and \( R_z \) by Eq. (3.56).
Chapter 3  Real-time Parameter Estimation

The first part of the theorem is thus proven. To show the second part, notice that Eq. (3.52)

\[ R = \rho \begin{pmatrix} D_y & D_y K^T \\ KD_y & L_x D_x L_x^T + KD_y K^T \end{pmatrix} \]

Identification of the different terms gives

\[
R_y = \rho D_y \\
R_{xy} = \rho K D_y \\
R_x = \rho (L_x D_x L_x^T + KD_y K^T)
\]

Hence

\[ R_{xy} R_y^{-1} = K \]

and

\[ R_{x|y} = R_x - R_{xy} R_y^{-1} R_{yx} = \rho L_x D_x L_x^T \]

Remark. It follows from the theorem that the calculation of the conditional mean of a Gaussian random variable is equivalent to transforming the joint covariance matrix of the variables to the form of Eq. (3.52). Notice that this form may be viewed as a square root representation of R. □

Application to Recursive Estimation

The basic step in recursive estimation can be described as follows. Let \( \theta \) be Gaussian \( N(\theta^0, P) \). Assume that a linear observation

\[ y = \varphi^T \theta + \epsilon \]

is made, where \( \epsilon \) is normal \( N(0, \sigma^2) \). The new estimate is then given as the conditional mean \( E(\theta|y) \). The joint covariance matrix of \( y \) and \( \theta \) is

\[ R = \begin{pmatrix} \varphi^T P \varphi & \varphi^T P \\ P \varphi & P \end{pmatrix} + \begin{pmatrix} \sigma^2 & 0 \\ 0 & 0 \end{pmatrix} \]

The symmetric nonnegative matrix \( P \) has a decomposition \( P = LDL^T \), where \( L \) is a lower triangular matrix with unit diagonal and \( D \) a nonnegative diagonal matrix. The matrix \( R \) can then be written as

\[
R = \begin{pmatrix} \varphi^T L D L^T \varphi + \sigma^2 & \varphi^T L D L^T \\ L D L^T \varphi & L D L^T \end{pmatrix} = \begin{pmatrix} 1 & \varphi^T L \\ 0 & L \end{pmatrix} \begin{pmatrix} \sigma^2 & 0 \\ 0 & D \end{pmatrix} \begin{pmatrix} 1 & 0 \\ L^T \varphi & L^T \end{pmatrix}
\]

(3.57)
If this matrix can be transformed to
\[ R = \begin{pmatrix} 1 & 0 \\ K & \tilde{L} \end{pmatrix} \begin{pmatrix} \hat{\sigma}^2 & 0 \\ 0 & \tilde{D} \end{pmatrix} \begin{pmatrix} 1 & K^T \\ 0 & \tilde{L}^T \end{pmatrix} \] (3.58)

the last theorem can be used to obtain the recursive estimate as
\[ \hat{\theta} = \theta^0 + K(y - \varphi^T \theta) \]

with covariance
\[ P = \tilde{L} \tilde{D} \tilde{L}^T \]

The algorithm can thus be described as follows.

**Algorithm 3.2—Square root RLS**

1. Start with \( L \) and \( D \) as a representation of \( P \).
2. Form the matrix of Eq. (3.57), where \( \varphi \) is the regression vector.
3. Reduce this to the lower triangular form of Eq. (3.58).
4. The updating gain is \( K \), and the new \( P \) is represented by \( \tilde{L} \) and \( \tilde{D} \).

It now remains to find the appropriate transformation matrices. A convenient method is dyadic decomposition.

**Dyadic Decomposition**

Given vectors
\[ a = [1 \ a_2 \ldots a_n]^T \]
\[ b = [b_1 \ b_2 \ldots b_n]^T \]

and scalars \( \alpha \) and \( \beta \), find new vectors
\[ \tilde{a} = [1 \ \tilde{a}_2 \ldots \tilde{a}_n]^T \]
\[ \tilde{b} = [0 \ \tilde{b}_2 \ldots \tilde{b}_n]^T \]

such that
\[ \alpha aa^T + \beta bb^T = \tilde{a} \tilde{a}^T + \tilde{b} \tilde{b}^T \] (3.59)

If this problem can be solved, we can perform the composition of Eq. (3.58) by repeated application of the method.
Equation (3.59) can be written as

\[
\begin{bmatrix}
1 \\
\ddots \\
a_n
\end{bmatrix}
\begin{bmatrix}
a_2 \\
a_3 \\
\vdots \\
a_n
\end{bmatrix}
+ \beta
\begin{bmatrix}
b_1 \\
b_2 \\
\vdots \\
b_n
\end{bmatrix}
= \alpha
\begin{bmatrix}
1 \\
\ddots \\
a_n
\end{bmatrix}
\begin{bmatrix}
\bar{a}_2 \\
\ddots \\
\bar{a}_n
\end{bmatrix}
+ \beta
\begin{bmatrix}
0 \\
\ddots \\
0
\end{bmatrix}
\begin{bmatrix}
\bar{b}_2 \\
\ddots \\
\bar{b}_n
\end{bmatrix}
\]  

(3.60)

Equating the \((1,1)\) elements gives

\[
\alpha + \beta b_1^2 = \bar{\alpha}
\]  

(3.61)

Equating the \((1,k)\) elements for \(k > 1\) gives

\[
\alpha a_k + \beta b_1 b_k = \bar{\alpha} \bar{a}_k
\]  

(3.62)

Adding and subtracting \(\beta b_1^2 a_k\) gives

\[
(\alpha + \beta b_1^2) a_k + \beta b_1 b_k - \beta b_1^2 a_k = \bar{\alpha} \bar{a}_k
\]

Hence

\[
\bar{a}_k = a_k + \frac{\beta b_1}{\bar{\alpha}} (b_k - b_1 a_k)
\]  

(3.63)

The numbers \(\bar{\alpha}\) and \(\bar{a}_k\) can thus be determined. It now remains to compute \(\bar{\beta}\) and \(\bar{b}_k\). Equating the \((k,l)\) elements of Eq. (3.60) for \(k, l > 1\) gives

\[
\alpha a_k a_l + \beta b_k b_l = \bar{\alpha} \bar{a}_k \bar{a}_l + \bar{\beta} \bar{b}_k \bar{b}_l
\]

\[
= \frac{(\alpha a_k + \beta b_1 b_k)(\alpha a_l + \beta b_1 b_l)}{\bar{\alpha}} + \bar{\beta} \bar{b}_k \bar{b}_l
\]

where Eq. (3.62) has been used to eliminate \(\bar{a}_k \bar{a}_l\). Inserting the expression in Eq. (3.61) for \(\bar{\alpha}\) gives, after some calculations,

\[
(b_k - b_1 a_k) (b_l - b_1 a_l) = \frac{\bar{\alpha} \bar{\beta}}{\alpha \beta} \bar{b}_k \bar{b}_l
\]
PROCEDURE
DyadicReduction(VAR a,b:col; VAR alpha,beta:REAL;
i0,i1,i2 :CARDINAL);
CONST
  mzero = 1.0E-10;
VAR
  i   : CARDINAL;
  w1,w2,b1,gam : REAL;
BEGIN
  IF beta{mzero} THEN beta:=0.0; END;
b1 := b[i0];
w1 := alpha;
w2 := beta*b1;
alpha := alpha + w2*b1;
IF alpha } mzero THEN
  beta := w1*beta/alpha;
gam := w2/alpha;
FOR i:=i1 TO i2 DO
  b[i] := b[i] - b1*a[i];
a[i] := a[i] + gam*b[i];
END;
END; END DyadicReduction;

Listing 3.2  Dyadic decomposition.

There are several solutions to these equations. A simple one is

\[ \tilde{b}_k = b_k - b_1 a_k \]
\[ \tilde{\beta} = \frac{\alpha \beta}{\tilde{\alpha}} \]

A solution to the dyadic decomposition problem of Eq. (3.59) is given by the equations

\[ \tilde{\alpha} = \alpha + \beta b_1^2 \]
\[ \tilde{\beta} = \frac{\alpha \beta}{\tilde{\alpha}} \]
\[ \beta = \frac{\beta b_1}{\tilde{\alpha}} \]
\[ \tilde{b}_k = b_k - b_1 a_k \quad k = 2, \ldots, n \]
\[ \tilde{a}_k = a_k + \gamma \tilde{b}_k \quad k = 2, \ldots, n \]
PROCEDURE
LDFilter(VAR theta,d:col; VAR l:matr; phi:col;
        lambda:REAL; n:CARDINAL);
VAR
    i,j : CARDINAL;
    e,w : REAL;
BEGIN
    d[0] := lambda;
    e := phi[0];
    FOR i:=1 TO n DO
        e:=e-theta[i]*phi[i];
        w:=phi[i];
        FOR j:=i+1 TO n DO w:=w+phi[j]*l[i,j]; END;
        l[0,i] := 0.0;
        l[i,0] := w;
        END;
    FOR i:=n TO 1 BY -1 DO (* Notice backward loop *)
        DyadicReduction(l[0],l[i],d[0],d[i],0,i,n);
        END;
    FOR i:=1 TO n DO
        theta[i] := theta[i]+l[0,i]*e;
        d[i] := d[i]/lambda;
        END;
END LDFilter;

Listing 3.3 LD decomposition.

Computer Code
The algorithm in Listing 3.2 is an implementation of the dyadic decom-
position. In this code the type

\[
\text{col} = \text{ARRAY}[0..\text{maxindex}] \text{ OF REAL};
\]

has been introduced. Using the procedure DyadicReduction it is now straightforward to write a procedure that implements Algorithm 3.2. Such a procedure is given in Listing 3.3. The algorithm performs one step of a recursive least-squares estimation. Starting from the current estimate \( \theta \), the covariance represented by its LD decomposition, and the regression vector, the procedure generates updated values of the estimate and its covariance. The data type

\[
\text{matr} = \text{ARRAY}[0..\text{maxindex}] \text{ OF col};
\]

is used in the program.
3.7 Conclusions

In this chapter we have summarized some methods for recursive parameter estimation. The focus has been on least squares and methods closely related to least squares. The idea has been to develop a few schemes that are useful for recursive estimation of parameters in dynamical systems. There are other algorithms and many fine points on recursive estimation; a full treatment would require a course by itself.

Problems

3.1 Consider the function

\[ V(x) = x^T A x + b^T x + c \]

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

\[ \nabla_x V = (A + A^T)x + b. \]

3.2 Consider the model

\[ y(t) = a + b \cdot t + \epsilon(t) \quad t = 1, 2, 3, \ldots \]

where \( \{\epsilon(t)\} \) 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.

3.3 Consider the model in Problem 3.2 but assume continuous-time observation where \( \epsilon(t) \) is white noise, i.e., a random function with covariance \( \delta(t) \). Determine the estimate and its covariance. Analyze the behavior of the covariance for large observation intervals.

3.4 Consider the following model of time-varying parameters

\[ \theta(t) = \Phi \theta(t - 1) + v(t) \]

\[ y(t) = \varphi^T(t) \theta(t) + \epsilon(t) \]

where \( \{v(t), t = 1, 2, \ldots\} \) and \( \{\epsilon(t), t = 1, 2, \ldots\} \) 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 $\theta$ are given by

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

$$K(t) = \Phi P(t-1) \varphi(t-1) \left( R_2 + \varphi^T(t-1)P(t-1)\varphi(t-1) \right)^{-1}$$

$$P(t) = \Phi P(t-1)\Phi^T + R_1 - \Phi P(t-1)\varphi(t) \left( R_2 + \varphi^T(t)P(t-1)\varphi(t) \right)^{-1} \varphi^T(t)P(t-1)\Phi^T$$

### 3.5 Consider the FIR model

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

where $\{e(t)\}$ is a sequence of independent normal $N(0, \sigma)$ random variables. Determine the least-squares estimate of the parameters $b_0$ and $b_1$ when the input signal $u$ is (a) a step and (b) white noise with unit variance. Analyze the covariance of the estimate when the number of observations goes to infinity. Relate the results to the notion of persistent excitation.

### 3.6 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. Determine the estimates obtained for large observation sets when the input $u$ is (a) a step and (b) a sequence of independent $N(0, \sigma)$ random variables. (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.)

### 3.7 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 $\sigma$ and 1. Determine the covariance of the estimates obtained for large observation sets.
3.8 Consider data generated by the least-squares model
\[ y(t + 1) + ay(t) = bu(t) + e(t + 1) + ce(t) \quad t = 1, 2, \ldots \]
where \{u(t)\} and \{e(t)\} are sequences of independent random variables with zero mean values and standard deviations 1 and \( \sigma \). 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 when the observation sets \( T_1 = \{1, 2, \ldots, t\} \) and \( T_2 = \{2, 4, \ldots, 2t\} \) are used.

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

3.10 Use the program from Problem 3.9 to estimate the parameters \( a \) and \( b \) in Problem 3.8. Investigate how the bias of the estimate depends on \( c \).

3.11 Consider the estimation problem in Problem 3.8. Use the computer program developed in Problem 3.9 to explore what happens when the control signal \( u \) is generated by the feedback
\[ u = -ky \]
Try to support your observations by analysis.

3.12 Consider recursive least squares. Prove a result analogous to Theorem 3.9, or to be specific, that
(i) \[ \| \hat{\theta}(t) - \theta^0 \| \leq C\| \hat{\theta}(0) - \theta^0 \| \quad C = \lambda_{\text{max}}(P(0))/\lambda_{\text{min}}(P(0)) \]
(ii) \[ \lim_{t \to \infty} \frac{e(t)}{\sqrt{1 + \varphi^T(t-1)P(t-1)\varphi(t-1)}} = 0 \]
(iii) \[ \lim_{t \to \infty} \| \hat{\theta}(t + k) - \hat{\theta}(t) \| = 0 \text{ for any finite } k \]
Why is it advantageous to choose \( P(0) \) as \( \alpha I \)?

3.13 Consider recursive least-squares estimation with covariance resetting. This means that the matrix \( P \) is reset as
\[ P(t) = \alpha I \quad t = n \cdot T \]
where \( T \) is larger than the number of parameters. Formulate and prove a result analogous to Theorem 3.9.
SUBROUTINE EST(Y,PHI,D,U,K,THETA,NPAR,E,LAMBDA)
REAL K
DIMENSION PHI(1), D(1), U(1), K(1), THETA(1)
C U/D FILTER, AFTER BIERMAN AND THORNTON
C Y TRUE OUTPUT VALUE
C PHI REGRESSION, MOST RECENT IN PHI(1)
C U,D U = UPPER TRIANGL. D = DIAG
C K KALMAN GAIN VECTOR
C THETA PARAMETER ESTIMATES VECTOR
C NPAR NUMBER OF PARAMETERS - LENGTH OF PHI,D,K,THETA
C E PREDICTION ERROR: Y - PHI (TRANSPOSED) THETA
C LAMBDA FORGETTING FACTOR
E = Y
DO 1 I = 1, NPAR
1 E = E - THETA(I) * PHI(I)
FJ = PHI(I)
VJ = D(I) * FJ
K(I) = VJ
ALPHAJ = 1.0 + VJ * FJ
D(I) = D(I)/ALPHAJ
IF (NPAR. EQ. 1) GO TO 100
KF = 0
KU = 0
DO 4 J = 2, NPAR
FJ = PHI(J)
DO 2 I = 1, J-1
KF = KF+1
2 FJ = FJ + PHI(I) * U(KF)
VJ = FJ * D(J)
K(J) = VJ
AJLAST = ALPHAJ
ALPHAJ = AJLAST + VJ * FJ
D(J) = D(J) * AJLAST/ALPHAJ/LAMBDA
PJ = -FJ/AJLAST
DO 3 I = 1, J-1
KU = KU+1
W = U(KU) + K(I) * PJ
3 K(I) = K(I) + U(KU) * VJ
KU = W
4 CONTINUE
100 CONTINUE
DO 5 I = 1, NPAR
5 THETA(I) = THETA(I) + E * K(I)/ALPHAJ
RETURN
END

Listing 3.4 Fortran listing of a parameter estimator.
3.14 Consider data generated by

\[ y(t) = b + e(t) \quad t = 1, 2, \ldots, N \]

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

\[ e(2) = a \]

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.)

3.15 Consider least-squares estimation at 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.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1001</td>
<td>2001</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>2001</td>
</tr>
</tbody>
</table>

Compare numerically the solution obtained by solving the normal equations with the straightforward solution.

3.16 Listing 3.4 gives a Fortran program for a recursive estimator. Explain the algorithm behind the program.

References

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


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


Recursive identification is treated in depth in:


Good sources are also the proceedings from the IFAC symposia on system identification that have been held triannually since 1967. The numerical solution to least-squares problems is well treated in:


Recursive square root algorithms are discussed in:


The dyadic decomposition method described in Section 3.6 is based on:


Chapter 4

MODEL-REFERENCE
ADAPTIVE SYSTEMS

4.1 Introduction

The model-reference adaptive system (MRAS) is one of the main approaches to adaptive control. The basic principle is illustrated in Fig. 4.1. The desired performance is expressed in terms of a reference model, which gives the desired response to a command signal. The system also has an ordinary feedback loop composed of the process and the regulator. The error $e$ is the difference between the outputs of the system and the reference model. The regulator has parameters that are changed based on the error. There are thus two loops in Fig. 4.1: an inner loop, which provides the ordinary control feedback, and an outer loop, which adjusts the parameters in the inner loop. The inner loop is assumed to be faster than the outer loop.

Figure 4.1 is the original MRAS, proposed by Whitaker in 1958, in
which two new ideas were introduced. First, the performance of a system is specified by a model; secondly, the parameters of the regulator are adjusted based on the error between the reference model and the system. Model-reference adaptive systems were originally derived for the servo problem in deterministic continuous-time systems. The idea and the theory have been extended to cover discrete-time systems and systems with stochastic disturbances. This chapter focuses on the basic ideas. In order to make the presentation clear we concentrate on the configuration in Fig. 4.1. This is called a parallel MRAS. It is one of many possible ways of making a model-reference system. This chapter deals mainly with continuous-time direct MRAS, in which the regulator parameters are updated directly (compare Section 1.2).

There are essentially three basic approaches to the analysis and design of a MRAS

- The gradient approach
- Lyapunov functions
- Passivity theory.

The gradient method was used by Whitaker in the original work on the MRAS. This approach is based on the assumption that the parameters change more slowly than the other variables in the system. This assumption, which admits a quasi-stationary treatment, is essential for the computation of the sensitivity derivatives that are needed in the adaptation mechanism. The gradient approach will not necessarily result in a stable closed-loop system. This observation inspired the application of stability theory. Lyapunov’s stability theory and passivity theory have been used to modify the adaptation mechanism.
Model-following is an important part of the MRAS, as in other adaptive controllers. A thorough discussion of model-following and regulator design is given in Appendix A. Section 4.2 contains a brief discussion of the model-reference control problem. The basic principle of the MRAS, based on the gradient approach, is given in Section 4.3. The stability approaches to the MRAS are analyzed in Section 4.4. By following the historic development of the MRAS we can easily bring out some of the important difficulties of adaptive control. Stability of the closed-loop system is discussed, together with convergence of the controller parameters. In the first sections of the chapter we introduce the different ideas of the MRAS using Example 1.1, in which only a feedforward gain must be adjusted. This system is not difficult to control, but it clearly illustrates the difficulties in adaptive control.

Section 4.5 gives a derivation of an MRAS for a general linear system, pointing out the relation between parameter estimation and the MRAS. This also serves as a natural connection to the discussion of the self-tuning regulator in the next chapter. Section 4.6 shows how prior knowledge can be incorporated into a MRAS, using the example of a robot. The chapter is summarized in Section 4.7.

4.2 The MRAS Problem

For a system with adjustable parameters (as in Fig. 4.1) the model-reference adaptive method gives a general approach for adjusting the parameters so that the closed-loop transfer function will be close to a prescribed model. This is called the model-following problem. One important question is how small we can make the error $e$. This depends both 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.

Model-following

The model-following problem can be solved using pole placement design. (An overview of pole placement design is given in Appendix A.) Model-following is a simple and useful way to formulate and solve a servo control problem. The basic idea is very simple. Servo performance is specified indirectly by giving a mathematical model for the desired response. The specified model can be linear as well as nonlinear. The parameters in the system are adjusted in order to get $y$ as close as possible to $y_m$ for a given class of input signals. Optimization methods are thus natural tools in MRAS design. This approach is discussed in Section 4.3. Although perfect model-following can only be obtained in idealized situations; an
analysis of this case gives good insight into the design problem. It also provides a way of unifying the treatment of the MRAS and the self-tuning regulator.

Consider a single-input, single-output system, which may be either a continuous-time model or a discrete-time model:

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

(4.1)

where \(u\) is the control signal and \(y\) the output signal. The symbols \(A\) and \(B\) denote polynomials in either the differential operator \(p\) or the forward shift operator \(q\). It is assumed that \(A\) and \(B\) are relatively prime. Furthermore, it is assumed that \(\text{deg} A \geq \text{deg} B\), i.e., that the system is proper (the continuous-time case) or causal (the discrete-time case). The polynomial \(A\) is assumed to be monic, i.e., the first coefficient is unity.

Assum that we want to find a regulator such that the relation between the command signal \(u_c\) and the desired output signal \(y_m\) is given by

$$y_m(t) = \frac{B_m}{A_m} u_c(t)$$

(4.2)

where \(A_m\) and \(B_m\) are polynomials in the differential operator or the forward shift operator. A general linear control law can be described as

$$Ru = Tu_c - Sy$$

(4.3)

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 \(T/R\). See Fig. 4.2. Elimination of \(u\) between Eqs. (4.1) and (4.3) gives the following equation for the closed-loop system:

$$(AR + BS)y = BTu_c$$

(4.4)

To obtain the desired closed-loop response, \(A_m\) must divide \(AR + BS\). The process zeros, given by \(B = 0\), will also be closed-loop zeros unless
they are canceled by corresponding closed-loop poles. Since unstable or poorly damped zeros cannot be canceled, the polynomial $B$ is factored as

$$B = B^+ B^-$$  \hspace{1cm} (4.5)

where $B^+$ contains those factors that can be canceled, and $B^-$ contains the remaining factors of $B$. The zeros of $B^+$ must be stable and well damped. To make the factorization unique, $B^+$ is assumed to be monic.

It follows from Eq. (4.4) that the characteristic polynomial of the closed-loop system is $AR + BS$. This polynomial must have $A_m B^+$ as a factor and will generally be of higher order than $A_m B^+$. The remaining factor can be interpreted as observer dynamics. There are thus three types of factors of the characteristic polynomial: canceled process zeros given by $B^+$, desired model poles given by $A_m$, and observer poles given by the observer polynomial $A_o$. Hence

$$AR + BS = B^+ A_o A_m$$ \hspace{1cm} (4.6)

which is called the Diophantine equation. (Some authors prefer to call it the Bezout identity.) It follows from this equation that $B^+$ divides $R$. Hence

$$R = B^+ R_1$$ \hspace{1cm} (4.7)

Dividing Eq. (4.6) by $B^+$ gives

$$AR_1 + B^- S = A_o A_m$$ \hspace{1cm} (4.8)

Now require that the relation in Eq. (4.4) between the command signal $u_c$ and the process output $y$ should be equal to the desired closed-loop response given by Eq. (4.2). The specifications must also be such that $B^-$ divides $B_m$; otherwise there is no solution to the design problem. Hence

$$B_m = B^- B'_m$$

$$T = A_o B'_m$$ \hspace{1cm} (4.9)

To complete the solution of the problem we must give conditions to guarantee that there exist solutions to Eq. (4.8) that give a proper (continuous-time) or causal (discrete-time) control law:

$$\deg A_o \geq 2 \deg A - \deg A_m - \deg B^+ - 1$$ \hspace{1cm} (4.10)

$$\deg A_m - \deg B_m \geq \deg A - \deg B$$ \hspace{1cm} (4.11)

A discussion of these conditions is given in Appendix A.
Model-following or Pole Placement

The control law of Eq. (4.3), with the controller polynomials given by Eqs. (4.7), (4.8), and (4.9), gives perfect model-following if the compatibility conditions of Eqs. (4.10) and (4.11) are fulfilled. Notice that the design above contains the solution of the Diophantine equation (Eq. 4.8) and is thus not suited for a direct adaptive controller. Assume, however, that all zeros are canceled (compare Appendix A). Then

\[ A_o A_m = AR_1 + b_0 S \]

Multiply this by \( y \) and use the model equation of Eq. (4.1). This gives

\[ A_o A_m y = BR_1 u + b_0 Sy \]
\[ = b_0 (Ru + Sy) \]  \hspace{1cm} (4.12)

The polynomials on the left-hand side are known, and those on the right-hand side are the unknown controller parameters. The \( T \) polynomial is obtained directly from Eq. (4.9). The reparameterized model of Eq. (4.12) can now be used to estimate the unknown regulator parameters using the methods presented in Chapter 3. This leads to a direct MRAS. The general solution will be discussed in Section 4.5.

4.3 The Gradient Approach

The gradient approach to model-reference adaptive control is developed in this section. This is a fundamental idea in the MRAS approach. The parameter adjustment scheme is usually called the MIT rule because the work was done at the Instrumentation Laboratory (now the Draper Laboratory) at MIT. A discussion of a particular example illustrates the idea and reveals a fundamental difficulty. Modified adjustment rules that avoid the difficulty are then given.

The MIT Rule

Assume that we attempt to change the parameters of the regulator so that the error between the outputs of process and the reference model is driven to zero. Let \( e \) denote the error and \( \theta \) the parameters. Introduce the criterion

\[ J(\theta) = \frac{1}{2} e^2 \]  \hspace{1cm} (4.13)

To make \( J \) small it is reasonable to change the parameters in the direction of the negative gradient of \( J \), i.e.,

\[ \frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta} \]  \hspace{1cm} (4.14)
If it is assumed that the parameters change much more slowly than the other variables in the system, then the derivative $\partial e/\partial \theta$ can be evaluated under the assumption that $\theta$ is constant. The derivative $\partial e/\partial \theta$ is the sensitivity derivative of the system. The adjustment rule of Eq. (4.14), where $\partial e/\partial \theta$ is the sensitivity derivative, is commonly referred to as the MIT rule. The choice of loss function in Eq. (4.13) is, of course, arbitrary. If the loss function is chosen as

$$J(\theta) = |e|$$

the adjustment rule becomes

$$\frac{d\theta}{dt} = -\gamma \frac{\partial e}{\partial \theta} \operatorname{sign}(e)$$

This is actually the way that the first MRAS was implemented. Use of the MIT rule for adaptation of a feedforward gain was illustrated in Examples 1.1 and 1.5. An even simpler implementation is obtained by using the update rule

$$\frac{d\theta}{dt} = -\gamma \operatorname{sign}\left(\frac{\partial e}{\partial \theta}\right) \operatorname{sign}(e)$$

This is called the sign-sign algorithm. A discrete-time version of this algorithm is used in telecommunications, where simple implementation and fast computations are required.

Equation (4.14) also applies to the case of many adjustable parameters. The variable $\theta$ should then be interpreted as a vector and $\partial e/\partial \theta$ as the gradient of the error with respect to the parameters. The application of the MIT rule is illustrated by two examples.

**Example 4.1—Adaptation of a feedforward gain**

Consider the problem of adjusting a feedforward gain, discussed in Examples 1.1 and 1.5. Let the model and the process have the transfer function $G(s)$. The error is

$$e = y - y_m = G(p)\theta u_c - G(p)\theta^0 u_c$$

where $u_c$ is the command signal, $y_m$ the model output, $y$ the process output, $\theta$ the adjustable parameter, and $p = d/dt$ the differentiation operator. The sensitivity derivative is

$$\frac{\partial e}{\partial \theta} = G(p)u_c = y_m/\theta^0$$

The MIT rule then gives

$$\frac{d\theta}{dt} = -\gamma' y_m e/\theta^0$$
If the sign of the parameter $\theta^0$ is known, the parameter $\gamma = \gamma'/\theta^0$ can be introduced. The rate of change of the parameter should thus be made proportional to the product of the error and the model output. □

Notice that no approximations were needed in Example 4.1. When the MIT rule is applied to more complicated problems, it is necessary to use approximations to obtain the sensitivity derivatives. This is illustrated by an additional example.

**Example 4.2—MRAS for a first-order system**

Consider a system described by the model

$$\frac{dy}{dt} = -ay + bu$$  \hspace{1cm} (4.17)

where $u$ is the control variable and $y$ the measured output. Assume that it is desirable to obtain a closed-loop system described by

$$\frac{dy_m}{dt} = -a_my_m + b_mu_c$$

Perfect model-following can be achieved with the controller

$$u(t) = t_0u_c(t) - s_0y(t)$$  \hspace{1cm} (4.18)

with the parameters

$$t_0 = \frac{b_m}{b}$$

$$s_0 = \frac{a_m - a}{b}$$

Notice that the feedback will be positive if $a_m < a$, i.e., if the desired model is slower than the process. To apply the MIT rule, introduce the error

$$e = y - y_m$$

where $y$ now denotes the closed-loop output. It follows from Eqs. (4.17) and (4.18) that

$$y = \frac{bt_0}{p + a + bs_0} u_c$$

where $p$ is the differential operator. The sensitivity derivatives are obtained by taking partial derivatives with respect to the regulator parameters $t_0$ and $s_0$

$$\frac{\partial e}{\partial t_0} = \frac{b}{p + a + bs_0} u_c$$

$$\frac{\partial e}{\partial s_0} = -\frac{b^2t_0}{(p + a + bs_0)^2} u_c = -\frac{b}{p + a + bs_0} y$$
These formulas cannot be used, because the process parameters \( a \) and \( b \) are not known. Approximations are therefore required in order to obtain realizable parameter adjustment rules. To derive these, first observe that with the optimal values of the regulator parameters we have

\[
p + a + bs_0 = p + a_m
\]

Furthermore, notice that the parameter \( b \) may be absorbed in the adaptation gain \( \gamma \), since it appears in the product \( \gamma b \). This requires, however, that the sign of \( b \) be known. After these approximations the following equations for updating the regulator parameters are obtained:

\[
\begin{align*}
\frac{dt_0}{dt} &= -\gamma \left( \frac{1}{p + a_m} u_c \right) e \\
\frac{ds_0}{dt} &= \gamma \left( \frac{1}{p + a_m} y \right) e
\end{align*}
\] (4.19)

Figure 4.3 shows a simulation of the MRAS with \( a = 1 \), \( b = 0.5 \), \( a_m = 2 \), and \( b_m = 2 \). The input signal is a square wave with amplitude 1, and \( \gamma = 2 \). The closed-loop system is close to the desired behavior after only a few transients. The convergence rate depends critically on the parameters \( \gamma \) and \( b \). The code for the MRAS is given in Listing 4.1. This indicates the simplicity of the control law.

The example shows how the MIT rule can be used to derive a parameter adjustment law. Certain characteristics are worth noticing.

- It is not necessary to require perfect model-following. The procedure can be applied to nonlinear systems. The method can also be used to handle partially known systems.
- The structure of Fig. 1.6 appears again. There is a multiplication of \( e \) and \( \partial e / \partial \theta \). The integration of Eq. (4.19) gives the parameters, which are transferred to the regulator using a second multiplier.
- Approximations are necessary in order to obtain a realizable parameter adjustment control law.

The MIT rule will perform well if the adaptation gain \( \gamma \) is small. The allowable size depends, however, on the magnitude of the reference signal and the process gain. Consequently it is not possible to give fixed limits that guarantee stability. The MIT rule can thus give an unstable closed-loop system. Modified adjustment rules can be obtained using stability theory. These rules are similar to the MIT rule. The sensitivity derivatives are, however, replaced by other functions. This is discussed further in Section 4.4.
CONTINUOUS SYSTEM mras

"MRAS for first-order system and two regulator parameters t0 and s0
"The model has the transfer function
" \( G_m(s) = \frac{bm}{s + am} \)
"
"Variables: \( y_m \): state of the model
" \( t0, s0 \): regulator parameters
" \( xt, xs \): sensitivity derivatives

INPUT y uc "input variables
STATE ym t0 s0 xt xs "states of the MRAS
DER dym dt0 ds0 dxs "derivatives of the states

dym=-am*ym+bm*uc
dxt=-am*xt+uc
dx=-am*xs+y
e=y-ym
den=alfa*xt*xt+xs*xs
dt0=-gamma*e*xt/den
ds0=gamma*e*xs/den

am:2 "model parameter
bm:2 "model parameter
gamma:2 "mras gain
alfa:0.0001 "parameter in normalization denominator

END

Listing 4.1 A Simmon program for a first-order MRAS system.

General Linear Systems
A model-reference control law based on the gradient approach will now be derived for a general single-input, single-output system. Let the system be described by the model of Eq. (4.1):

\[
Ay = Bu
\]

Assume that it is desired to obtain a system characterized by

\[
A_m y_m = B_m u_c
\]
Figure 4.3 Process output $y$, model output $y_m$, and control signal $u$ when simulating the system in Example 4.2 using an MRAS.

It was shown in Section 4.2 that an appropriate regulator structure is Eq. (4.3):

$$Ru = T u_c - S y$$

The closed-loop system is described by

$$y = \frac{BT}{AR + BS} u_c$$

and

$$u = \frac{AT}{AR + BS} u_c$$

The error is

$$e = y - y_m$$

To obtain the parameter adjustment law the sensitivity derivatives (i.e., the partial derivatives of the error with respect to the regulator parameters) have to be determined. Let $r_i$, $s_i$, and $t_i$ be the coefficients of the
polynomials $R$, $S$, and $T$. The sensitivity derivatives are then given by

$$\frac{\partial e}{\partial r_i} = -\frac{BTAp^{k-i}}{(AR + BS)^2} u_c = -\frac{Bp^{k-i}}{AR + BS} u \quad i = 1, \ldots, k$$

$$\frac{\partial e}{\partial s_i} = -\frac{BTBp^{l-i}}{(AR + BS)^2} u_c = -\frac{Bp^{l-i}}{AR + BS} y \quad i = 0, \ldots, l$$

$$\frac{\partial e}{\partial t_i} = \frac{Bp^{m-i}}{AR + BS} u_c \quad i = 0, \ldots, m$$

where $k = \deg R$, $l = \deg S$, and $m = \deg T$. The second equalities are obtained by using the equations for $y$ and $u$ given above. The sensitivity derivatives cannot be computed, because the right-hand side includes the polynomials $A$ and $B$, containing the unknown process parameters. There are several ways to make approximations that give realizable updating laws. One possibility is

$$AR + BS \approx A_oA_mB^+$$

This approximation will be exact when the parameters have their desired values. The sensitivity derivative is then approximated by

$$\frac{\partial e}{\partial r_i} \approx -\frac{Bp^{k-i}}{A_oA_m} u$$

The approximations are analogous for $s_i$ and $t_i$.

The right-hand side is still not realizable because it contains $B^-$. However, if all process zeros are canceled, we get $B^- = b_0$. A realizable updating law can now be obtained if the sign of $b_0$ is known. The magnitude of $b_0$ can be absorbed in the adaptation gain. The following equations are then obtained for the parameter adjustments:

$$\frac{dr_i}{dt} = \gamma e \frac{p^{k-i}}{A_oA_m} u \quad i = 1, \ldots, k = \deg R$$

$$\frac{ds_i}{dt} = \gamma e \frac{p^{l-i}}{A_oA_m} y \quad i = 0, \ldots, l = \deg S$$

$$\frac{dt_i}{dt} = -\gamma e \frac{p^{m-i}}{A_oA_m} u_c \quad i = 0, \ldots, m = \deg T$$

To implement this parameter adjustment law, three state space representations of the filter $1/A_oA_m$ have to be constructed. These systems are
driven by $u$, $y$, and $u_c$. The rate of change of each parameter is then obtained by multiplying the error $e$ by signals obtained by tapping the filters. To derive the parameter adjustment it would be necessary to assume that the process zeros are stable and that the sign of the instantaneous gain $b_0$ is known.

This assumption can be avoided by a more complex algorithm. If a parameter estimator is added, the following approximations can be made:

$$AR + BS \approx \hat{AR} + \hat{BS}$$

$$B^- \approx \hat{B}^-$$

It is then possible to obtain an algorithm that works for nonminimum phase systems and for systems where the sign of $b_0$ is not known. Notice, however, that the algorithm is then indirect, because process parameters are estimated.

**Explicit Criterion Minimization**

The MIT rule can be extended to optimization of more general loss functions than Eq. (4.13). The approach is straightforward in principle. The adaptive controller is obtained in the following way. A model and a regulator with adjustable parameters are specified, and the performance is given as a loss function. The parameter adjustment law is obtained by computing the gradient of the loss function with respect to the parameters and making the rate of change of the parameters in the direction opposite to the gradient. This approach can be applied to a wide variety of control problems; the details may be complicated. The method has the same problems as the MIT rule. It is necessary to know the model parameters in order to compute the sensitivity derivatives. Since this is not realistic, some approximations have to be made. One possibility is to replace the process parameters by their estimates. (Further references are given in the end of this chapter.)

**Error and Parameter Convergence**

Model-reference adaptive systems are based on the idea of driving the error $e = y - y_m$ to zero. This does not necessarily imply that the regulator parameters will approach their correct values. The case where the input signal is zero is a simple counterexample. Another example is the following.

**Example 4.3—Error convergence**

Consider the updating of a feedforward gain, discussed in Examples 1.1 and 1.5. We have the process $y = u$, the control law $u = \theta u_c$, and the model $y_m = \theta^0 u_c$. The error is

$$e = (\theta - \theta^0)u_c$$
The gradient method gives the following equation for the parameter:

\[
\frac{d\theta}{dt} = -\gamma u_c^2(\theta - \theta^0)
\]

This differential equation has the solution

\[
\theta(t) = \theta^0 + (\theta(0) - \theta^0)e^{-\gamma I_t}
\]

(4.20)

where

\[
I_t = \int_0^t u_c^2(\tau) \, d\tau
\]

and \(\theta(0)\) is the initial value of the parameter \(\theta\), and the error is given by

\[
e(t) = u_c(t)(\theta(0) - \theta^0)e^{-\gamma I_t}
\]

The error will always go to zero as time increases. Either \(u_c(t) \to 0\), or, if this is not the case, then the integral of \(u_c^2(t)\) will diverge.

The limiting value of the parameter \(\theta\) will, however, depend on the properties of the input signal.

Example 4.3 illustrates the property that the error \(e\) goes to zero but that the parameters do not necessarily go to their correct values. This is a characteristic feature of model-reference adaptive systems. The input signal must be sufficiently irregular for the parameters to converge. The precise condition is that the command signal must be persistently exciting. (Compare Section 3.4.)

**Stability of the Adaptive Loop**

It follows from Example 4.3 that the convergence rate of the parameters is proportional to the square of the magnitude of the command signal. This may be reasonable in some cases, because a larger command signal makes it easier to detect that \(\theta\) has the wrong value. However, if the measurement errors are negligible, it may be of interest to have adjustment rules where the parameter adjustment rate does not depend on the magnitude of the input signal. The fact that the parameter adjustment rate depends on the magnitude of the command signal can also lead to instability. This is illustrated by an example.

**Example 4.4—Stability of the adaptive loop**

Consider the simple MRAS shown in Fig. 4.4, in which the problem is to adjust a feedforward gain \(\theta\) to the value \(\theta^0\). Let the transfer function \(G\) be given by

\[
G(s) = \frac{1}{s^2 + a_1 s + a_2}
\]
The error is given by
\[ e = G(p)(\theta - \theta^0)u_c \]
where \( p \) denotes the derivation operator. Hence
\[ \frac{\partial e}{\partial \theta} = G(p)u_c = y_m/\theta^0 \]
The MIT rule gives the following formula for adjusting the parameter:
\[ \frac{d\theta}{dt} = -\gamma' e \frac{\partial e}{\partial \theta} = -\gamma' e y_m/\theta^0 = -\gamma e y_m \quad (4.21) \]
where \( \gamma = \gamma'/\theta^0 \). The adaptive system can thus be represented by the differential equations
\[
\begin{align*}
\frac{d^2 y_m}{dt^2} + a_1 \frac{dy_m}{dt} + a_2 y_m &= \theta^0 u_c \\
\frac{d^2 y}{dt^2} + a_1 \frac{dy}{dt} + a_2 y &= \theta u_c \\
\frac{d\theta}{dt} &= -\gamma(y - y_m)y_m 
\end{align*}
\]
The differential equation for \( y_m \) can be solved if the command signal is a given time function. The variable \( y_m \) can then be regarded as a known function of time. Using the differential equations for \( y \) and \( \theta \) gives
\[
\frac{d^3 y}{dt^3} + a_1 \frac{d^2 y}{dt^2} + a_2 \frac{dy}{dt} + \gamma u_c(t)y_m(t)y(t) = \theta(t) \frac{du_c}{dt} + \gamma u_c(t)y_m^2(t)
\]
This is a time-varying linear differential equation. To get some insight into the behavior of the system, an experiment will be performed. First, assume that the adaptation mechanism is disconnected. Let the command input $u_c$ be a constant $u_c^0$. The model output $y_m$ will then go towards an equilibrium $y_m^0$. Assume that the adaptation mechanism is connected when the equilibrium is reached. In this special case the above equations have constant coefficients. This equation has the equilibrium solution

$$y(t) = y_m^0 = \theta^0 u_c^0 / a_2$$

which is stable if

$$a_1 a_2 > \gamma u_c^0 y_m^0 = \frac{\gamma'}{a_2} (u_c^0)^2$$

and unstable otherwise. It thus follows that the solution $y = y_m^0$ will always be unstable if the command signal $u_c$ or the adaptive gain $\gamma'$ is sufficiently large. Since the bound involves the magnitude of the command signal, it may well 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. 4.5. The figure also shows that the convergence rate depends
4.3 The Gradient Approach

on the magnitude of the command signal. In the simulation the adaptation rate $\gamma$ was adjusted to give a good response when $u_c$ is a square wave with unit amplitude. Notice that the solution is unstable when the amplitude of $u_c$ is 3.5. The response is untolerably slow for low amplitudes of $u_c$. □

An Important Remark

When analyzing the MRAS with time-varying parameters, it is important to consider a property of time-varying systems. The expression

$$G(p)(\theta - \theta^0)u_c$$

should be interpreted as the differential operator $G(p)$ acting on the signal $(\theta - \theta^0)u_c$. When $\theta$ is time-varying, this will be different from

$$(\theta - \theta^0)G(p)u_c$$

For instance,

$$p(\theta - \theta^0)u_c = \frac{d\theta}{dt} u_c + (\theta - \theta^0) \frac{du_c}{dt}$$

while

$$(\theta - \theta^0)pu_c = (\theta - \theta^0) \frac{du_c}{dt}$$

Care must be taken when manipulating block diagrams and expressions.

Modified Adjustment Rules

The insight obtained from the calculations in Example 4.3 indicates that it may be useful to modify the MIT rule. The MIT rule is basically a gradient procedure. The rate of decrease obtained by the MIT rule is determined by the parameter $\gamma$, whose choice is left to the user.

It is possible to obtain modified gradient procedures, in which the adjustment rate does not depend on the magnitude of the command signal. One possibility is to make a normalization and replace the MIT rule with

$$\frac{d\theta}{dt} = -\gamma \frac{e \frac{\partial e}{\partial \theta}}{\alpha + (\frac{\partial e}{\partial \theta})^T (\frac{\partial e}{\partial \theta})}$$

The parameter $\alpha > 0$ has been introduced in order to avoid a possible division by zero. Notice that it may make sense to have the parameter adjustment rate depend on the magnitude of the command signal for small
levels, because of measurement noise. As an extra precaution we can also introduce a saturation to guarantee that the parameter adjustment rate is always below a given limit. The following adjustment rule is then obtained:

$$\frac{d\theta}{dt} = -\gamma \text{ sat} \left( \frac{e \frac{\partial e}{\partial \theta}}{\alpha + \left( \frac{\partial e}{\partial \theta} \right)^T \left( \frac{\partial e}{\partial \theta} \right)}, \beta \right)$$

(4.22)

where

$$\text{sat}(x, \beta) = \begin{cases} 
-\beta & x < -\beta \\
               x & |x| \leq \beta \\
               \beta & x > \beta 
\end{cases}$$

The algorithm of Eq. (4.22) is called **normalized**. The usefulness of the normalized adjustment law is illustrated by an additional example.

**Example 4.5—Modified adjustment rule**

Consider the same system as in Example 4.4, but use the modified adjustment rule of Eq. (4.22) instead of the MIT rule (Eq. 4.21). Simple calculations show that Eq. (4.22) is replaced by

$$\frac{d\theta}{dt} = -\gamma \text{ sat} \left( \frac{(G(p)(\theta - \theta^0)u_c)G(p)u_c}{\alpha + (G(p)u_c)^2}, \beta \right) = -\gamma \text{ sat} \left( \frac{e y_m/\theta^0}{\alpha + y_m^2/\theta^0_0}, \beta \right)$$

Figure 4.6 shows the same cases as in Fig. 4.5. Notice that the modified adjustment rule performs very well even in the cases where difficulties were encountered with the MIT rule. It is in fact possible to make the modified adjustment rule work very well over a large range of command signal amplitudes. However, unstable systems may still be obtained, for instance by making $\gamma$ sufficiently large or by using a “bad” $y_m(t)$. □

A comparison between Examples 4.4 and 4.5 shows convincingly that it is useful to introduce nonlinear modifications of the parameter adjustment rule. The usefulness of similar nonlinearities will be frequently encountered later in the book.

**Summary**

Let the reference model be a linear time-invariant system characterized by the transfer function $G_m$. Assume that the process (together with the regulator for fixed values of the parameters) is a linear time-invariant system characterized by the transfer operator $G(p, \theta)$. The modified MIT
4.3 The Gradient Approach

Figure 4.6 Simulation of the MRAS in Example 4.5 with the modified MIT rule. The parameters used are $a_1 = a_2 = \theta^0 = 1, \alpha = 0.01, \beta = 2$, and $\gamma = 0.1$. The command signal is a square wave with the amplitude (a) 0.1, (b) 1, and (c) 3.5. Compare with Fig. 4.5.

The rule gives

$$e = (G(p, \theta) - G_m(p))u_c$$

$$\frac{\partial e}{\partial \theta} = G_\theta(p, \theta)u_c$$

$$\frac{d\theta}{dt} = -\gamma \text{sat} \left( \frac{(G(p, \theta) - G_m(p))u_c G_\theta(p, \theta)u_c}{\alpha + (G_\theta(p)u_c)^T G_\theta(p)u_c} , \beta \right)$$

The equilibrium values of the parameters are those that correspond to local minima of the function

$$\left\| (G(p, \theta) - G_m(p))u_c(t) \right\|^2$$

How many such local minima exist depends on the command signal $u_c$, the structure, and the parameterization of the model. For example, if the model $G(p, \theta)$ is linear in $\theta$, there will be only one minimum.

Gradient procedures provide a simple way to design parameter adjustment rules in model-reference adaptive systems. This approach can be
used for control systems with different structures. The crucial calculation is the computation of the sensitivity derivatives and the associated approximations. The gradient algorithm will work if the adaptation gain is chosen sufficiently small and if the initial values of the parameters correspond to a stable closed-loop system. The MIT rule is a simple example of a gradient algorithm. A characteristic feature is that the convergence rate depends on the magnitude of the reference signal. If this behavior is not desired, the rule can be modified to give adjustment rates that are virtually independent of the magnitude of the command signal. It has also been shown that the gradient approach will not necessarily give a stable closed-loop system.

4.4 MRAS Based on Stability Theory

When parameter adjustment rules based on the gradient method were discussed in Section 4.3, the approach taken was to derive an adjustment rule that appeared reasonable heuristically. We then attempted to show that the model error would in fact go to zero. Another possibility to obtain the outer loop of a model-reference adaptive system is to try to find adjustment laws such that the error is guaranteed to go to zero. The search for such techniques has stimulated research over a long period of time. The fundamental ideas behind the design of adjustment laws based on stability theory are presented in this section. The presentation follows the historical development.

To focus on the principles and avoid unnecessary details, the problem of adjusting the feedforward gain of an otherwise known system is used throughout this section. It is the same system and model as in Fig. 4.4, but the adaptation mechanism will now be different. The problem is to find a feedback law such that the error \( e = y - y_m \) in Fig. 4.4 is guaranteed to go to zero. Notice that the problem of controlling a system with known dynamics and unknown gain is not particularly difficult. The particular problem has been chosen to illustrate ideas rather than to represent a realistic problem. Once the basic ideas are developed, the extension to more general configurations is comparatively straightforward. The details of this are given in Section 4.5.

Lyapunov’s Second Method

Lyapunov introduced an interesting direct method to investigate the stability of a solution to a nonlinear differential equation. The key idea is illustrated in Fig. 4.7. The equilibrium will be stable if we can find a real function on the state space whose level curves enclose the equilibrium such
that the derivative of the state variables always points towards the interior of the level curves.

To state the results formally, let the differential equation be

\[ \dot{x} = f(x, t) \quad f(0, t) = 0 \]  

(4.23)

where \( x \) is a state vector of dimension \( n \). It is assumed that \( f \) is such that solutions exist for all \( t \geq t_0 \). The equilibrium point is assumed to be at the origin. This involves no loss of generality, since this can be achieved through a simple coordinate transformation.

**Theorem 4.1—Lyapunov stability theorem**

Let the function \( V : \mathbb{R}^{n+1} \to \mathbb{R} \) satisfy the conditions

1. \( V(0, t) = 0 \) for all \( t \in \mathbb{R} \).
2. \( V \) is differentiable in \( x \) and \( t \).
3. \( V \) is positive definite, i.e., \( V(x, t) \geq g(\|x\|) > 0 \) where \( g : \mathbb{R} \to \mathbb{R} \) is continuous and increasing with

\[ \lim_{x \to \infty} g(x) = \infty \]

A sufficient condition for uniform asymptotic stability of the system in Eq. (4.23) is then that the function

\[ \dot{V}(x, t) = f^T(x, t) \text{grad} V + \frac{\partial V}{\partial t} < 0 \quad \text{for} \ x \neq 0 \]

(i.e., that \( \dot{V}(x, t) \)) is negative definite.

Proof of the theorem can be found, e.g., in Hahn (1967) and Vidyasagar (1978), listed in the References at the end of this chapter. When using
Lyapunov theory on adaptive control problems, we often find that $\dot{V}$ only is negative semidefinite. This implies that additional conditions must be imposed on the system. The following lemma gives a useful result.

**Lemma 4.1—Convergence**

If $g$ is a real function of a real variable $t$, defined and uniformly continuous for $t > 0$, 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.** Another way to solve the problem that $\dot{V}$ is only negative semidefinite is to show that $\dot{e} \in L_2$. □

When applying Lyapunov theory to an adaptive control problem, we will 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, the lemma above can be used on $\dot{V}$ to prove stability.

**Design of MRAS Using Lyapunov Theory**

Assuming that all state variables of a system are measured, the Lyapunov stability theorem can be used to design adaptive control laws that guarantee the stability of the closed-loop system. An example illustrates the idea.

**Example 4.6—First-order MRAS based on stability theory**

Consider the same problem as in Example 4.2. When the parameters of the process are known, the control law of Eq. (4.18) gives the desired result. A model-reference adaptive system, which can find the appropriate gains $t_0$ and $s_0$ when the parameters $a$ and $b$ are not known, is obtained as follows. Introduce the error

$$
e = y - y_m$$

Taking derivatives and using Eqs. (4.17) and (4.8) and the desired model to eliminate the derivatives of $y$ and $y_m$ gives

$$
\frac{de}{dt} = -a_m e + (a_m - a - bs_0) y + (bt_0 - b_m) u_c
$$
Notice that the error goes to zero if the parameters are equal to the desired ones. We will now attempt to construct a parameter adjustment mechanism that will drive the parameters $t_0$ and $s_0$ to the desired values. For this purpose the Lyapunov function

$$V(e, t_0, s_0) = \frac{1}{2} \left( e^2 + \frac{1}{b\gamma} (bs_0 + a - a_m)^2 + \frac{1}{b\gamma} (bt_0 - b_m)^2 \right)$$

is introduced. This function is zero when $e$ is zero and the controller parameters are equal to the optimal values. The derivative of $V$ is

$$\frac{dV}{dt} = e \frac{de}{dt} + \frac{1}{\gamma} (bs_0 + a - a_m) \frac{ds_0}{dt} + \frac{1}{\gamma} (bt_0 - b_m) \frac{dt_0}{dt}$$

$$= -a_m e^2 + \frac{1}{\gamma} (bs_0 + a - a_m) \left( \frac{ds_0}{dt} - \gamma ye \right)$$

$$+ \frac{1}{\gamma} (bt_0 - b_m) \left( \frac{dt_0}{dt} + \gamma u_c e \right)$$

If the parameters are updated as

$$\frac{dt_0}{dt} = -\gamma u_c e$$

$$\frac{ds_0}{dt} = \gamma ye$$

we get

$$\frac{dV}{dt} = -a_m e^2$$

The function $V$ will thus decrease as long as the error $e$ is different from zero; it can thus be concluded that the error will go to zero. Notice, however, that it does not follow that the parameters $t_0$ and $s_0$ will converge to the equilibrium values unless more conditions are imposed. The rule is thus similar to the MIT rule, but sensitivity derivatives are replaced by other signals.

Stable parameter adjustment rules for systems in which all the state variables are measurable can be obtained by a direct generalization of the technique used in the example. The adjustment rule of Eq. (4.24), obtained by applying stability theory, is similar to the adjustment law obtained by the MIT rule. (Compare Example 4.2.) In both cases the adjustment law can be written as

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

(4.25)
where $\theta$ is a vector of parameters, and

$$\varphi = [-u_c \ y]^T$$

for the Lyapunov rule and

$$\varphi = \frac{1}{p + a_m} [-u_c \ y]^T$$

for the MIT rule. The vector $\varphi$ can be interpreted as the negative value of the gradient of the loss function.

The Lyapunov method will now be applied to the adjustment of a feedforward gain.

**Example 4.7—Feedforward gain adaptation**

Consider the problem of adjusting a feedforward gain only. The error is given by

$$e = G(p)(\theta - \theta^0)u_c$$

Introduce a state space representation of the transfer function $G$. The relation between the parameter $\theta$ and the error $e$ can then be written as

$$\begin{align*}
\frac{dx}{dt} &= Ax + B(\theta - \theta^0)u_c \\
e &= Cx
\end{align*}$$ \hfill (4.26)

Notice that if the homogeneous system $\dot{x} = Ax$ is asymptotically stable, there exist positive definite matrices $P$ and $Q$ such that

$$A^TP + PA = -Q$$ \hfill (4.27)

Choose the following function as a candidate for a Lyapunov function:

$$V = \frac{1}{2} (\gamma x^TPx + (\theta - \theta^0)^2)$$

The time derivative of $V$ along the differential equation (Eq. 4.26) is given by

$$\frac{dV}{dt} = \frac{\gamma}{2} \left( \frac{dx}{dt}^T P x + x^T P \frac{dx}{dt} \right) + (\theta - \theta^0) \frac{d\theta}{dt}$$

Using Eq. (4.26), we get

$$\begin{align*}
\frac{dV}{dt} &= \frac{\gamma}{2} \left( (Ax + Bu_c(\theta - \theta^0))^T P x + x^T P (Ax + Bu_c(\theta - \theta^0)) \right) \\
&\quad + (\theta - \theta^0) \frac{d\theta}{dt} \\
&= -\frac{\gamma}{2} x^T Q x + (\theta - \theta^0) \left( \frac{d\theta}{dt} + \gamma u_c B^T P x \right)
\end{align*}$$
If the parameter adjustment law is chosen as

\[
\frac{d\theta}{dt} = -\gamma u_c B^T P x
\]  
(4.28)

we thus find that the derivative of the Lyapunov function will be negative as long as \( x \neq 0 \). With the adjustment law of Eq. (4.28), the state vector \( x \) and also the error \( e = C x \) will thus go to zero. Notice, however, that the parameter error \( \theta - \theta^0 \) will not necessarily go to zero. \( \Box \)

**Output Feedback**

The result in Example 4.7 is quite restrictive, because it requires that all state variables be known. It is of interest to obtain adjustment rules that can be based on output feedback only. One special case can be obtained with only a modest effort, and also gives useful hints towards the general solution. The matrix \( P \) in the feedback law of Eq. (4.28) is given by the Lyapunov function of Eq. (4.27). If the Lyapunov function could be chosen such that

\[ B^T P = C \]

where \( C \) is the output matrix of the system in Eq. (4.26), we get

\[ B^T P x = C x = e \]

and the adjustment rule can be written as

\[
\frac{d\theta}{dt} = -\gamma u_c e
\]

Compare with Example 4.1. A condition that allows this is given by the celebrated Kalman-Yakubovich lemma. To state this lemma, the following definition is needed.

**Definition 4.1**

A rational transfer function \( G \) with real coefficients is **positive real** (PR) if \( \text{Re} \ G(s) \geq 0 \) for \( \text{Re} \ s > 0 \). A transfer function \( G \) is **strictly positive real** (SPR) if \( G(s - \varepsilon) \) is positive real for some real \( \varepsilon > 0 \). \( \Box \)

The concept of SPR is further discussed in Appendix B. The following result gives a state-space interpretation of SPR.

**Lemma 4.2—Kalman-Yakubovich**

Let the time-invariant linear system

\[
\frac{dx}{dt} = Ax + Bu
\]
\[
y = C x
\]
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
\[ B^T P = C \]
\[ \square \]
A proof of the Lemma is given in the book by Lefschetz (1965) listed in the References at the end of this chapter. The following result is now obtained.

**Theorem 4.2—MRAS using the SPR rule**
Let the transfer function \( G \) be strictly positive real and assume that the inverse of \( G \) is BIBO stable. Then the parameter adjustment rule
\[ \frac{d\theta}{dt} = -\gamma u_c e \]  
(4.29)
where \( \gamma \) is a positive constant, makes the output error in Eq. (4.26) go to zero. The parameter \( \theta \) also converges to \( \theta^0 \) if the input signal is exciting of order 1. Convergence is exponential if the input signal is persistently exciting of order 1.

**Proof:** The first part of the proof is already completed. For the second part, first observe that \( G \) is stable and SPR. Further, it has a stable inverse. Since
\[ e = G(\theta - \theta^0)u_c \]
and \( e \) goes to zero, it follows that \( (\theta - \theta^0)u_c \) must also go to zero. If \( u_c \) is exciting of order 1, it follows from Chapter 3 and Example 4.3 that \( \theta \) must converge to \( \theta^0 \).

**Remark 1.** Notice that the feedback law of Eq. (4.29) has the characteristic configuration of Figure 1.6, with an integrator between two multipliers.

**Remark 2.** The control law of Eq. (4.29) is very similar to the control law obtained by the gradient method. See Fig. 4.8, which shows block diagrams of both systems.

**Remark 3.** The result is of limited practical value because of the strong assumptions made. If the process dynamics is strictly positive real, it is very easy to design a fixed high-gain regulator that gives a closed-loop response that will work well for a wide range of parameter values. \[ \square \]
Figure 4.8 Block diagrams of the adaptive systems for feedforward gain compensation obtained by (a) the MIT rule and (b) the SPR rule.

Input-output Stability

We will now use the results above to broaden the class of control laws for the MRAS. A short digression will be made in order to develop the necessary mathematical tools. The basic idea is to divide the system into two blocks, as shown in Fig. 4.9. The system is described by the equations

\[
\begin{align*}
    e &= u - Hy \\
    y &= Ge
\end{align*}
\]  

(4.30)

where \( H \) and \( G \) are general nonlinear and time-varying systems. We will first establish properties for each of the two blocks and then use them to conclude the stability of the closed-loop system. The problem is that the subsystems \( G \) and/or \( H \) can be unstable, but the closed-loop system should be stable. The function spaces for \( u \) and \( y \) must then be such that this situation is allowed. This is achieved by introducing the concept of
extended spaces. Let $\mathcal{F}$ be the space of functions from $R$ to $R^n$. Let

$$\langle x \mid y \rangle = \int_0^\infty x^T(s)y(s) \, ds$$

be a scalar product defined for functions in $\mathcal{F}$. Let $P_T$ be the projection operator

$$[P_T x](t) = x_T(t) = \begin{cases} x(t) & t \leq T \\ 0 & t > T \end{cases}$$

The projections define a linear subspace of $\mathcal{F}$

$$P_T \mathcal{F} = \{ x \in \mathcal{F} \mid x(t) = 0 \quad \text{for} \quad t > T \}$$

Furthermore, introduce the spaces

$$L_2 = \{ x \in \mathcal{F} \mid \| x \|^2 = \langle x \mid x \rangle < \infty \}$$

$$L_{2e} = \{ x \in \mathcal{F} \mid x_T \in L_2, \forall T \geq 0 \}$$

The scalar product in $L_{2e}$ is

$$\langle x \mid y \rangle_T = \int_0^\infty x^T_T(s)y_T(s) \, ds$$

To develop a system theory, signals will now be considered as elements of $L_{2e}$. Systems will be general (i.e., nonlinear and time-varying) operators from $L_{2e}$ to $L_{2e}$. The functions in $L_{2e}$ can be regarded as "locally" squared integrable.

**Definition 4.2**

A system is bounded input, bounded output (BIBO) stable if an $L_2$ bounded input gives an $L_2$ bounded output.
Definition 4.3
A system with input $u$ and output $y$ is passive if and only if there exist some constant $\beta$ such that

$$\langle y \mid u \rangle_T \geq \beta$$

for all $u \in L_{2e}$ and all $T$. The system is input strictly passive if and only if there exist $\delta > 0$ and $\beta$ such that

$$\langle y \mid u \rangle_T \geq \delta \| u_T \|^2 + \beta$$

and output strictly passive

$$\langle y \mid u \rangle_T \geq \delta \| y_T \|^2 + \beta$$

for all $u \in L_{2e}$ and all $T$. $\square$

The following result can now be established.

Theorem 4.3—Passivity theorem
Let $G$ be both input and output strictly passive and $H$ be passive. Then the feedback system of Eq. (4.30) is BIBO stable, i.e., $y$ and $e \in L_2$ for all $u \in L_2$.

A proof of the theorem is given in the books by Desoer and Vidyasagar (1975) and Hill and Moylan (1988) listed in the References at the end of the chapter.

Remark. If the input signal $u$ has a bounded $L_2$ norm, it follows from the theorem that the output $y$ also has a bounded $L_2$ norm. This implies that $y(t)$ must go to zero for large $t$ if $y$ is uniformly continuous (see Lemma 4.1). $\square$

Some criteria for a system to be passive will now be given.

Lemma 4.3—Passive linear system
A linear system is passive if its transfer function is positive real.

Proof: We have

$$\langle y \mid u \rangle_T = \int_0^\infty y_T(t)^T u_T(t) \, dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} Y_T(i\omega)^T U_T(-i\omega) \, d\omega$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} G(i\omega)^T U_T(i\omega)^T U_T(-i\omega) \, d\omega$$

$$= \frac{1}{\pi} \int_0^\infty \text{Re} \{ G(i\omega)^T \} U_T(i\omega) U_T(-i\omega) \, d\omega \geq 0$$
where \( Y_T \) and \( U_T \) are the Laplace transforms of \( y_t \) and \( u_t \), respectively. If \( G(\imath \omega) \) is positive real, we have \( \text{Re} G(\imath \omega) \geq 0 \), and we get
\[
\langle y | u \rangle_T \geq 0
\]
which shows that the system is passive.

Remark. Similar characterizations can be obtained for strict passivity. See the last section in this chapter for references. \( \square \)

**Lemma 4.4—Property of SPR systems 1**

Let \( G(s) \) be a strictly positive real rational transfer function. Then the operator \( G \) is output strictly passive.

This lemma gives a connection between SPR and output strict passivity. Also, it can be shown that input strict passivity implies that the inverse of \( G \) is BIBO stable.

**Lemma 4.5—Property of SPR systems 2**

Let \( r \) be an arbitrary function of time and let \( G \) be a transfer function that is (strictly) positive real. The system whose input-output relation is given by
\[
y = r(G(p)ru)
\]
is then (strictly) passive, provided \( r \in L_{2e} \).

**Proof:** It follows that
\[
\langle y | u \rangle_T = \int_0^\infty y_T^T(\tau)u_T(\tau) \, d\tau = \int_0^T (u(\tau)r(\tau))^T(G(p)ru)(\tau) \, d\tau
\]
\[
= \int_0^T w(\tau)^T(G(p)w)(\tau) \, d\tau = \langle w | Gw \rangle_T
\]
where \( w = ru \). It follows from Lemma 4.3 that
\[
\langle w | Gw \rangle_T \geq 0 \quad \text{if } G \text{ positive real}
\]
\[
\langle w | Gw \rangle_T \geq \delta \| w \|_T^2 \quad \text{if } G \text{ strictly positive real} \quad \square
\]

Another interesting result is the following.

**Theorem 4.4—Small gain theorem**

Consider the feedback system in Fig. 4.9. Suppose there exist constants \( \gamma_g \) and \( \gamma_h \) such that \( G \) and \( H \) satisfy the conditions
\[
\| Gx \|_T \leq \gamma_g \| x \|_T \quad \forall T \geq 0 \quad \forall x \in L_{2e}
\]
\[
\| Hx \|_T \leq \gamma_h \| x \|_T \quad \forall T \geq 0 \quad \forall x \in L_{2e}
\]
Under these conditions the system of Eq. (4.30) is $L_2$ stable provided that $\gamma_g \gamma_h < 1$. \hfill \Box

The theorem gives a sufficient condition for stability. It has the interpretation that it is possible to investigate the $L_2$ gain of each subsystem and conclude stability for the closed-loop system if the product of the gains is less than 1. The passivity theorem achieves stability by restricting the phase shift, while the small gain theorem gives stability by restricting the loop gain.

### The Error Model

Having developed suitable tools, we can now return to the problem of finding adjustment rules for the model-reference adaptive system. Consider a system with an adjustable feedforward gain. A parameter adjustment mechanism was given by Theorem 4.2. A block diagram of the system obtained, given in Fig. 4.8, can be redrawn as shown in Fig. 4.10. This figure 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, where $\theta^0$ is the correct value of $\theta$. The other system has the model error $e$ as the input and the quantity $(\theta - \theta^0) u_c$ as the output. This interpretation suggests the use of hyperstability theory for design of adjustment rules. Since an integrator is positive real, it follows from Lemma 4.3 and the fact that $u_c$ is bounded that the system $H$ is passive. If the transfer function $G$ is strictly positive real and has a BIBO stable inverse, it follows from the passivity theorem that the closed-loop system is $L_2$ stable. In Fig. 4.10 there are no external inputs, as in Fig. 4.9. On the other hand, the system in Fig. 4.10 may have initial conditions, because the process and the model may have different initial conditions. There may also be an initial condi-
Figure 4.11 A stable parameter adjustment law is obtained if $GG_c$ is SPR.

tion on the integrator in the adjustment loop. These initial conditions 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 4.3, where the input signal $u$ is bounded in $L_2$. It then follows from the remark of the theorem that the error $e(t)$ will go to zero as $t$ goes to infinity. Since $G$ is strictly positive real, it follows that the signal $(\theta - \theta^0)u_c$ is also $L_2$ bounded. This means that

$$\int_0^\infty u_c^2(\tau)(\theta(\tau) - \theta^0)^2 d\tau < C$$

If $\theta$ does not go to $\theta^0$, we have a contradiction if the command signal is persistently exciting. 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.

The passivity theorem provides a very convenient way to generate stable adjustment laws. We will 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. One example is illustrated in Fig. 4.11. 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. However, for systems with higher pole excess than 1 the compensator required to make $GG_c$ strictly positive real will contain derivatives.

**PI Adjustments**

All adjustment laws discussed so far have been integral regulators. 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
4.4 MRAS Based on Stability Theory

mechanism $H$ in Fig. 4.10. 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. (4.29) is replaced by

$$\theta(t) = -\gamma_1 u_c(t)e(t) - \gamma_2 \int_0^t u_c(\tau)e(\tau) d\tau$$  

(4.31)

Since the transfer function

$$H(s) = \gamma_1 + \gamma_2 / s$$

is positive real for positive $\gamma_1$ and $\gamma_2$, it follows from the passivity theorem that Eq. (4.31) gives a stable adjustment law if $GG_c$ is strictly positive real and has a BIBO stable inverse.

The Augmented Error

Some progress has now been made towards the construction of stable parameter adjustment rules for the simple model-reference adaptive system presented at the beginning of this section. The use of hyperstability theory gave good insight and led to the idea of filtering the model error so that $GG_c$ is SPR. So far, we have not derived any solution that does not require derivative compensation for the case when $G$ has a pole excess larger than 1.

To obtain an adjustment law for a general transfer function $G$, first factor the transfer function $G$ as

$$G = G_1 G_2$$

where the transfer function $G_1$ is SPR. The error $e = y - y_m$ can then be written as

$$e = G(\theta - \theta^0)u_c = G_1 G_2 (\theta - \theta^0)u_c$$

$$= G_1 \left( (G_2 (\theta - \theta^0)u_c) + (\theta - \theta^0)G_2 u_c - (\theta - \theta^0)G_2 u_c \right)$$

$$= G_1 \left( ((\theta - \theta^0)G_2 u_c) - G_1 ((\theta - \theta^0)G_2 u_c - G_2 (\theta - \theta^0)u_c) \right)$$

Introduce the augmented error $\varepsilon$ defined by

$$\varepsilon = e + \eta$$

where $\eta$ is the error augmentation defined by

$$\eta = G_1 (\theta - \theta^0)G_2 u_c - G(\theta - \theta^0)u_c$$

$$= G_1 (\theta G_2 u_c) - G\theta u_c$$

The second equality follows, because $G\theta^0 u = \theta^0 Gu$ when $\theta^0$ is constant. The augmented error is thus obtained by adding a correction term $\eta$ to
the error. Notice that the correction term vanishes when the parameter $\theta$ is constant. It follows that

$$\varepsilon = G_1((\theta - \theta^0)G_2u_c) = G_1(\theta - \theta^0)\tilde{u}_c$$

where $\tilde{u}_c$ is the reference signal filtered through $G_2$. This equation is an error model similar to the ones used previously.

**Theorem 4.5—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 and has a BIBO stable inverse. The parameter adjustment law

$$\frac{d\theta}{dt} = -\gamma \varepsilon(G_2u_c)$$

where

$$\varepsilon = e + G_1(\theta G_2u_c) - G(\theta u_c)$$

gives a closed-loop system where the error goes to zero as $t$ goes to infinity. If the command signal $u_c$ is exciting and if $G_2$ has no zeros in the right half-plane, the parameter $\theta$ also converges to $\theta^0$.

**Proof:** Since $G_1$ is SPR and has a BIBO stable inverse, the discussion of the error model shows that $\varepsilon(t) \to 0$ and that $\varepsilon \in L_2$. The parameter also converges, which implies that $e$ also goes to zero. If $G_2$ has no zeros in the right half-plane and if $u_c$ is exciting, it follows from Theorem 3.8 that $G_2u_c$ is also exciting. The same argument that was used in Theorem 4.2 can then be used to show that $\theta$ converges to $\theta^0$.

**Remark 1.** The trivial factorization with $G_1 = 1$ is one possibility. A block diagram of the system with augmented error is shown in Fig. 4.12. To implement the augmented error it is necessary to introduce realizations of the transfer functions $G_1$ and $G_2$.

**Remark 2.** Notice that $G_2$ must be minimum phase in order to establish that $\theta$ converges to $\theta^0$. 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 4.5. \(\Box\)

**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. Design based on Lyapunov theory was first considered. This approach is based on finding suitable candidates for Lyapunov functions. It was shown that systems with a stable equilibrium could be obtained.
The parameter adjustment rules were similar to those obtained by the gradient method.

By introducing the error model in Fig. 4.9 it is possible to use passivity theory if the system $H$ is passive. The strictly positive real condition of $G$ or $G_cG$ is an important condition in the design. If the pole excess of $G$ is greater than 1 the compensation network $G_c$ introduces derivatives of the error signal. The augmented error may then be introduced to avoid derivations. Theorem 4.5 shows that the augmented error method gives convergence for the problem discussed. The arguments are also true in general, provided that $\eta$ and $e$ are bounded. However, it can happen that $e$ converges to zero but $\eta$ and $e$ can be unbounded. This problem was unnoticed for a long time and was finally settled around 1980. The full problem is discussed further in Chapter 6.

The Lyapunov theory can of course also be used to analyze the systems with augmented error, and a suitable Lyapunov function can be derived. The passivity approach has the advantage that the stability problem is divided into two parts. The first problem is to show that one part of the system is passive; the second problem is to make one part of the system strictly positive real. It should also be emphasized that the SPR condition is quite restricted, since small perturbations or unmodeled dynamics can make the system no longer strictly positive real.

### 4.5 Direct MRAS for General Linear Systems

The previous sections discussed some of the stability and convergence problems for the MRAS using a simplified example. In this section we will return to the MRAS problem discussed in Section 4.2. A direct MRAS for
general single input, single output (SISO) linear systems is derived. This will be done by combining a parameter estimator with a design principle. This approach will give a natural interpretation of the augmented error, for example.

The method for a direct MRAS can be described in the following steps:

**Step 1:** Find a regulator structure that admits perfect model-following for the plant.

**Step 2:** Derive an error model of the form

$$
\varepsilon = G_1(p) \{ \varphi^T(t)(\theta^0 - \theta) \}
$$

where $G_1$ is a strictly positive real transfer function, $\theta^0$ the true parameters, and $\theta$ the actual parameters. The right-hand side should be expressed in computable quantities.

**Step 3:** Use the parameter adjustment law

$$
\frac{d\theta}{dt} = \gamma \varphi \varepsilon
$$

or

$$
\frac{d\theta}{dt} = \gamma \frac{\varphi \varepsilon}{\alpha + \varphi^T \varphi}
$$

which is called the SPR rule.

Notice that the error model of Eq. (4.32) is linear in the parameters. This imposes restrictions on the models and regulators that can be dealt with. The tricks used to derive the error model include filtering and error augmentation. Notice that the error may be nonlinear in the parameters if gradient methods are used. The tools for the different steps have been treated in previous chapters and sections of the book. What remains is a systematic procedure and analysis of the algorithm. One essential part of the procedure is to find a model that is linear in the parameters.

We will now derive an MRAS for minimum-phase SISO systems along the lines outlined in Section 4.2. It is assumed that the plant is described by

$$
Ay = Bu
$$

where the polynomial $B$ is assumed to have all its zeros in the left half-plane. It is convenient to write $B$ as

$$
B = b_0 B^+$$
where the polynomial $B^+$ is monic and stable. The variable $b_0$ is called the *instantaneous gain* or the *high-frequency gain*. The model, which gives the desired response to command signals, is given by

$$A_m y_m = B_m u_c$$  \hfill (4.36)

**Finding a Regulator Structure**

The first step in the design procedure is to find a suitable regulator structure. To do this we will first consider the model-following design problem when the process model is known. This problem was discussed and solved in Section 4.2 and Appendix A using the pole placement design procedure. To carry out the design, an observer polynomial $A_o$ must also be specified. Since the process was assumed to be minimum phase, a regulator that cancels all the process zeros can be used. The regulator is then given by Eq. (4.3) where

$$R = R_1 B^+$$
$$T = A_o B_m / b_0$$  \hfill (4.37)

and $R_1$ and $S$ satisfy

$$A R_1 + b_0 S = A_o A_m$$  \hfill (4.38)

The compatibility conditions for the pole placement design must also be satisfied. This implies that the pole excess of the model cannot be smaller than the pole excess of the process. The degree of the observer polynomial must also be sufficiently large, i.e.,

$$\deg A_o \geq 2 \deg A - \deg A_m - \deg B^+ - 1$$

**The Error Model**

Having obtained a suitable regulator structure, we will now proceed to derive an error model. It follows from Eq. (4.38) that

$$A_o A_m y = A R_1 y + b_0 S y = R_1 B u + b_0 S y = b_0 (R u + S y)$$  \hfill (4.39)

where the second equality follows from Eq. (4.35) and the third from Eq. (4.37). Furthermore, it follows from Eq. (4.36) that

$$A_o A_m y_m = A_o B_m u_c = b_0 T u_c$$

We thus get

$$A_o A_m e = A_o A_m (y - y_m) = b_0 (R u + S y - T u_c)$$
or
\[
e = \frac{b_0}{A_o A_m} (R u + S y - T u_c)
\]
This expression is not yet a suitable error model, because the transfer function \(b_0/A_o A_m\) 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 \(\text{deg} A_o A_m\), such that
\[
\frac{b_0 Q}{A_o A_m}
\]
is SPR. The filtered error can be written as
\[
e_f = \frac{b_0 Q}{A_o A_m} \left( \frac{R}{P} u + \frac{S}{P} y - \frac{T}{P} u_c \right)
\]
Let \(P = P_1 P_2\) where \(P_2\) is a stable monic polynomial of the same degree as \(R\). Rewrite \(R/P\) as
\[
\frac{R}{P} = \frac{R - P_2 + P_2}{P_1 P_2} = \frac{1}{P_1} + \frac{R - P_2}{P}
\]
The filtered error then becomes
\[
e_f = \frac{b_0 Q}{A_o A_m} \left( \frac{1}{P_1} u + \frac{R - P_2}{P} u + \frac{S}{P} y - \frac{T}{P} u_c \right)
\]
Let \(k, l, \) and \(m\) be the degrees of the polynomials \(R, S,\) and \(T,\) respectively. Introduce a vector of nominal regulator parameters
\[
\theta^0 = (r'_1 \ldots r'_k s_0 \ldots s_l t_0 \ldots t_m)^T
\]
where \(r'_i\) are the coefficients of the polynomial \(R - P_2\). Also introduce a vector of filtered input, output, and command signals
\[
\phi^T = \left( \frac{p^{k-1}}{P(p)} u \ldots \frac{1}{P(p)} u \frac{p^l}{P(p)} y \ldots \frac{1}{P(p)} y - \frac{p^m}{P(p)} u_c \ldots - \frac{1}{P(p)} u_c \right)
\]
The filtered error can then be written as
\[
e_f = \frac{b_0 Q}{A_o A_m} \left( \frac{1}{P_1} u + \phi^T \theta^0 \right)
\]
To obtain an error model, we must introduce a parameterization of the regulator. In the nominal case when the parameters are known, the control law can be expressed as

\[ u = -P_1(\varphi^T \theta^0) = -P_1((\theta^0)^T \varphi) = -(\theta^0)^T(P_1 \varphi) \quad (4.44) \]

where \( P_1 \) is a polynomial in the differential operator. Let \( \theta \) denote the adjustable regulator parameters. The feedback law

\[ u = -P_1(\varphi^T \theta) \]

would give the desired error model. This control law is, however, not realizable if \( P_1 \) has a degree greater than 1, because the term \( P_1 \theta \) implies taking derivatives of the parameters. However, the control law

\[ u = -\theta^T(P_1 \varphi) \quad (4.45) \]

is realizable because of Eq. (4.42). Using this control law, it follows from Eq. (4.43) that the filtered error can be written as

\[ e_f = \frac{b_0Q}{A_o A_m} \left( \varphi^T \theta^0 - \frac{1}{P_1} \theta^T(P_1 \varphi) \right) \]

\[ = \frac{b_0Q}{A_o A_m} \left( \varphi^T \theta^0 - \varphi^T \theta - \frac{1}{P_1} \theta^T(P_1 \varphi) + \varphi^T \theta \right) \]

Introduce the signals \( \eta \) and \( \varepsilon \), defined by

\[ \eta = \frac{1}{P_1} \theta^T(P_1 \varphi) - \varphi^T \theta = - \left( \frac{1}{P_1} u + \varphi^T \theta \right) \]

\[ \varepsilon = e_f + \frac{b_0Q}{A_o A_m} \eta = \frac{b_0Q}{A_o A_m} \varphi^T(\theta^0 - \theta) \quad (4.46) \]

The signal \( \varepsilon \) is called the augmented error, and \( \eta \) is called the error augmentation. The augmented error is computed as follows:

\[ \varepsilon = \frac{Q}{P} (y - y_m) + \frac{b_0Q}{A_o A_m} \eta \quad (4.47) \]

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. (4.46) is also linear in the parameters, and the transfer function \( b_0Q/(A_o A_m) \) is SPR. The error model thus satisfies the requirements of Step 2, and the parameters can then be updated by
Eq. (4.33) or Eq. (4.34). So far, the derivation has been done along the lines developed in Sections 4.3 and 4.4. To show the stability of the closed-loop system, it is not sufficient that Eq. (4.40) is SPR. It is also necessary that the signals in $\varphi$ are bounded. This condition can be difficult to show. Furthermore, Eq. (4.46) is valid only if the control signal is generated from Eq. (4.45). This implies, for instance, that the control signal cannot be saturated. Notice that it is necessary to know the parameter $b_0$ in order to compute the augmented error $\varepsilon$.

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. (4.43) can be written as

$$e_f = b_0 (\varphi_f^T \theta^0 + u_f) \quad (4.48)$$

where

$$\varphi_f = \frac{Q}{A_o A_m} \varphi$$

$$u_f = \frac{Q}{A_o A_m P_1} u$$

A simple gradient estimator for $b_0$ and $\theta^0$ is then given by

$$\frac{d\theta}{dt} = \gamma \hat{b}_0 \varphi_f \varepsilon_p = \gamma \varphi_f \varepsilon_p$$

$$\frac{d\hat{b}_0}{dt} = \gamma (\varphi_f^T \theta + u_f) \varepsilon_p$$

(4.49)

where $\varepsilon_p$ is the prediction error

$$\varepsilon_p = e_f - \hat{e}_f = e_f - \hat{b}_0 (\varphi_f^T \theta + u_f)$$

Notice that $\hat{b}_0$ can be absorbed in the adaptation gain if its sign is known.

**An Interpretation of the Augmented Error**

Equation (4.49), obtained by parameter estimation, is very similar to the equation obtained by the model-reference approach. The differences are that the regressors are filtered in Eq. (4.49) and that the augmented error $\varepsilon$ is replaced by the prediction error $\varepsilon_p$. The approaches will be identical if $b_0$ is known and $Q = A_o A_m$. It follows from Eq. (4.48) that $\varphi = \varphi_f$. Furthermore $\varepsilon = \varepsilon_p$, since

$$\varepsilon_p = e_f - b_0 (\varphi^T \theta + u_f) = b_0 \varphi^T (\theta^0 - \theta) = \varepsilon$$
The augmented error that was derived by using a collection of tricks is thus identical to the prediction error that occurs naturally in parameter estimation. Also notice that the updating formula obtained from parameter estimation does not depend on how the signal $u$ is generated.

**Realization**

The equations needed to implement the general MRAS can now be summarized:

$$y_m = \frac{B_m}{A_m} u_c$$

$$e_f = \frac{Q}{P} e = \frac{Q}{P} (y - y_m)$$

$$\eta = -\left(\frac{1}{P_1} u + \varphi^T \theta\right)$$

$$\varepsilon = e_f + \frac{b_0 Q}{A_o A_m} \eta$$

$$\dot{\theta} = \gamma \varphi \varepsilon$$

$$u = -\theta^T(P_1 \varphi)$$

A block diagram of the model-reference adaptive system is shown in Fig. 4.13. The block labeled “Filter” in Fig. 4.13 is a linear system, which generates $P_1 \varphi$ and $\varphi$ from the signals $u_c$, $u$, and $y$. The vector $\varphi$ is composed of three parts having the same structure. It therefore suffices
to discuss one part. Consider, e.g., how to generate $\varphi_u$ 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

$$P_1 = p^n + \alpha_1 p^{n-1} + \cdots + \alpha_n$$
$$P_2 = p^k + \beta_1 p^{k-1} + \cdots + \beta_k$$

We also assume that $\deg P_1 > \deg P_2$. The vectors $x$ and $\varphi_u$ can then be realized as follows:

$$\frac{dx}{dt} = \begin{pmatrix} -\beta_1 & -\beta_2 & \cdots & -\beta_{k-1} & -\beta_k \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} x + \begin{pmatrix} 1 \\ \vdots \\ 0 \end{pmatrix} u$$

$$\frac{dz}{dt} = \begin{pmatrix} -\alpha_1 & -\alpha_2 & \cdots & -\alpha_{n-1} & -\alpha_n \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} z + \begin{pmatrix} 1 \\ \vdots \\ 0 \end{pmatrix} x_k$$

where $x_k = 1/P_2 \cdot u$ is the last element of the $x$ vector. The elements of $\varphi_u$ are the $k$ last elements of the state vector $z$. Furthermore $1/P_1 \cdot u$ can also be obtained from the generation of $\varphi_u$ and $P_1\varphi_u$. To generate the full vectors $\varphi$ and $P_1\varphi$, we will thus need three realizations of the transfer functions $P_1$ and $P_2$. The block labeled “Filter” in Fig. 4.13 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 regulator complexity (i.e. the degrees of the polynomials $R$, $S$, and $T$)
- The polynomials $P_1$, $P_2$, and $Q$. 
Many different model-reference adaptive systems can be obtained by the
different choices of the design parameters. A popular choice of the poly-
nomials is \( P_1 = A_m, P_2 = A_o, \) and \( Q = A_o A_m. \)

**A Priori Knowledge**

To apply the MRAS procedure it is necessary to know the following prior
information:

- The instantaneous gain \( b_0 \)
- That plant is minimum phase
- The pole excess of the plant
- The order of the plant or the regulator complexity.

**Example 4.8—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 as

\[
A_o(s) = s + a_o
\]

\[
R(s) = s + r_1
\]

\[
S(s) = s_0 s + s_1
\]

\[
T(s) = t_0 s + t_1
\]

The Diophantine equation (Eq. 4.8) gives the solution

\[
r_1 = 2\zeta \omega + a_o - a
\]

\[
s_0 = (2\zeta \omega a_o + \omega^2 - ar_1)/K
\]

\[
s_1 = a_o \omega^2/K
\]

\[
t_0 = \omega^2/K
\]

\[
t_1 = a_o \omega^2/K
\]

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 4.14 Simulation of the system in Example 4.8. (a) The process output and the model output. (b) The control signal. (c) The error $e = y - y_m$.

Figure 4.14 shows a simulation of the system with $\gamma = 1$, $\zeta = 0.7$, $\omega = 1$, $a_0 = 2$, $a = 1$, and $K = 2$. In the simulation it is assumed that $\delta_0 = b_0$. 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 away from the optimal values, but the error is small (see Fig. 4.14c). The control law at $t = 150$ gives a closed-loop system with a pole in $-1.95$ and two complex poles corresponding to $\omega = 0.84$ and $\zeta = 0.78$, which should be compared to the roots of $A_oA_m$, which are in $-2$, and complex poles corresponding to $\omega = 1$ and $\zeta = 0.7$.

The implementation of the MRAS used in the simulation is given in Listing 4.2.
CONTINUOUS SYSTEM gmras

"To simulate a general MRAS where
"    G(s)=K/[s(s+a)]
"    Gm(s)=om*om/(s*s + 2*z*om*s +om*om)
"    Ao(s)=s+ao
"    P1(s)=Am=s*s+p11*s+p12
"    P2(s)=Ao=s+p2
STATE x1 x2 xm1 xm2 xc xu xy
STATE th1 th2 th3 th4 th5
STATE zc1 zc2 zu1 zu2 zy1 zy2
DER dx1 dx2 dmx1 dmx2 dxc dxu dxy
DER dth1 dth2 dth3 dth4 dth5
DER dzc1 dzc2 dzu1 dzu2 dzy1 dzy2
TIME t

"Generation of fi=[fi1 fi2 fi3 fi4 fi5],
"P1*fi=[fd1 fd2 fd3 fd4 fd5], and uf=(1/P1)*u
p11=2*z*om
p12=om*om
p2=ao
dxc=-p2*xc+uc
dxu=-p2*xu+u
dxy=-p2*xy+y
fd1=uc-p2*xc
fd2=xc
fd3=-xu
fd4=-y+p2*xy
fd5=-xy
dzc1=-p11*zc1-p12*zc2+xc
dzc2=zc1
dzu1=-p11+zu1-p12+zu2+xu
dzu2=zu1
dzy1=-p11+zy1-p12+zy2+xy
dzy2=zy1
fi1=zc1
fi2=zc2
fi3=-zu2
fi4=-zy1
fi5=-zy2
uf=zu1+p2*zu2

Listing 4.2 A Simmon program to implement a model-reference adaptive system.
"The system
dx1=-a*x1+K*u
dx2=x1
y=x2
"The model
dx1=-2*z*om*xm1-om*om*xm2+om*om*uc
dx2=xm1
ym=xm2
uc=IF mod(t,per){per/2 THEN step ELSE -step "ref signal
"The residual
ef=y-ym
eta=-(uf-th1*fi1-th2*fi2-th3*fi3-th4*fi4-th5*fi5)
eps=ef+b0*eta
"The control signal
ut=th1*fd1+th2*fd2+th3*fd3+th4*fd4+th5*fd5
ul=IF 0<ut<1 THEN 1 ELSE IF ut<0 THEN ut ELSE ul
"Parameter update
den=fi1*fi1+fi2*fi2+fi3*fi3+fi4*fi4+fi5*fi5+alfa
ths1=-gamma*fi1*eps/den
ths2=-gamma*fi2*eps/den
ths3=-gamma*fi3*eps/den
ths4=-gamma*fi4*eps/den
ths5=-gamma*fi5*eps/den
"Parameters
a:1
K:2
b0:2
om:2
z:0.7
ao:3
gamma:1
ul:5
alfa:0.1
per:20
step:1
END

Listing 4.2  Cont'd

Discrete-time MRAS

The MRAS discussed has been derived for continuous-time systems without noise, but it is also possible to make discrete-time MRAS. The same type of algorithm as above can be used also in the discrete-time case. The
estimation can then be based on least-squares estimation. This discussion is deferred until self-tuning regulators are treated in Chapter 5.

4.6 MRAS for Partially Known Systems

In the previous sections it has been assumed that the whole process model is unknown. In many situations part of the dynamics is known while other parts are unknown. This a priori knowledge can also be incorporated into the MRAS. How this can be done depends crucially on the parameterization and the structure of the process model. The methodology is illustrated by an example.

Adaptive Manipulator Control

Provided that all state variables are measured, it is often possible to find an error variable that is linear in the parameters, which can facilitate the construction of stable model-reference adaptive systems. This is illustrated by the problem of manipulator control when the dynamics are nonlinear.

A direct drive manipulator can be described by the model

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = T$$

(4.50)

where \(q\) is a vector of generalized coordinates, \(H\) the inertia matrix, \(C\) the damping matrix, and \(G\) the gravity vector. The control variable is torque \(T\) applied by the actuators. The manipulator equations have the property that

$$\frac{1}{2} \frac{d}{dt} (\dot{q}^T H \dot{q}) = \dot{q}^T H(q) \ddot{q} + \dot{q}^T C(q, \dot{q}) \dot{q} = \dot{q}^T (T - G)$$

(4.51)

Physically, this property can be interpreted as the derivative of the kinetic energy \(\dot{q}^T H \dot{q}\) is equal to the power input provided by the actuators and gravitational torques.

Example 4.9—Two-link manipulator

Consider the two-link manipulator with an unknown load in Fig. 4.15. The second link with the unknown load can be regarded as an augmented link with four unknown parameters, mass \(m_e\), moment of inertia \(I_e\), the distance \(\ell_{ce}\) of its mass center to the second joint, and the angle \(\delta_e\) relative to the original second link. The system is described by Eq. (4.50), with

$$H = \begin{pmatrix}
\theta_1 + 2\theta_3 \cos q_2 + 2\theta_4 \sin q_2 & \theta_2 + \theta_3 \cos q_2 + \theta_4 \sin q_2 \\
\theta_2 + \theta_3 \cos q_2 + \theta_4 \sin q_2 & \theta_2
\end{pmatrix}$$
Figure 4.15 Two-link planar manipulator with unknown load.

\[ G = \begin{cases} \theta_3 Y_1 + \theta_4 Y_2 + (\theta_1 - \theta_2 + e_1) e_2 \cos \theta_1 \\ \theta_3 Y_3 + \theta_4 Y_4 \end{cases} \]

where

\[ Y_1 = -2 \sin(q_2) \dot{q}_1 \dot{q}_2 - \sin(q_2) \dot{q}_2^2 \]
\[ Y_2 = 2 \cos(q_2) \dot{q}_1 \dot{q}_2 + \cos(q_2^2) \]
\[ Y_3 = \sin(q_2) \dot{q}_1^2 + e_2 \cos(q_1 + q_2) \]
\[ Y_4 = -\cos(q_2) \dot{q}_1^2 + e_2 \sin(q_1 + q_2) \]
\[ e_1 = m_1 l_1 l_{c1} - m_1 l_{c1}^2 \]
\[ e_2 = g/l_1 \]

where \( g \) is the normal acceleration and the four unknown parameters \( \theta_1, \ldots, \theta_4 \) are functions of the unknown physical parameters

\[ \theta_1 = I_1 + m_1 l_{c1}^2 + I_e + m_e l_{ce}^2 + m_e l_1^2 \]
\[ \theta_2 = I_1 + m_e l_{ce}^2 \]
\[ \theta_2 = m_e l_1 l_{ce} \cos \delta_e \]
\[ \theta_4 = m_e l_1 l_{ce} \sin \delta_e \]

The four unknown parameters \( m_e, I_e, l_{ce}, \) and \( \delta_e \) are uniquely determined by \( \theta_1, \ldots, \theta_4 \). The system can be written

\[ \varphi^T(q, \dot{q}, \ddot{q})\theta = T \]
where $\varphi^T$ is given by

$$
\begin{pmatrix}
\ddot{q}_1 + e' & \ddot{q}_2 - e' & 2\cos(q_2)\dot{q}_1 + \cos(q_2)\dot{q}_2 + Y_1 & 2\sin(q_2)\dot{q}_1 + \sin(q_2)\dot{q}_2 + Y_2 \\
0 & \ddot{q}_1 + \ddot{q}_2 & \cos(q_2)\dot{q}_1 + Y_3 & \sin(q_2)\dot{q}_1 + Y_4
\end{pmatrix}
$$

$$
\theta = \begin{pmatrix} \theta_1, \theta_2, \theta_3, \theta_4 \end{pmatrix}^T
$$

$$
T = \begin{pmatrix} \tau_1 - e_1 e_2 \cos q_1 \\
\tau_2 \end{pmatrix}
$$

where $e' = e_2 \cos(q_1)$ and $\tau_1$ and $\tau_2$ are the applied torques. The dynamics can thus be written in a form that is linear in the parameters, provided that all states and the accelerations can be measured.

The example can be generalized, and Eq. (4.50) can be written as

$$
T - H'(q)\ddot{q} - C'(q, \dot{q})\dot{q} - G'(q) = \varphi^T(q, \dot{q}, \ddot{q})\theta^0
$$

(4.52)

where $H'$, $C'$, $G'$, and $\varphi$ are known or measurable. Even if the model is nonlinear, it is linear in the parameters that may vary. It is very important that this a priori knowledge be used and that the system not be considered as a black box model with time-varying parameters. The model is still not satisfactory, since the accelerations have to be measured together with positions and velocities.

Let the reference trajectories for the positions and velocities be $q_m$ and $\dot{q}_m$. Introduce the following Lyapunov function candidate:

$$
V = \frac{1}{2} \left( \ddot{q}^T H(q) \dot{q} + \dot{q}^T K_p \ddot{q} + \Gamma \dot{\theta}^T \dot{\theta} \right)
$$

where $\ddot{q} = q - q_m$, $\dot{q} = \dot{q} - \dot{q}_m$, $\dot{\theta} = \theta - \theta^0$, and $K_p$ and $\Gamma$ are positive definite matrices. Differentiating $V$ and using Eq. (4.51) gives

$$
\dot{V} = \ddot{q}^T H \ddot{q} + \frac{1}{2} \ddot{q}^T \dot{H} \dot{q} + \dot{q}^T K_p \ddot{q} + \dot{\theta}^T \Gamma \dot{\theta} 
$$

$$
= \ddot{q}^T (H \ddot{q} - H \ddot{q}_m + C \dot{q} + K_p \ddot{q}) + \dot{\theta}^T \Gamma \dot{\theta} 
$$

$$
= \ddot{q}^T (T - G - H \ddot{q}_m - C \dot{q}_m + K_p \ddot{q}) + \dot{\theta}^T \Gamma \dot{\theta} 
$$

Introduce the control law

$$
T = H'\ddot{q}_m + C'\dot{q}_m + G' - K_p \ddot{q} - K_D \dot{q}
$$

(4.53)
The control law contains a feedforward term from the known part of the model and a proportional and velocity feedback term. This implies that

\[ \dot{V} = \ddot{q}^T \left( \dddot{H} \ddot{q} - \dddot{C} \dot{q} + \dddot{G} - K_d \ddot{q} \right) + \ddot{\theta}^T \Gamma \ddot{\theta} \]

where

\[ \dddot{H}(q) = H'(q) - H(q) \]
\[ \dddot{C}(q, \dot{q}) = C'(q, \dot{q}) - C(q, \dot{q}) \]
\[ \dddot{G}(q) = G'(q) - G(q) \]

Introduce

\[ \dddot{H} \ddot{q} - \dddot{C} \dot{q} + \dddot{G} = \varphi_m^T \dddot{\theta} \]

This can be done because the model is linear in the parameters. Furthermore \( \varphi_m = \varphi_m(q, \dot{q}, \ddot{q}_m, \dddot{q}_m) \), which implies that only the accelerations of the model trajectory must be known, not the true accelerations. This gives

\[ \dot{V} = -\ddot{q}^T K_d \ddot{q} + \ddot{\theta}^T \left( \Gamma \ddot{\theta} + \varphi_m^T \dddot{\theta} \right) \]

which suggests the parameter updating

\[ \dot{\theta} = \dot{\ddot{\theta}} = -\Gamma^{-1} \varphi_m^T \dddot{q} = -\Gamma^{-1} \varphi_m^T (\dddot{q} - \dddot{q}_m) \quad (4.54) \]

The candidate for the Lyapunov function has the property that it is positive definite and that the derivative

\[ \ddot{V} = -\dddot{q} K_d \ddot{q} \]

is negative semidefinite. This implies that the closed-loop system is stable and that the steady-state velocity errors are zero. The controller can also be modified to guarantee that the position errors will be zero too.

The control law of Eq. (4.53) and the parameter update of Eq. (4.54) are functions of \( q, \dot{q}, q_d, \dot{q}_d, \) and \( \ddot{q}_d \), but the joint accelerations do not need to be measured. Notice that the control law is a special case of the general MRAS, with \( e = \dddot{q} - \dddot{q}_m \).

**Summary**

The adaptive manipulator control problem shows how a priori knowledge can be incorporated in a model-reference adaptive controller if the full state vector is available. Other ways to use a priori knowledge will be discussed in connection with self-tuning regulators in Chapter 5. The problem also shows the importance of using as much a priori knowledge as
possible. In Example 4.9 it was seen that the model was linear in the four unknown variables. These variables change only occasionally, when the load is changed. If the two-arm example is treated as a black box model, it will contain two second-order differential equations with six unknown parameters. These parameters are dependent on \( q \) and \( \dot{q} \) as well as on the unknown load. Thus the parameters will change very rapidly, and it will be virtually impossible to get good tracking of the reference trajectories.

### 4.7 Conclusions

The fundamental ideas behind the MRAS have been covered in this chapter, including

- Gradient methods
- Lyapunov and hyperstability design
- 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 \( \varphi \) 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 \( \varphi \) is a regression vector, which is found by filtering inputs, outputs, and command signals. The quantity \( \varepsilon \) 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 rule is based on a gradient calculation, it can immediately be asserted that the method will converge, provided that the adaptation gain \( \gamma \) is chosen sufficiently small. Further, the initial values of the parameters must be such that the closed-loop system is stable. The method may be unstable if the adaptation gains are too high. The problem is that it is difficult to find the stability limits a priori.

A general MRAS is derived in Section 4.5, based on the model-following design in Section 4.2. 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 other ways than those given in
Eqs. (4.33) and (4.34). Various modifications will be discussed in Chapter
6.

Problems

4.1 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.

4.2 Consider the system in Example 4.2. Use simulations to investigate
how the phase plane for \( t_0 \) and \( s_0 \) depends on the gain \( \gamma \).

4.3 Consider the system in Example 4.4. 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, when \( u_c \) and \( y_m \) were constant.

4.4 Let the process be described by

\[
\begin{align*}
\dot{x} &= A(\theta)x + B(\theta)u_c \\
y &= C(\theta)x
\end{align*}
\]

and let the model be

\[
\begin{align*}
\dot{x}_m &= A_m x_m + B_m u_c \\
y_m &= C_m x_m
\end{align*}
\]

Assume that the orders of the process and the model are the same
and that all states are measurable.

(a) Derive conditions in which perfect model-following of the states
can be achieved (i.e., for \( e - x - x_m \)).

(b) Use Lyapunov theory to derive stable adjustment mechanism,
(i.e., determine \( \dot{A}(\theta) \) and \( \dot{B}(\theta) \)).

*Hint:* Use as Lyapunov function

\[ V = e^T Pe + tr(A - A_m)^T Q_a (A - A_m) + tr(B_m - B)^T Q_b (B_m - B) \]

where \( e = x - x_m \).
4.5 Consider the simple MRAS in Fig. 4.4 with \( G = 1/s \). Let the parameter adjustment law be Eq. (4.31) (i.e., of PI type). Determine the differential equation for \( \theta \) and discuss how \( \gamma_1 \) and \( \gamma_2 \) influence the convergence rate.

4.6 Investigate through simulation the convergence rate of the parameters in Example 4.2 when the control law of Eq. (4.19) 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.

4.7 Consider the system

\[
G(s) = G_1(s)G_2(s)
\]

where

\[
G_1(s) = \frac{b}{s + a} \\
G_2(s) = \frac{q}{s + p}
\]

where \( a \) and \( b \) are unknown parameters and \( q \) and \( p \) are known. Discuss how to make an MRAS based on the gradient approach. Let the desired model be described by

\[
G_m(s) = \frac{\omega^2}{s^2 + 2\zeta \omega s + \omega^2}
\]

4.8 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.

4.9 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 regulator

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

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 a model-reference adaptive system that gives the desired result and investigate the system by simulation.

4.10 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

$$A = \begin{bmatrix}
-a_1 & -a_2 & \cdots & -a_{n-1} & -a_n \\
1 & 0 & 0 & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 1 & 0 & 0
\end{bmatrix}$$

and $Q$ is a symmetric positive definite matrix.

4.11 The general MRAS procedure in Section 4.5 was derived for known instantaneous gain $b_0$. If $b_0$ is unknown, we may use the following augmented error:

$$\varepsilon = \frac{Q}{A_0 A_m} \left( (b_0 - \hat{b}_0) \left( \varphi^T \theta + \frac{u}{P_1} \right) + b_0 \varphi^T (\theta - \theta^0) \right)$$

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 $\theta$.

4.12 Consider the process

$$G(s) = \frac{50}{s(s + 4)}$$

and the criteria

$$\int_0^\infty (y - u_c)^2 + \rho u^2 \, dt$$

Let the control law have the form

\[ u(t) = -s_0(y - u_c) \]

or

\[ u(t) = -\frac{s_0p + s_1}{p + \tau_1} (y - u_c) \]

Determine the regulator parameters through explicit minimization of the criteria, and let the gradients be obtained from an estimated model of the process. *(Hint: see Trulsson and Ljung, 1985.)*

4.13 Consider the system in Example 4.8. Figure 4.14(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.

4.14 Consider the system in Problem 2.7 and design an MRAS that fulfills the specifications. Investigate the possibility of obtaining a rapid convergence to the desired closed-loop response. How sensitive will the system be for disturbances and unmodeled dynamics?

4.15 Redo Problem 4.14 with the process given in Problem 2.8.

**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:


The problem with stability of the gradient method was analyzed using Lyapunov stability theory in:


The stability problems had also been pointed out in:


The different approaches to MRAS are treated in:


A comparison of the Lyapunov and the input-output stability approaches is given in:


The augmented error method was introduced in:


A unification of MRAS and self-tuning regulators is found in:


Stability of continuous-time MRAS is discussed in:


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:


The error model plays an important role in the analysis of the MRAS. Different generic error models are discussed in:

References


PI adjustment of the parameters in the MRAS is discussed in Landau (1979) (see above) and in:


Lyapunov theory and passivity theory are basic tools for the stability analysis. Some general references are:


The early work on passivity and input-output stability was done by Sandberg and Zames. The theory is summarized in:


A proof of the Kalman-Yakobovich Lemma is given in


Discrete-time MRAS is discussed in detail in Egardt (1979) and Landau (1979), given above.

The explicit criterion minimization approach to adaptive control can be found in:


The example with adaptive manipulator control is adapted from:


A related approach is taken in:


Chapter 5

SELF-TUNING REGULATORS

5.1 The Basic Idea

In an adaptive system it is assumed that the regulator parameters are adjusted all the time. This implies that the regulator parameters follow changes in the process. However, it is difficult to analyze the convergence and stability properties of such systems. To simplify the problem it can be assumed that the process has constant but unknown parameters. When the process is known, the design procedure specifies a set of desired controller parameters. The adaptive controller should converge to these parameter values even when the process is unknown. A regulator with this property is called *self-tuning*, since it automatically tunes the controller to the desired performance.

The *self-tuning regulator* (STR) is based on the idea of separating the estimation of unknown parameters from the design of the controller. The
The basic idea is illustrated in Fig. 5.1. The unknown parameters are estimated on-line, using a recursive estimation method. The estimated parameters are treated as if they are true; i.e., the uncertainties of the estimates are not considered. This is called the *certainty equivalence principle*. Many different estimation schemes can be used, such as stochastic approximation, least squares, extended and generalized least squares, instrumental variable, and maximum likelihood. Recursive estimation methods were thoroughly discussed in Chapter 3. The block “Design” in Fig. 5.1 represents an on-line solution to the design problem for a system with known parameters. This is called the *underlying design problem*. Examples of design methods that can be used are minimum variance, linear quadratic, pole placement, and model-following. The design method is chosen depending on the specifications of the closed loop system. Different combinations of estimation methods and design methods lead to regulators with different properties. The goal of this chapter is to give the basic ideas and properties of some classes of self-tuning regulators. Self-tuning regulators were originally derived for sampled data systems, but continuous-time and hybrid algorithms have also been developed.

In this chapter it is assumed that the process is described by the single input, single output (SISO) system

\[ A(q)y(t) = B(q)u(t) + C(q)e(t) \]  

(5.1)

where \( y \) is the output and \( u \) is the input of the process, and \( \{e(t)\} \) is a sequence of independent, equally distributed Gaussian variables. \( A, B, \) and \( C \) are polynomials in the forward shift operator \( q \). It is assumed that \( \deg A = \deg C = n \) and that \( \deg A - \deg B = d_0 \). It is sometimes convenient to write the process model in the delay operator \( q^{-1} \). This
can be done by introducing the reciprocal polynomial

\[ A^*(z) = z^n A(z^{-1}) \]

where \( n = \text{deg } A \). The model (5.1) is then

\[ A^*(q^{-1})y(t) = B^*(q^{-1})u(t - d_0) + C^*(q^{-1})e(t) \]

Self-tuning regulators are based on the idea of estimating some parameters of the process. The most straightforward approach is to estimate the parameters of the transfer function of the process and the disturbances. This gives an indirect adaptive algorithm (compare Section 1.2). The regulator parameters are not updated directly, but rather indirectly via the estimation of the process model. This type of adaptive controller, based on least-squares estimation and deadbeat control, was first described by Kalman in 1958. No analysis was 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. A similar controller, based on least-squares estimation and minimum-variance control, in which the uncertainties of the estimates were considered, was published by Wieslander and Wittenmark in 1971.

It is often possible to reparameterize the model in the regulator parameters, such that the regulator parameters can be estimated directly. That is, a direct adaptive algorithm is obtained (compare the discussion of the direct MRAS in Chapter 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 regulator parameters has been called implicit self-tuning control. 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 related to the process and/or the specifications of the closed-loop system.

Analysis of the asymptotic properties of a direct self-tuning regulator was first given in 1973 by Åström and Wittenmark who also coined the phrase “self-tuning.” Their regulator was based on least-squares estimation and minimum-variance control. Similar self-tuning regulators were developed by Peterka and Clarke and Gawthrop in the early 1970s. These works stimulated extensive research into different types of self-tuning regulators.

Indirect self-tuning algorithms are discussed in Section 5.2. Direct self-tuning regulators and their asymptotic properties are treated in Section 5.3. Historically, MRAS and STR have been treated as separate kinds of adaptive controllers. The main reason has perhaps been that the MRAS
was first developed for continuous-time systems and model-following, while
the STR was developed for discrete-time stochastic systems and loss func-
tion minimization. Today the treatment of MRAS and STR can be unified,
as discussed in Section 5.4. Linear quadratic Gaussian (LQG) self-tuners
are treated in Section 5.5. Adaptive predictive control is discussed in Sec-
tion 5.6, and the use of a priori knowledge in self-tuning regulators is
covered in Section 5.7.

5.2 Indirect Self-tuning Regulators

In this section it is assumed that the process is described by the model of
Eq. (5.1). The most straightforward way to make a self-tuning regulator
along the lines indicated in Section 5.1 is to estimate the parameters of
the polynomials $A$, $B$, and $C$. The estimates are then used in the design
of the regulator.

The deterministic case (i.e., $\epsilon(t) = 0$) is first considered. Several of
the recursive estimation methods outlined in Chapter 3 can be used to
estimate the coefficients of the $A$ and $B$ polynomials. Assume that

$$
\theta^T = [b_0 \ b_1 \ldots b_m \ a_1 \ldots a_n]
$$

$$
\varphi^T(t - 1) = [u(t - d_0) \ldots u(t - d_0 - m) - y(t - 1) \ldots - y(t - n)]
$$

where $n - m = d_0$. Then the least-squares estimator with exponential
forgetting (compare Eq. (3.16)) is given by:

$$
\hat{\theta}(t) = \hat{\theta}(t - 1) + K(t)\epsilon(t) \tag{5.2}
$$

$$
\epsilon(t) = y(t) - \varphi^T(t - 1)\hat{\theta}(t - 1) \tag{5.3}
$$

$$
K(t) = P(t - 1)\varphi(t - 1) (\lambda + \varphi^T(t - 1)P(t - 1)\varphi(t - 1))^{-1} \tag{5.4}
$$

$$
P(t) = (I - K(t)\varphi^T(t - 1)) P(t - 1) / \lambda \tag{5.5}
$$

In the stochastic case the least-squares method gives biased estimates if
$C(q) \neq q^n$. Methods such as recursive maximum likelihood or generalized
least squares must then be used instead.

Convergence

If the input signal to the process is sufficiently exciting and the structure
of the estimated model is appropriate, the estimates will converge to the
true values if the closed-loop system is stable. Conditions for convergence
for the different estimation methods are of great importance.

In both the deterministic and stochastic cases it is possible to give
conditions for convergence, which are related to the properties of the
input signal, to the process, and to the noise acting on the system. The control signal, \( u(t) \), is generated through feedback. This will complicate the analysis, since it is also necessary to require that the closed-loop system be stable. Exactly as for the MRAS, it is necessary to distinguish between parameter convergence and performance convergence. The convergence analysis is further discussed in Chapter 6.

The Underlying Design Problem for Known Systems

Various design methods can be used in self-tuning regulators, depending on the specifications of the closed-loop system. Most of the design methods can be interpreted as pole placement design. Model-following and pole placement are discussed in Section 4.2 and in Appendix A. For easy reference, the basic equations are repeated. In the discrete-time case the forward shift operator \( q \) is used as an argument in the polynomials.

Let the process model be Eq. (5.1) and let the desired closed-loop response be specified by

\[
A_m(q)y(t) = B_m(q)u_c(t)
\]  

(5.6)

The controller is

\[
R(q)u(t) = T(q)u_c(t) - S(q)y(t)
\]  

(5.7)

where \( R_1 \) and \( S \) are the solution to the Diophantine equation

\[
AR_1 + B^- S = A_o A_m
\]  

(5.8)

where

\[
B = B^+ B^-
\]  

(5.9)

\[
B_m = B^- B'_m
\]

(5.10)

\[
T = A_o B'_m
\]

(5.11)

\[
R = B^+ R_1
\]

A number of conditions must be satisfied to ensure that the controller is causal. (See Appendix A.) The equations above are the basis for many different design problems.

A Prototype for an Indirect Self-tuner

An indirect self-tuner based on the pole placement design can be expressed as in the following algorithm.

Algorithm 5.1—Indirect self-tuning regulator

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 $A$, $B$, and $C$ in Eq. (5.1) recursively using the least-squares method of Eqs. (5.2)–(5.5) or a method that also gives the $C$ polynomial.

Step 2: Replace $A$, $B$, and $C$ with the estimates obtained in Step 1 and solve Eq. (5.8) to obtain $R_1$ and $S$. Calculate $R$ by Eq. (5.11) and $T$ by Eq. (5.10).

Step 3: Calculate the control signal from Eq. (5.7).

Repeat Steps 1, 2, and 3 at each sampling period.

Among the potential problems with the algorithm are:

- The degrees of the polynomials in Eq. (5.1), or at least upper bounds of the degrees, must be known.
- Common factors of the estimated $A$ and $B$ polynomials make it impossible to solve Eq. (5.8).
- Stability of the closed-loop system must be guaranteed.
- The signals should be persistent exciting to ensure parameter convergence.

Examples

The properties of indirect self-tuning regulators are illustrated with two examples.

Example 5.1—Deterministic indirect self-tuning regulator

Let the process to be controlled be described by the transfer function

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

This can be regarded as a normalized model for a motor. The pulse transfer operator for the sampling period $h = 0.5$ s is

$$H(q) = \frac{B}{A} = \frac{0.107q + 0.090}{q^2 - 1.61q + 0.61} = \frac{0.107(q + 0.84)}{(q - 1)(q - 0.61)}$$

The sampled data system has a zero $z = -0.84$, which is inside the unit circle but poorly damped. Assume that the desired closed-loop system is

$$\frac{B_m}{A_m} = \frac{0.18}{q^2 - 1.32q + 0.50}$$

This corresponds to a natural frequency of 1 rad/s and a damping $\zeta = 0.7$. Finally, it is assumed that the observer polynomial is

$$A_o = (q - 0.5)^2$$
Figure 5.2  Output and input when using an indirect self-tuning regulator to control the system in Example 5.1. Notice the “ringing” in the control signal due to cancellation of the zero $z = -0.84$.

Figure 5.2 shows the output and the control signal of the process when an indirect self-tuning regulator is used with least-squares estimation and when the process zero $z = -0.84$ is canceled. Figure 5.3 shows that the estimates of the process parameters rapidly converge to the true model parameters. There is a severe oscillation in the control signal due to the cancellation of the zero; this oscillation is a consequence of a bad choice of the underlying design problem, not merely because self-tuning is used. The oscillation in the control signal is avoided by changing the design such that no process zeros are canceled (i.e., by requiring that $B_m = B$). Figure 5.4 shows the same simulation as Fig. 5.2 for the modified design, in which the process zero is not canceled. The behavior of the closed-loop system is now satisfactory.

Example 5.2—Stochastic indirect self-tuning regulator
Let the process be described by the difference equation.

$$y(t) + ay(t - 1) = bu(t - 1) + c(t) + ce(t - 1)$$

with $a = -0.9$, $b = 3$, and $c = -0.3$. The underlying design problem is assumed to be minimum-variance control. The minimum-variance controller
Figure 5.3  Parameter estimates corresponding to the simulation in Fig. 5.2. The estimates converge to the true process parameters.

is given by

$$u(t) = -\frac{c - a}{b} y(t) = -0.2y(t)$$

This gives the closed-loop system

$$y(t) = e(t)$$

The recursive maximum-likelihood method is used to estimate the unknown parameters $a$, $b$, and $c$. The estimates are obtained from Eqs. (5.2)–(5.5), with

$$\theta^T = [b \ a \ c]$$

$$\varphi^T(t-1) = [u(t-1) \ -y(t-1) \ \varepsilon(t-1)]$$

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

The controller is

$$u(t) = -\hat{s}_0(t)y(t)$$

where

$$\hat{s}_0(t) = \frac{\hat{c}(t) - \hat{a}(t)}{\hat{b}(t)}$$
5.2 Indirect Self-tuning Regulators

![Figure 5.4](image)

Figure 5.4 Same as in Fig. 5.2 but without cancellation of the process zero.

Figure 5.5 shows the result of a simulation of the algorithm. Figure 5.6 shows the loss function

\[ V(t) = \sum_{i=1}^{t} y^2(i) \]

when the optimal minimum-variance controller and the indirect self-tuning regulator are used. The curve for the accumulated loss of the STR is almost a translation of the optimal curve. This means that the performance of the self-tuning regulator is almost optimal except for a short start-up transient. Finally, Fig. 5.7 shows the controller parameter \( \hat{s}_0(t) \).

The figures show 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.

\[ \square \]

Summary

The indirect self-tuning algorithms are straightforward applications of the idea of self-tuning. They can be applied to a wide range of control design and parameter estimation methods. There are three main difficulties with the method. The stability analysis is complicated because the regulator parameters depend on the estimated parameters in a complicated way. Usually a set of linear equations in the controller parameters has
to be solved. The map from process parameters to regulator parameters may have singular points. This happens in design methods based on pole placement, for example, if the estimated process models have poles and zeros that coincide. Common poles and zeros have to be eliminated before the pole placement problem can be solved. Stability analysis has therefore been carried out in only a few cases. To ensure that the parameters converge to the correct values, it is necessary that the model structure be correct and that the process input be persistently exciting.

**Figure 5.6** The loss function when a self-tuning regulator and the optimal minimum-variance controller are used on the system in Example 5.2.
5.3 Direct Self-tuning Regulators

The design calculations for the algorithms discussed in the previous section may be time-consuming, and the stability properties may be difficult to analyze. It is possible to derive other algorithms in which the design calculations are simplified considerably. The idea is to use the specifications, in terms of the desired locations of the poles and the zeros, to rewrite the process model such that the design step is trivial. This leads to a reparameterization of the model.

Multiply the Diophantine equation (Eq. 5.8) by $y(t)$ and use the model of Eq. (5.1). Then

$$A_o A_m y(t) = R_1 A y(t) + B^- S y(t)$$

$$= R_1 B u(t) + B^- S y(t) + R_1 C e(t)$$

$$= B^- (R u(t) + S y(t)) + R_1 C e(t)$$

(5.12)

Notice that Eq. (5.12) can be considered as a process model that is parameterized in $B^-$, $R$, and $S$. Estimation of these parameters gives the regulator polynomials $R$ and $S$ directly. Together with Eq. (5.10), the control signal is computed from Eq. (5.7). One problem is that the model of Eq. (5.12) is nonlinear in the parameters unless $B^-$ is a constant. Compare with Eq. (4.49).

Another way to achieve the parameterization is to write the model of Eq. (5.12) as

$$A_o A_m y = \bar{R} u + \bar{S} y + R_1 C e$$

(5.13)

where

$$\bar{R} = B^- R$$

and

$$\bar{S} = B^- S$$
Notice that the polynomial $R$ of Eq. (5.12) is monic but $\tilde{R}$ of Eq. (5.13) is not monic. The polynomials $\tilde{R}$ and $\tilde{S}$ have a common factor, which represents poorly damped zeros. This factor should be canceled before calculating the control law. The following control algorithm is then obtained.

**Algorithm 5.2—Direct self-tuning regulator**

*Step 1:* Estimate the coefficients of the polynomials $\tilde{R}$ and $\tilde{S}$ in the model of Eq. (5.13).

*Step 2:* Cancel possible common factors in $\tilde{R}$ and $\tilde{S}$ to obtain $R$ and $S$. Those are the ones obtained in Step 2.

*Step 3:* Calculate the control signal from Eq. (5.7) where $R$ and $S$ are obtained in Step 2.

Repeat Steps 1, 2, and 3 at each sampling period.

This algorithm avoids the nonlinear estimation problem, but there are more parameters to estimate than when Eq. (5.12) is used, because the parameters of the polynomial $B^-$ are estimated twice. Step 2 may, however, be difficult.

Because of the difficulties of estimating the parameters in Eq. (5.12), it is of interest to consider special, that lead to simpler calculations. One such case is obtained when $B^-$ is a constant. Assuming that all zeros can be canceled (i.e., $B^- = b_0$),

$$A_oA_my(t) = b_0(Ru(t) + Sy(t)) + R_1Ce(t)$$

(5.14)

Let the desired response be described by

$$A_my_m(t) = b_0Tu_c(t)$$

where $\text{deg}A = n$ and $A_o$ divides $T$. The error $\varepsilon(t) = y(t) - y_m(t)$ is then given by

$$\varepsilon(t) = \frac{b_0}{A_oA_m}(Ru(t) + Sy(t) - Tu_c(t)) + \frac{R_1C}{A_oA_m}e(t)$$

Different cases can now be considered. First assume that $\varepsilon$ is zero. The observer polynomial can then be chosen freely, as in Chapter 4. When using continuous-time models as in Chapter 4, it is necessary to assume that $b_0/(A_oA_m)$ is SPR to get a stable MRAS. A similar condition is also necessary for discrete-time models. The model can, however, be rewritten as

$$\varepsilon(t) = b_0\left(R\frac{u(t)}{A_oA_m} + S\frac{y(t)}{A_oA_m} - T\frac{u_c(t)}{A_oA_m}\right)$$

$$= b_0\left(R^*u_f(t-d_0) + S^*y_f(t-d_0) - T^*u_{cf}(t-d_0)\right)$$
where
\[ u_f(t) = \frac{1}{A_o^*(q^{-1})A_m^*(q^{-1})} u(t) \]
\[ y_f(t) = \frac{1}{A_o^*(q^{-1})A_m^*(q^{-1})} y(t) \]
\[ u_c(t) = \frac{1}{A_o^*(q^{-1})A_m^*(q^{-1})} u_c(t) \]

This corresponds to the case \( P = Q = A_oA_m \) in Section 4.5. The convergence properties will now depend on knowledge of the sign of \( b_0 \). This shows again the close relation between MRAS and STR. The following direct self-tuning algorithm is now obtained.

**Algorithm 5.3—Deterministic direct self-tuning algorithm**

**Data:** Given a lower bound on the delay \( d_0 \) and the sign of \( b_0 \). The specifications are given in form of a desired closed-loop pulse transfer operator \( b_0/A_m^* \) and a desired observer polynomial \( A_o \).

**Step 1:** Estimate the coefficients of the polynomials \( R^* \), \( S^* \), and \( T^* \) in the model of Eq. (5.14), using a recursive estimation method.

**Step 2:** Calculate the control signal from
\[ R^*u(t) = -S^*y(t) + T^*u_c(t) \]

Repeat Steps 1 and 2 at each sampling period.

This algorithm corresponds to the model-reference adaptive controller in Section 4.5. Notice that the algorithm requires that \( b_0 \) is known. If \( b_0 \) is unknown it can be estimated. This is done formally by replacing Eq. (5.14) by
\[ A_oA_my(t) = Ru(t) + Sy(t) + R_1Ce(t) \]
where \( R \) now is not monic.

**Minimum-variance and Moving-average Controllers**

Different stochastic control algorithms for the system given by Eq. (5.1) will now be derived. Assume first that the model is known, that \( e \) is a stochastic process, and that \( u_c = 0 \). The optimal observer polynomial for the model of Eq. (5.1) is \( A_o = C \). The design criterion is minimum-variance control or moving-average control.

If the process is minimum-phase, the minimum-variance regulator is given by
\[ R^*(q^{-1})u(t) = -S^*(q^{-1})y(t) \]
where the polynomials $R^*$ and $S^*$ are the minimum degree solution to the Diophantine equation
\[ A^*(q^{-1})R^*(q^{-1}) + q^{-d_0}B^*(q^{-1})S^*(q^{-1}) = B^*(q^{-1})C^*(q^{-1}) \] (5.16)

with $d_0 = \deg A - \deg B$. The minimum-variance controller corresponds to a desired model with a delay of $d_0$ steps, $A_m^* = 1$. It follows from Eq. (5.16) that $B^*$ divides $R^*$, i.e.,
\[ R^* = R_1^*B^* \]

where $\deg R_1 = d_0 - 1$. Equation (5.16) can then be written as
\[ A^* R_1^* + q^{-d_0}S^* = C^* \] (5.17)

Letting Eq. (5.17) operate on $y(t)$ gives
\[
C^*y(t) = A^* R_1^* y(t) + S^* y(t - d_0) \\
= B^* R_1^* u(t - d_0) + S^* y(t - d_0) + R_1^* C^* e(t) \\
= R^* u(t - d_0) + S^* y(t - d_0) + R_1^* C^* e(t)
\]

This equation can also be written as
\[ y(t + d_0) = \frac{1}{C^*} (R^* u(t) + S^* y(t)) + R_1^* e(t + d_0) \] (5.18)

With the controller of Eq. (5.15) the output of the closed loop system becomes
\[ y(t) = R_1^* (q^{-1}) e(t) \]

The output is thus a moving average of order $d_0 - 1$. In Åström (1970) it is shown that the regulator minimizes the output variance. An interesting feature of the controller is that the output becomes a moving average of order $d_0 - 1$. Notice that the integer $d_0$ can be interpreted as the number of samples it takes for a change in the input to be noticeable in the output.

A drawback with the minimum-variance controller is that all process zeros are canceled. This means that there will be difficulties if the $B$ polynomial has zeros outside the unit disc or inside it but close to the unit circle. These difficulties are avoided in the moving-average controller. This controller gives an output that is a moving average of order greater than $d_0 - 1$. The controller is derived as follows. Factor the polynomial $B$ in $B^+$ and $B^-$ as in Eq. (5.9), where $B^+$ corresponds to well-damped zeros. Determine $R^*$ and $S^*$ from
\[ A^* R^* + q^{-d_0} B^* S^* = B^{++} C^* \]
5.3 Direct Self-tuning Regulators

Proceeding as was done when deriving Eq. (5.18) gives

\[ y(t + d) = \frac{1}{C^*} (R^* u(t) + S^* y(t)) + R_1^* e(t + d) \]  

(5.19)

where \( R^* = R_1^* B^{**} \). With the feedback law of Eq. (5.15), the controlled output becomes

\[ y(t) = R_1^* (q^{-1}) e(t) \]

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

\[ d = \deg A = n \]

Both the minimum-variance control law and the moving-average control law lead to the equivalent models of Eq. (5.18) and Eq. (5.19), respectively. The only difference is in the value of the integer \( d \), which controls the number of process zeros that are canceled. With \( d = d_0 = \deg A - \deg B \), all process zeros are canceled. With \( d = \deg A \), no process zeros are canceled. Filtering with \( A_0^* \), as in Eq. (5.14), can also be introduced into the model of Eq. (5.19). This gives

\[ y(t + d) = \frac{A_0^*}{C^*} (R^* u_f(t) + S^* y_f(t)) + R_1^* e(t + d) \]

(5.20)

The case when \( B^+ \) contains all the stable process zeros corresponds to the suboptimal minimum-variance controller in Åström (1970).

Self-tuning Minimum-variance and Moving-average Controllers

Using the reparametrized model of Eq. (5.20) it is possible to make a self-tuning moving-average controller.

**Algorithm 5.4—Basic direct self-tuning algorithm**

Data: Given the prediction horizon \( d \). Let \( k \) and \( l \) be the numbers of parameters in the \( R^* \) and \( S^* \) polynomials, respectively.
Step 1: Estimate the coefficients of the polynomials $R^*$ and $S^*$ of the model

$$y(t + d) = R^*(q^{-1})u_f(t) + S^*(q^{-1})y_f(t) + \varepsilon(t + d)$$  \hspace{1cm} (5.21)

where

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

$$S^*(q^{-1}) = s_0 + s_1 q^{-1} + \cdots + s_l q^{-l}$$

and

$$u_f(t) = \frac{1}{A^*_0(q^{-1})} u(t)$$

$$y_f(t) = \frac{1}{A^*_0(q^{-1})} y(t)$$

using Eqs. (5.2)—(5.5), with

$$\varepsilon(t) = y(t) - R^* u_f(t - d) - S^* y_f(t - d) = y(t) - \varphi^T(t - d) \hat{\theta}(t - 1)$$

$$\varphi^T(t) = \frac{1}{A^*_0(q^{-1})} \left[ u(t) \cdots u(t - k) y(t) \cdots y(t - l) \right]$$

$$\theta^T = [r_0 \ldots r_k \; s_0 \ldots s_l]$$

Step 2: Calculate the control signal from

$$R^*(q^{-1})u(t) = -S^*(q^{-1})y(t)$$  \hspace{1cm} (5.22)

with $R^*$ and $S^*$ replaced by their estimates obtained in Step 1.

Repeat Steps 1 and 2 at each sampling period.

Remark. 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

$$\varepsilon(t) = y(t) - r_0 u_f(t - d) - \varphi^T(t - d) \hat{\theta}(t - 1)$$

$$\varphi^T(t) = \frac{1}{A^*_0(q^{-1})} \left[ r_0 u(t - 1) \ldots r_0 u(t - k) \ y(t) \cdots y(t - l) \right]$$

$$\theta^T = [r'_1 \ldots r'_k \; s_0 \ldots s_l]$$
Asymptotic Properties

The models of Eqs. (5.18) and (5.19) can be interpreted as reparameterizations of the process model of Eq. (5.1) in terms of the controller parameters. They are identical to the model of Eq. (5.21) in Algorithm 5.4 if the polynomial \( A_o \) is chosen as \( C \). The regression vector is then uncorrelated with the errors and the least-squares estimate can be expected to converge to the true parameters. The surprising result is that the algorithm also self-tunes to the correct controller when \( C \neq A_o \). The following result shows that the correct regulator parameters are equilibrium values for Algorithm 5.4 also when \( A_o \neq C \). A more detailed analysis of stability and convergence is found in Chapter 6.

**Theorem 5.1—Asymptotic properties**

Let Algorithm 5.4 with \( A_o^* = 1 \) be used with a least-squares estimator. The parameter \( b_0 = r_0 \) can either be 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

\[
\begin{align*}
\bar{y}(t + \tau)y(t) &= 0 \quad \tau = d, d + 1, \ldots, d + l \\
\bar{y}(t + \tau)u(t) &= 0 \quad \tau = d, d + 1, \ldots, d + k
\end{align*}
\]

(5.23)

where the bar indicates time averages. Also, \( k \) and \( l \) are the number of estimated parameters in the polynomials \( R^* \) and \( S^* \), respectively.

**Proof:** The model of Eq. (5.21) can be written as

\[
y(k + d) = \varphi^T(k)\theta + \varepsilon(k + d)
\]

and the control law becomes

\[
\varphi^T(k)\hat{\theta}(k + d) = 0
\]

At an equilibrium, the estimated parameters \( \hat{\theta} \) are constant. Furthermore, they satisfy the normal equations (see Eq. (3.4)), which in this case are written as

\[
\frac{1}{t} \sum_{k=1}^{t} \varphi(k)y(k + d) = \frac{1}{t} \sum_{k=1}^{t} \varphi(k)\varphi^T(k)\hat{\theta}(t + d)
\]

Using the control law it follows 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 vectors are bounded, the right-hand side goes to zero. Equation (5.23) now follows from \( A_o^* = 1 \) and the definition of the regression vector in Algorithm 5.4.

Stronger statements can be made if more is assumed about the system to be controlled.

**Theorem 5.2—Asymptotic properties 2**

Assume that Algorithm 5.4 with least-squares estimation is applied to Eq. (5.1). Further, assume that

\[
\min(k, l) \geq n - 1 \quad (5.24)
\]

If the asymptotic estimates of \( R \) and \( S \) are relative prime, the equilibrium solution is such that

\[
y(t + \tau)y(t) = 0 \quad \tau = d, d + 1, \ldots \quad (5.25)
\]

i.e., that the output is a moving-average process of order \( d - 1 \).

**Proof:** The closed-loop system is described by

\[
\begin{align*}
R^*u(t) & = -S^*y(t) \\
A^*y(t) & = B^*u(t - d_0) + C^*e(t)
\end{align*}
\]

Hence

\[
\begin{align*}
(A^*R^* + q^{-d_0}B^*S^*)y & = R^*C^*e \\
(A^*R^* + q^{-d_0}B^*S^*)u & = -S^*C^*e
\end{align*}
\]

Introduce the signal \( w \) defined by

\[
(A^*R^* + q^{-d_0}B^*S^*)w = C^*e \quad (5.26)
\]

Hence

\[
y = R^*w \quad \text{and} \quad u = -S^*w \quad (5.27)
\]

The condition of Eq. (5.23) then implies

\[
\begin{align*}
Rw(t + \tau)y(t) & = 0 \quad \tau = d, d + 1, \ldots, d + l \\
Sw(t + \tau)y(t) & = 0 \quad \tau = d, d + 1, \ldots, d + k
\end{align*}
\]

Introducing

\[
C_{wy}(\tau) = \frac{w(t + \tau)y(t)}{
}
the above equations can be written as

\[
\begin{pmatrix}
  r_0 & r_1 & r_2 & \ldots & r_k & 0 & \ldots & 0 \\
  0 & r_0 & r_1 & & r_k & & \\
  \vdots & \ddots & \ddots & \ddots & \vdots \\
  0 & \ldots & 0 & r_0 & r_1 & r_2 & \ldots & r_k \\
  s_0 & s_1 & s_2 & \ldots & s_l & 0 & \ldots & 0 \\
  0 & s_0 & s_1 & s_2 & \ldots & s_l & \\
  \vdots & \ddots & \ddots & \ddots & \ddots \\
  0 & \ldots & 0 & s_0 & s_1 & s_2 & \ldots & s_l
\end{pmatrix}
\begin{pmatrix}
  C_{wy}(d + k + l) \\
  \vdots \\
  C_{wy}(d)
\end{pmatrix} = 0
\]

Since the matrix on the left is regular when \( R^* \) and \( S^* \) are relatively prime (compare Appendix A), it follows that

\[ C_{wy}(\tau) = 0 \quad \tau = d, d + 1, \ldots, d + k + l \]

The covariance function satisfies the equation

\[ F^*(q^{-1})C_{wy}(\tau) = 0 \quad \tau \geq 0 \]

The system of Eq. (5.26) has the order

\[ n + k = n + \max(k, l) \]

If

\[ k + l + 1 \geq n + \max(k, l) \]

or equivalently,

\[ \min(k, l) \geq n - 1 \]

it follows that

\[ C_{wy}(\tau) = 0 \quad \tau = d, d + 1, \ldots \]

It also follows from Eq. (5.27) that

\[ C_y(\tau) = 0 \quad \tau = d, d + 1, \ldots \]

which completes the proof.

Remark 1. Notice that the algorithm can be interpreted as if it attempts to drive the correlation of the output to zero starting at lag \( \tau = d \). It follows from Theorem 5.1 that the correlations at lags \( d, d + 1, \ldots, d + l \) will always be zero at equilibrium. If there are enough parameters in the regulator, the covariance of the output will be zero for all higher lags.
Notice that the condition of Eq. (5.25) is easily checked by monitoring the covariances of the output.

Remark 2. Notice that it is possible to influence cancellation of the process zeros simply by choosing the integer $d$. With $d = d_0$ a controller which cancels all zeros are obtained with $d = n$. The controller will not cancel any process zeros. □

Theorems 5.1 and 5.2 imply that if the estimates converge, and if there are sufficiently many parameters in the controller, then Algorithm 5.4 will converge to the moving-average controller.

Connection between MRAS and STR

Direct model-reference adaptive systems were discussed in Chapter 4. In Appendix A it is shown that model-following and pole placement are related. It will now be shown that the direct self-tuning pole placement controller in Algorithm 5.2 is equivalent to an MRAS. The deterministic case, when $B^-$ is a constant, is considered. The process model can be written as

$$y(t) = \varphi_f^T(t - d_0)\theta$$

In the direct algorithm the estimated parameters are equal to the regulator parameters. The least-squares method can be used for the estimation, and the residual $\varepsilon(t)$ can then be written as

$$\varepsilon(t) = y(t) - \hat{y}(t) = y(t) - \varphi_f^T(t - d_0)\hat{\theta}$$

(5.28)

The parameter update can be written as

$$\hat{\theta}(t) = \hat{\theta}(t - 1) + P(t)\varphi_f^T(t - d_0)\varepsilon(t)$$

(5.29)

Notice that it follows from Eq. (5.28) that

$$\varphi_f^T(t - d_0) = -\text{grad}_\theta \varepsilon(t)$$

The vector $\varphi_f^T(t - d_0)$ can be interpreted as the sensitivity derivative. The parameter update of Eq. (5.29) 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 residual $\varepsilon(t)$. Another difference is that the gain $\gamma$ in the MRAS is replaced by the matrix $P(t)$, which is given by Eq. (5.5). $P$ modifies the gradient direction and it gives an appropriate step length. Whereas the ordinary MIT rule can be viewed as a gradient algorithm to minimize $e^2$, Eq. (5.29) can be considered as a quasi-Newton method to minimize $\varepsilon^2$. In Section 4.5 it was shown that the residual $\varepsilon$ was the same
as the augmented error. It is interesting to see that the augmented error is obtained automatically in the STR formulation.

Notice that in the identification-based schemes such as self-tuning controllers we normally attempt to obtain a form similar to

\[ y(t) = \varphi_t^T \theta \]

With the model-reference approach it is usually only possible to get a model of the form

\[ y(t) = G(p) (\varphi_t^T \theta) \]

where \( G(p) \) is SPR.

**Examples**

The properties of the minimum-variance and moving-average self-tuners are illustrated with two examples.

**Example 5.3—Direct minimum-variance self-tuning regulator**

Consider the same process as in Example 5.2. The process model of Eq. (5.21) 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 using the least-squares method. The control law becomes

\[ u(t) = -\frac{s_0}{\hat{r}_0} y(t) \]

Figure 5.8 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. 5.9, which shows the loss function when the self-tuner and the optimal minimum-variance controller are used. Compare Figs. 5.6 and 5.7.

**Example 5.4—MA control of a non-minimum-phase system**

Consider an integrator with a time delay \( \tau \). 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

\[ C(q) = q(q + c) \quad |c| < 1 \]

The system is non-minimum-phase if \( \tau > h/2 \). This implies that the basic minimum-variance self-tuner can only be used if \( \tau < h/2 \). Let the regulator have the structure

\[ u(t) = -\hat{s}_0(t)y(t) - \hat{\tau}_1(t)u(t - 1) \]

Simulations of the system are shown in Fig. 5.10, when \( h = 1 \) and \( c = -0.8 \). The time delay is initially 0.4 and is increased to 0.6 at time \( t = 100 \). Figure 5.10(a) shows the results obtained with \( d = 1 \), the minimum-variance structure. The system is initially as well-behaved as can be expected. The parameters tend towards \( r_1 = 0.67 \) and \( s_0 = 0.33 \), which correspond to the minimum-variance controller. The system becomes unstable sometime after \( t = 100 \), when the process to be controlled has become non-minimum-phase. The reason it does not become unstable
exactly at $t = 100$ is that it takes a while for the regulator parameters to change. The control signal is bounded in the simulation, which explains why the signals do not grow exponentially. Figure 5.10(b) shows the results for the algorithm with $d = 2$. The moving-average controller is a stable equilibrium both for $\tau = 0.4$ 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 regulator 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 using the moving-average controller in this example is thus minor.
5.4 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 it is easy to implement. This has lead to a large interest in the algorithm. The algorithm can be explained as follows. Estimate a prediction model that allows prediction of the output \( d \) steps ahead. Use the prediction model to determine a control signal that brings the predicted value to a desired value. Notice also that there is a close similarity between the way the model is determined and how it is used in the control design.

The algorithm has been analyzed extensively. Various extensions have also been proposed. 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 will be demonstrated below.

Consider the model of Eq. (5.1) 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

\[
y_f(t + d_0) = R_1^*e(t + d_0) + \frac{1}{C^*Q^*} (S^*y_f(t) + B^*Q^*R_1^*u(t))
\]

Introducing

\[
y'_f(t) = \frac{1}{Q^*(q^{-1})} y_f(t) = \frac{1}{P^*(q^{-1})} y(t)
\]

gives the model

\[
y_f(t + d_0) = R_1^*e(t + d_0) + \frac{1}{C^*} (S^*y'_f(t) + B^*R_1^*u(t)) \quad (5.30)
\]
By analogy with Eq. (5.20), this model structure could be used together with Algorithm 5.4 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 5.5—Generalized direct self-tuning algorithm**

**Data:** Given the prediction horizon, \( d \), the order of the regulator, \( \text{deg} \, R^* \) and \( \text{deg} \, S^* \), the stable observer polynomial, \( A_o^* \), and the stable polynomials \( Q^* \) and \( P^* \). Define the filtered signals

\[
y_f(t) = \frac{Q^*}{P^*} y(t) \quad y'_f(t) = \frac{1}{P^*} y(t)
\]

**Step 1:** Estimate the coefficients of the polynomials \( R^* \) and \( S^* \) of the model

\[
y_f(t + d) = \frac{R^*}{A_o^*} u(t) + \frac{S^*}{A_o^*} y'_f(t) + \varepsilon(t + d)
\]

using the least-squares method.

**Step 2:** Calculate the control signal from

\[
u(t) = -\frac{S^*}{R^*} y'_f(t)
\]

with \( R^* \) and \( S^* \) replaced by their estimates obtained in Step 1. Repeat Steps 1 and 2 at each sampling period. \( \square \)

From Eq. (5.30) and Theorems 5.1 and 5.2 it follows that if the estimates converge, then the closed-loop system will be

\[
y_f(t) = R_1^* e(t)
\]

or

\[
y(t) = \frac{P^* R_1^*}{Q^*} e(t)
\]

where \( R_1^* \) is given by the identity (compare Eq. (5.17))

\[
C^* Q^* = A^* P^* R_1^* + q^{-d} B^{-*} S^*
\]

and the control signal is given by

\[
u(t) = -\frac{S^*}{R^*} y'_f(t) = -\frac{S^*}{R^* P^*} y(t)
\]

The closed-loop poles will thus be influenced by \( Q^* \), and additional zeros can be introduced through \( P^* \).
Algorithm 5.5 is essentially the same as Algorithm 5.4 applied to filtered signals. The filter $Q^*/P^*$ and the prediction horizon will determine the pulse transfer operator of the closed-loop system. The observer polynomial $A_o^*$ will determine the convergence properties. This difference can essentially be interpreted as if another observer polynomial were used. This will not influence the asymptotic properties as long as the filter and its inverse are stable.

Minimum-variance control usually results 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 of the control signal. Linear quadratic controllers are of this type; a minor drawback with LQ self-tuning regulators is the computational burden. One way to simplify the problems is to use a one-step-ahead loss function of the form

$$E \left\{ (P^*(q^{-1})y(t + d_0))^2 + (Q^*(q^{-1})u(t))^2 \mid Y_t \right\}$$

where $Y_t$ is the data available at time $t$. The resulting controller, sometimes called a generalized minimum-variance controller, was introduced in the self-tuning context by Clarke and Gawthrop. 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

$$E \left\{ y^2(t + d_0) + \rho u^2(t) \mid Y_t \right\} \quad (5.35)$$

Assume that the process is governed by Eq. (5.1). Using the representation of the process dynamics given by Eq. (5.18) it can be shown that the control law that minimizes Eq. (5.35) is

$$(R^* + \frac{\rho}{r_o} C^*) u(t) = -S^* y(t) \quad (5.36)$$

where

$$R^* = R_1^* B^*$$

and $R^*$ and $S^*$ are given by Eq. (5.16).

Using the same idea as above, it is possible to construct a new system, which has Eq. (5.36) as its minimum-variance regulator. Augment the original system with a parallel connection with the pulse transfer operator $\rho q^{-d_0}$ (see Fig. 5.11). This is in fact a standard technique to obtain an equivalent controller with a bounded gain. (See Problem 5.2.) The input-output relation of the augmented system is

$$A^* y_a(t) = \left( B^* + \frac{\rho}{r_o} A^* \right) u(t - d_0) + C^* e(t)$$
5.4 Unification of Direct Self-tuning Regulators

Figure 5.11 Equivalent systems.

The minimum-variance control law for this system is given by

$$R_1^*(B^* + \frac{\rho}{r_o} A^*)u(t) = -S^* y_a(t)$$  \hspace{1cm} (5.37)

where $R_1^*$ and $S^*$ satisfy Eq. (5.16). It follows from Fig. 5.11 that

$$y_a(t) = y(t) + \rho q^{-d_0} u(t)$$

Then Eq. (5.37) can be written as

$$\left( R_1^* B^* + \frac{\rho}{r_o} A^* R_1^* \right) u(t) = -S^* \left( y(t) + \frac{\rho}{r_o} q^{-d_0} u(t) \right)$$

or

$$\left( R_1^* B^* + \frac{\rho}{r_o} (A^* R_1^* + q^{-d_0} S^*) \right) u(t) = -S^* y(t)$$

Equation (5.16) gives

$$(R_1^* B^* + \frac{\rho}{r_o} C^*) u(t) = -S^* y(t)$$

which is identical to Eq. (5.36). Notice that with the control law of Eq. (5.37) the canceled factor is not $B^*$ but $B^* + \frac{\rho}{r_o} A^*$. This implies that problems can be expected when the system is non-minimum-phase and close to the stability boundary.

In the Clarke and Gawthrop algorithm it is assumed that $C^*(q^{-1}) = 1$. Their 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 $\rho$ of the parallel path.

The analysis above shows that the Algorithm 5.5 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. (5.7) 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 $v(t)$ can, for instance, be the reference value. The polynomials $R^*$, $S^*$, and $T^*$ are estimated in the usual way, and the control law is chosen as

$$u(t) = -\frac{S^*}{R^*} y(t) - \frac{T^*}{R^*} v(t)$$

Self-tuning feedforward control has been used successfully in many industrial applications.

**Examples**

The behavior of Algorithm 5.5 is illustrated through two examples.

**Example 5.5—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 5.2 and 5.3. Let the filter be

$$\frac{Q^*}{P^*} = \frac{1 + q_1 q^{-1}}{1 + p_1 q^{-1}}$$
Figure 5.12  The output and input variance as functions of $q_1$ of the system in Example 5.5 when $p_1 = -0.3, 0$, and 0.3. Three different cases are marked in the curves: (a) $p_1 = q_1 = 0$; (b) $p_1 = 0, q_1 = -0.3$; (c) $p_1 = -0.3, q_1 = -0.9$.

The identity of Eq. (5.33) gives the solution

$$s_0 = c + q_1 - a - p_1$$
$$s_1 = cq_1 - ap_1$$

The control law is given by Eq. (5.34), with

$$R_1^*P^*B^{**} = b(1 + p_1q^{-1})$$

The closed-loop system is given by

$$y(t) = \frac{1 + p_1q^{-1}}{1 + q_1q^{-1}} e(t)$$
$$u(t) = -\frac{s_0 + s_1q^{-1}}{b(1 + q_1q^{-1})} e(t)$$

There are many different ways to choose the filter $Q^*/P^*$. In principle it should be a lead network. This implies that the closed-loop system given by Eq. (5.32) will be low-pass filtered. Fig. 5.12 shows how the output and input variances change with $q_1$ for some values of $p_1$. Case
Figure 5.13 Simulation of the generalized self-tuning algorithm on the system in Example 5.5 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).

(a) in Fig. 5.12 corresponds to minimum-variance control. In case (b) the output variance is increased by 10%, while the input variance is reduced by about 60% compared with the minimum-variance case. In case (c) the input variance is zero; i.e., the system is open loop. Figure 5.13 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 marked in Fig. 5.12.

Example 5.6—The Clarke and Gawthrop self-tuning controller
The self-tuning controller that minimizes Eq. (5.35) will now be used to control the same system as in the previous example.

The controller in Eq. (5.36), with $R^*$ and $S^*$ given by Eq. (5.16), is

$$u(t) = -\frac{c - a}{b + \rho(1 + aq^{-1})} y_a(t)$$

Figure 5.14 shows the output variance as function of the input variance for different values of $\rho$. The curve has the same gross behavior as shown in Fig. 5.12. The parameter $\rho$ may, however, be easier to choose than the filter in the previous example. Figure 5.15 shows the accumulated losses of the output and the input for different values of $\rho$ when the self-tuner
in Algorithm 5.4 is used on the augmented system shown in Fig. 5.11. Compare Fig. 5.13.

Figure 5.15 Simulation of the Clarke-Gawthrop self-tuning controller on the system in Example 5.5 when (a) $\rho = 0$ (minimum-variance control); (b) $\rho = 4$; (c) $\rho = 100$ ("almost" open loop control).
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 5.5, is very flexible. 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 is also influenced by the filter $Q^*/P^*$.

For simplicity, Algorithm 5.5 has been derived for the regulator case, in which the reference value is equal to zero. It is easy to modify the algorithm such that the output follows a reference trajectory; some ideas are suggested in the Problems section in this chapter and in Chapter 11.

5.5 Linear Quadratic STRs

The linear quadratic design procedure can also be used as the design method in a self-tuning regulator. Consider the process model

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \quad (5.38)$$

and the steady-state loss function

$$J_{yu} = E \left\{ (y(t) - y_m(t))^2 + \rho u^2(t) \right\} \quad (5.39)$$

The optimal feedback law that minimizes Eq. (5.39) for the system of Eq. (5.38) is given by the following theorem:

**Theorem 5.3—LQG control**

Consider the system in Eq. (5.38). Let the polynomials $A(q)$ and $C(q)$ have degree $n$. Assume that $C(q)$ is monic and has all its zeros inside the unit disc, and assume that there is no polynomial that divides $A(q)$, $B(q)$, and $C(q)$. Let $A_2(q)$ be the greatest common divisor of $A(q)$ and $B(q)$ and let $A_2^-(q)$ of degree $m$ be the factor of $A_2(q)$ that has all its zeros outside the unit disc or on the unit circle.

The admissible control law that minimizes Eq. (5.39) with $\rho > 0$ is then given by

$$R(q)u(t) = -S(q)y(t) + T(q)y_m(t) \quad (5.40)$$
where the polynomials $R$ and $S$ satisfy the Diophantine equation

$$A(q)R(q) + B(q)S(q) = P(q)C(q) \quad (5.41)$$

with the additional constraints

$$\deg R(q) = \deg S(q) = n + m$$

$$A_2(q) \text{ divides } R(q)$$

$$\deg S^*(q) < n$$

The polynomial $P(q)$ is given by

$$P(q) = q^mP_1(q)A_2(q) \quad (5.42)$$

where

$$P_1(q)P_1(q^{-1}) = \rho A_1(q)A_1(q^{-1}) + B_1(q)B_1(q^{-1}) \quad (5.43)$$

and

$$A_1(q) = A(q)/A_2(q)$$

$$B_1(q) = B(q)/A_2(q)$$

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 (1984). To solve the design problem it is necessary to solve the spectral factorization problem of Eq. (5.43) and to solve the Diophantine equation Eq. (5.41). The solution to the LQG problem given by Theorem 5.3 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. (5.38) can be written in state space form as

$$x(t + 1) = \bar{A}x(t) + \bar{B}u(t) + \bar{K}e(t)$$

$$y(t) = \bar{C}x(t) + c(t) \quad (5.44)$$
where the matrices $\tilde{A}$, $\tilde{B}$, $\tilde{C}$, and $\tilde{K}$ are given by

$$
\tilde{A} = \begin{pmatrix}
-a_1 & 1 & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
-a_{n-1} & 0 & \cdots & 1 \\
-a_n & 0 & \cdots & 0
\end{pmatrix}
$$

$$
\tilde{B} = \begin{bmatrix}
0 \\ \vdots \\ 0 \bar{b}_0 \\ \vdots \\ 0 \bar{b}_m
\end{bmatrix}^T
$$

$$
\tilde{C} = \begin{bmatrix}
1 \\ 0 \\ \vdots \\ 0
\end{bmatrix}
$$

$$
\tilde{K} = \begin{bmatrix}
\bar{c}_1 - a_1 \\ \vdots \\ \bar{c}_n - a_n
\end{bmatrix}^T
$$

where $m = n - d_0$. The model in Eq. (5.44) is called the innovation model, and $\tilde{K}$ is the optimal steady-state gain in the Kalman filter; i.e., $\hat{x}(t+1|t) = x(t+1)$. It is also possible to derive the filter for $\hat{x}(t|t)$, which is given by

$$
\hat{x}(t|t) = (qI - \tilde{A} + \tilde{K}\tilde{C})^{-1}(\tilde{B}u(t) + \tilde{K}y(t))
$$

Notice that $\det(qI - \tilde{A} + \tilde{K}\tilde{C}) = C(q)$. That is, the optimal observer polynomial is equal to $C(q)$.

Introduce the loss function

$$
J_x = \mathbb{E}\left\{\sum_{t=1}^{N} x^T(t)Q_1x(t) + \rho u^2(t) + x^T(N)Q_0x(N)\right\}
$$

(5.45)

The optimal controller is given by

$$
u(t) = -L(t)\hat{x}(t|t)
$$

(5.46)

where $L(t)$ is a time-varying feedback gain given through a Riccati equation.

$$
S(t) = (\tilde{A} - \tilde{B}L(t))^TS(t+1)(\tilde{A} - \tilde{B}L(t)) + Q_1 + \rho L^T(t)L(t)
$$

$$
L(t) = (\rho + \tilde{B}^TS(t+1)\tilde{B})^{-1}\tilde{B}^TS(t+1)\tilde{A}
$$

(5.47)

The limiting controller

$$
\tilde{L} = \lim_{t \to \infty} L(t)
$$

is such that the closed-loop characteristic equation is

$$
P(q) = \det(q - \tilde{A} + \tilde{B}\tilde{L}) = 0
$$
where \( P(q) \) is the same as in Eq. (5.42).

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. (5.38). This can be done using the recursive maximum-likelihood method or the extended least-squares method. This leads to the following algorithm.

**Algorithm 5.6—Indirect STR based on spectral factorization**

**Data:** Given specifications in the form of the parameter \( \rho \) in the loss function of Eq. (5.39) and the order of the system.

**Step 1:** Estimate the coefficients of the polynomials \( A, B, \) and \( C \) in Eq. (5.38).

**Step 2:** Replace \( A, B, \) and \( C \) with the estimates obtained in Step 1 and solve the spectral factorization problem of Eq. (5.43) to obtain \( P(q) \).

**Step 3:** Solve the Diophantine equation of Eq. (5.41).

**Step 4:** Calculate the control signal from Eq. (5.40).

Repeat Steps 1, 2, 3, and 4 at each sampling period.

The state space formulation gives the following algorithm:

**Algorithm 5.7—Indirect STR based on Riccati equation**

**Data:** Given specifications in the form of the parameters \( Q_0, Q_1, \) and \( \rho \) in the loss function of Eq. (5.46) and the order of the system.

**Step 1:** Estimate the coefficients of the polynomials \( A, B, \) and \( C \) in Eq. (5.38).

**Step 2:** Replace \( A, B, \) and \( C \) with the estimates obtained in Step 1 and solve the algebraic Riccati equation or iterate Eq. (5.47) to obtain \( \tilde{L} \).

**Step 3:** Calculate the control signal from Eq. (5.40).

Repeat Steps 1, 2, and 3 at each sampling period.

**Remark.** If \( Q_1 = C^T \tilde{C} \), the steady-state solution to Eq. (5.46) will give the same result as the minimization of Eq. (5.39).

Algorithms 5.6 and 5.7 are indirect algorithms that are able to handle non-minimum-phase systems and varying time delays. The computations are more extensive for these algorithms than for the adaptive predictive controllers 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 using an eigenvalue method or by some iterative method. The iterative methods
will in general lead to shorter code; it is not necessary to iterate until the steady-state solution is obtained. In some algorithms it is suggested that the Riccati equation is iterated only one step at each sampling. It is, however, difficult to cope with iterations in an on-line algorithm. 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.

## 5.6 Adaptive Predictive Control

Algorithm 5.4 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 non-minimum-phase systems, as was illustrated in Section 5.3. Several other ways to achieve predictive control have been suggested in the literature; some of these will now be discussed and analyzed. As with the previous algorithms, it is of great importance to determine the underlying control problem in order to understand the asymptotic properties of the algorithms. 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*.

### 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. (5.30). Consider the process

$$A^*(q^{-1})y(t) = B^*(q^{-1})u(t - d_0) \quad (5.48)$$

and introduce the identity

$$1 = A^*(q^{-1})F_d^*(q^{-1}) + q^{-d}G_d^*(q^{-1}) \quad (5.49)$$

where

$$\deg F_d^* = d - 1$$

$$\deg G_d^* = n - 1$$
The index \(d\) is used to indicate that the prediction horizon is \(d\) steps. It is assumed that \(d \geq d_0\). The polynomial identity of Eq. (5.48) can be used to predict the output \(d\) steps ahead. Hence

\[
y(t + d) = A^* F_d^* y(t + d) + G_d^* y(t) = B^* F_d^* u(t + d - d_0) + G_d^* y(t) \tag{5.50}
\]

Introduce

\[
B^*(q^{-1}) F_d^*(q^{-1}) = R_d^*(q^{-1}) + q^{-(d - d_0 + 1)} \tilde{R}_d^*(q^{-1})
\]

where

\[
\text{deg } R_d^* = d - d_0 \\
\text{deg } \tilde{R}_d^* = n - 2
\]

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:

\[
\frac{q^{-d_0} B^*}{A^*} = q^{-d_0} B^* \left( F_d^* + q^{-d} G_d^* \right) \frac{1}{A^*}
= q^{-d_0} R_d^*(q^{-1}) + q^{-(d + 1)} \tilde{R}_d^*(q^{-1}) + \frac{B^*(q^{-1}) G_d^*(q^{-1})}{A^*(q^{-1})} q^{-(d + d_0)}
\tag{5.51}
\]

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 \(\text{deg } R_d^* = d - d_0\).

Equation (5.50) can be written as

\[
y(t + d) = R_d^*(q^{-1}) u(t + d - d_0) + \tilde{R}_d^*(q^{-1}) u(t - 1) + G_d^*(q^{-1}) y(t)
\]

\[
= R_d^*(q^{-1}) u(t + d - d_0) + \tilde{y}_d(t) \tag{5.52}
\]

\(R_d^*(q^{-1}) u(t + d - d_0)\) depends on \(u(t), \ldots, u(t + d - d_c)\), while \(\tilde{y}_d(t)\) is a function of \(u(t - 1), u(t - 2), \ldots\), and \(y(t), y(t - 1), \ldots\). The variable \(\tilde{y}_d(t)\) 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, i.e., that

\[
u(t) = u(t + 1) = \cdots = u(t + d - d_0) \tag{5.53}
\]

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, i.e., to minimize

\[
\sum_{k=t}^{t+d} u(k)^2 \tag{5.54}
\]
Constant Future Control

Choosing the predicted output equal to the desired output $y_m$ and assuming that Eq. (5.53) holds,

$$(R_d^*(1) + q^{-1}\tilde{R}_d^*(q^{-1})) u(t) + G^*_d(q^{-1})y(t) = y_m(t + d)$$

The control law is

$$u(t) = \frac{y_m(t + d) - G^*_d(q^{-1})y(t)}{R_d^*(1) + \tilde{R}_d^*(q^{-1})q^{-1}} \quad (5.55)$$

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. (5.55) is used again. Note that the value of the control signal is changed rather than kept constant, as was assumed when Eq. (5.55) was derived. The receding-horizon control principle is thus used. Note that the control law is stationary, in contrast to a fixed-horizon linear quadratic controller.

We will now analyze the closed-loop system when Eq. (5.55) is used to control the process of Eq. (5.48). 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. (5.49) can be written in the forward shift operator as

$$q^{n+d-1} = A(q)F_d(q) + G_d(q) \quad (5.56)$$

The characteristic polynomial of the closed-loop system is

$$P(q) = A(q) \left( q^{n-1}R_d(1) + \tilde{R}_d(q) \right) + G_d(q)B(q) \quad (5.57)$$

where

$$\deg P = \deg A + n - 1 = 2n - 1$$

The design equation (Eq. 5.56) can now be used to rewrite $P(q)$:

$$B(q)q^{n+d-1} = A(q)B(q)F_d(q) + G_d(q)B(q) = A(q) \left( q^{n-1}R_d(q) + \tilde{R}_d(q) \right) + G_d(q)B(q)$$

Hence

$$A(q)\tilde{R}_d(q) + G_d(q)B(q) = B(q)q^{n+d-1} - A(q)q^{n-1}R_d(q)$$

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. (5.51) that the last term vanishes as \( d \to \infty \). Thus

\[
\lim_{d \to \infty} P(q) = q^{n-1} A(q) R_d(1) \quad \text{if } A(z) \text{ is a stable polynomial}
\]

The properties of the predictive control law are illustrated by an example.

**Example 5.7—Predictive control**

Consider the process model

\[
y(t+1) = ay(t) + bu(t)
\]

The identity of Eq. (5.56) gives

\[
q^d = (q - a)(q^{d-1} + f_1q^{d-2} + \cdots + f_{d-1}) + g_0
\]

Hence

\[
F(q) = q^{d-1} + aq^{d-2} + \cdots + a^{d-1}
\]

\[
G(q) = a^d
\]

\[
R_d(q) = b F(q)
\]

\[
\bar{R}_d(q) = 0
\]

and the control law becomes, when \( y_m = 0 \),

\[
u(t) = -\frac{a^d}{b(1 + a + \cdots + a^{d-1})} y(t) = -\frac{a^d(a - 1)}{b(a^d - 1)} y(t)
\]

The characteristic polynomial of the closed-loop system is

\[
P(q) = q - a + \frac{a^d(a - 1)}{a^d - 1}
\]

which has the pole

\[
p_d = \frac{a^d - a}{a^d - 1}
\]

The location of the pole is given by

\[
0 \leq p_d < a \quad |a| \leq 1 \quad \text{(stable system)}
\]

\[
0 \leq p_d < 1 \quad |a| > 1 \quad \text{(unstable system)}
\]
The closed-loop pole for different values of $a$ and $d$ is shown in Fig. 5.16. 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 5.7 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. (5.53) is then not very useful.

**Minimum Control Effort**

The control strategy that brings $y(t + d)$ to $y_m(t + d)$ while minimizing Eq. (5.54) will now be derived. Equation (5.52) is

$$
y(t + d) = R_d^*(q^{-1})u(t + d - d_0) + \bar{y}_d(t)
= r_{d0}u(t + \nu) + \cdots + r_{d\nu}u(t) + \bar{y}_d(t)
$$

where $\nu = d - d_0$. Introduce the Lagrangian

$$
2J = u(t)^2 + \cdots + u(t + \nu)^2 + 2\lambda \left( y_m(t + d) - \bar{y}_d(t) - R_d^*(q^{-1})u(t + \nu) \right)
$$

Equating the partial derivatives with respect to $u(t), \cdots, u(t + \nu)$ and $\lambda$ to zero gives

$$
u(t) = \lambda r_{d\nu}
$$

$$
u(t + \nu) = \lambda r_{d0}
$$

$$
y_m(t + d) - \bar{y}_d(t) = r_{d0}u(t + \nu) + \cdots + r_{d\nu}u(t)
$$

This set of equations gives

$$
u(t) = \frac{y_m(t + d) - \bar{y}_d(t)}{\mu}
$$
where
\[ \mu = \frac{\sum_{i=0}^{\nu} r_{di}^2}{r_{d\nu}} \]

Using the definition of \( \gamma_d(t) \) gives
\[ \mu u(t) = y_m(t + d) - \tilde{R}_d^* u(t - 1) - G_d^* y(t) \]
or
\[ u(t) = \frac{y_m(t + d) - G_d^* y(t)}{\mu + q^{-1} R_d^*} = \frac{y_m(t + d + n - 1) - G_d(q) y(t)}{\mu q^{n-1} + R_d(q)} \quad (5.58) \]

Using Eq. (5.58) and the model of Eq. (5.48) gives the closed-loop characteristic polynomial
\[ P(q) = A(q) \left( q^{n-1} \mu + \tilde{R}_d(q) \right) + G_d(q) B(q) \]

This is of the same form as Eq. (5.55), with \( R_d(1) \) replaced by \( \mu \). 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 system is unstable? Consider the following example.

Example 5.8—Minimum-effort control
Consider the same system as in Example 5.7. The minimum-effort controller is in this case given by
\[ \mu = \frac{b(1 + a^2 + \cdots + a^{2(d-1)})}{a^{d-1}} = \frac{b(a^{2d} - 1)}{a^{d-1}(a^2 - 1)} \]

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
\[
\lim_{d \to \infty} p_d = a \quad |a| \leq 1 \quad \text{(stable system)}
\]
\[
\lim_{d \to \infty} p_d = 1/a \quad |a| > 1 \quad \text{(unstable system)}
\]
For this example, the minimum-effort controller gives a better closed-loop system than if the future control is assumed to be constant. \(\square\)

### Generalized Predictive Control

The predictive controllers discussed so far have considered the output at only one future point in time. Different generalizations of predictive control have been suggested, in which different loss functions are minimized. One possibility is to use

\[
J(N_1, N_2, N_u) = E \left\{ \sum_{k=N_1}^{N_2} (y(t + k) - y_m(t + k))^2 + \sum_{k=1}^{N_u} \rho \Delta u(t + k - 1)^2 \right\}
\]

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 using the loss function of Eq. (5.58) and the process model

\[
A^*(q^{-1})y(t) = B^*(q^{-1})u(t - d_0) + e(t)/\Delta
\]

This model is sometimes called the CARIMA model (Controlled Auto-Regressive Integrating Moving-Average model). It has the advantage that the controller will naturally contain an integrator. As with Eq. (5.49), the following identity is introduced:

\[
1 = A^*(q^{-1})F_d^*(q^{-1})(1 - q^{-1}) + q^{-d}G_d^*(q^{-1})
\]

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

\[
h(t + d) = F_d^* B^* \Delta u(t + d - d_0) + G_d^* y(t)
\]

Suppose that the future desired outputs, \(y_m(t + k), k = 1, 2, \ldots\) are available. The loss function of Eq. (5.59) can now be minimized, giving a sequence of future control signals. Notice that the expectation in Eq. (5.59) 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 a class called open-loop-optimal-feedback control. As the name suggests, it is assumed that feedback is used, but it is computed based only on the information available at the present time.

Using Eq. (5.52),

\[ y(t + 1) = R_1^*(q^{-1}) \Delta u(t + 1 - d_0) + \bar{y}_1(t) + F_1^* e(t + 1) \]
\[ y(t + 2) = R_2^*(q^{-1}) \Delta u(t + 2 - d_0) + \bar{y}_2(t) + F_2^* e(t + 2) \]
\[ \vdots \]
\[ y(t + N) = R_N^*(q^{-1}) \Delta u(t + N - d_0) + \bar{y}_N(t) + F_N^* e(t + N) \]

Each output value consists of future control signals (if \( d > d_0 \)), measured inputs, and future noise signals. The equations above can be written as

\[ y = R \Delta u + \bar{y} + e \]

where

\[ y = [y(t + 1) \ldots y(t + N)]^T \]
\[ \Delta u = [\Delta u(t + 1 - d_0) \ldots \Delta u(t + N - d_0)]^T \]
\[ \bar{y} = [\bar{y}_1(t) \ldots \bar{y}_N(t)]^T \]
\[ e = [F_1^* e(t + 1) \ldots F_N^* e(t + N)]^T \]

From Eq. (5.51) 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

\[
R = \begin{pmatrix}
  r_0 & 0 & \ldots & 0 \\
  r_1 & r_0 & \ldots & 0 \\
  \vdots & & \ddots & \vdots \\
  r_{N-1} & r_{N-2} & \ldots & r_0 \\
\end{pmatrix}
\]

If there is a dead time in the system, \( d_0 > 1 \), then the first \( d_0 - 1 \) rows of \( R \) will be zero. Also introduce

\[ \bar{y}_m = [y_m(t + 1) \ldots y_m(t + N)]^T \]
The expected value of the loss function can be written as

\[ J(1, N, N) = E \left\{ (y - y_m)^T(y - y_m) + \rho \Delta u^T \Delta u \right\} \]
\[ = (R \Delta u + \bar{y} - y_m)^T(R \Delta u + \bar{y} - y_m) + \rho \Delta u^T \Delta u \]

Minimization of this expression with respect to \( \Delta u \) gives

\[ \Delta u = (R^T R + \rho I)^{-1} R^T (y_m - \bar{y}) \] \hspace{1cm} (5.63)

The first component in \( \Delta u \) 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. (5.60).

The calculation of Eq. (5.63) 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:

\[ \Delta u(t + k - 1) = 0 \quad k > N_u \]

This implies that the control signal is assumed to be constant after \( N_u \) steps. Compare the constraint of Eq. (5.54). The control law (Eq. 5.63) then changes to

\[ \Delta u = (R_1^T R_1 + \rho I)^{-1} R_1^T (y_m - \bar{y}) \] \hspace{1cm} (5.64)

where \( R_1 \) is the \( N \times N_u \) matrix

\[
R_1 = \begin{pmatrix}
    r_0 & 0 & \cdots & 0 \\
    r_1 & r_0 & \cdots & 0 \\
    \vdots & \ddots & \ddots & \vdots \\
    \cdots & \cdots & \ddots & r_0 \\
    \cdots & \cdots & \cdots & \cdots \\
    r_{N-1} & r_{N-2} & \cdots & r_{N-N_u}
\end{pmatrix}
\]

The matrix to be inverted is now of the order \( N_u \times N_u \). The output and control horizons can be chosen as follows:

- \( N_1 \): If the time delay is known, then \( N_1 = d_0 \); otherwise choose \( N_1 = 1 \).
- \( N_2 \): 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.
\[ N_u: N_u = 1 \text{ often gives good results for simpler processes. For complex systems } N_u \text{ should be at least equal to the number of unstable or poorly damped poles.} \]

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. (5.64). The adaptive generalized predictive controller is thus an indirect control algorithm. The predictions of Eq. (5.62) can be computed recursively, which will simplify the computations. Finally, \( N_u \) is usually small, which implies that only a low-order matrix has to be inverted.

The control signal \( \Delta u(t) \) is obtained from Eq. (5.64) as

\[
\Delta u = [1 \ldots 0] [R_1^T R_1 + \rho I]^{-1} R_1^T [y_m - \bar{y}] = [\alpha_1 \ldots \alpha_N][y_m - \bar{y}]
\]

Further, from Eq. (5.60), using Eq. (5.51),

\[
\bar{y} = \left\{ \begin{array}{l}
\bar{R}_1^* \Delta u(t-1) + G_1^* y(t) \\
\vdots \\
\bar{R}_N^* \Delta u(t-1) + G_N^* y(t)
\end{array} \right\} = \left\{ \begin{array}{l}
\frac{\bar{R}_1^* A^* \Delta}{B^*} q^{d_0-1} + G_1^* \\
\vdots \\
\frac{\bar{R}_N^* A^* \Delta}{B^*} q^{d_0-1} + G_N^*
\end{array} \right\} y(t)
\]

The closed-loop system has the characteristic equation

\[
A^* \Delta + [\alpha_1 \ldots \alpha_N] \begin{pmatrix} \bar{R}_1^* A^* \Delta q^{d_0-1} + B^* G_1^* \\
\vdots \\
\bar{R}_N^* A^* \Delta q^{d_0-1} + B^* G_N^* \end{pmatrix}
\]

The identity of Eq. (5.61) gives

\[
B^* = A^* \Delta B^* F_d^* + q^{-d} G_d^* B^*
\]
\[= A^* \Delta (R_d^* + q^{-(d-1)+1} \bar{R}_d^*) + q^{-d} G_d^* B^*\]

This gives the characteristic equation

\[
A^* \Delta + [\alpha_1 \ldots \alpha_N] \begin{pmatrix} (B^* - A^* \Delta R_d^*)q \\
\vdots \\
(B^* - A^* \Delta R_N^*)q^N \end{pmatrix} = A^* \Delta + \sum_{i=1}^{N} \alpha_i q^i (B^* - A^* \Delta R_i^*)
\]

(5.65)
Equation (5.65) 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 $\rho$ 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.

### 5.7 A Priori Knowledge in STR

In Section 4.6 it was discussed how a priori knowledge can be incorporated into an MRAS. The same ideas can also be used for self-tuning regulators. One difficulty with STRs is that they are usually implemented as discrete-time controllers, but the a priori knowledge is usually available in continuous-time form. This will give rise to difficulties when utilizing the a priori knowledge. The problems are highlighted by an example.

**Example 5.9—Reparameterization through sampling**

Consider the continuous-time system with the transfer function
\[ G(s) = \frac{1}{(1 + \theta_1 s)(1 + \theta_2 s)} \]

The parameter $\theta_1$ is assumed to be known, while $\theta_2$ is an unknown parameter. The pulse transfer function with the sampling interval $h$ has the form
\[ H(z) = \frac{b_1 z + b_2}{z^2 + a_1 z + a_2} \]

where
\[ b_1 = \frac{\theta_1 (1 - e^{-h/\theta_1}) - \theta_2 (1 - e^{-h/\theta_2})}{\theta_1 - \theta_2} \]
\[ b_2 = \frac{\theta_2 (1 - e^{-h/\theta_2}) e^{-h/\theta_1} - \theta_1 (1 - e^{-h/\theta_1}) e^{-h/\theta_2}}{\theta_1 - \theta_2} \]
\[ a_1 = -(e^{-h/\theta_1} + e^{-h/\theta_2}) \]
\[ a_2 = e^{-(1/\theta_1 + 1/\theta_2) h} \]

The pulse transfer function is thus nonlinear in both $\theta_1$ and $\theta_2$. 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 \( \theta_2 \) will influence all the coefficients in the sampled data model.

The example shows that it is crucial how the process model is parameterized. A large class of physical systems have a transfer function of the form

\[
G(s) = \frac{B_0(s) + \sum k_i B_i(s)}{A_0(s) + \sum k_i A_i(s)} \tag{5.66}
\]

where \( A_i(s) \) and \( B_i(s) \) are known polynomials and \( k_i \) are multilinear functions of unknown parameters \( \theta_i \) in the continuous-time, state space representation.

Example 5.10—Reparameterization
Consider the circuit in Fig. 5.17. The state space representation is

\[
\dot{x} = \begin{pmatrix} 0 & -1/C \\ 1/L & -R/L \end{pmatrix} x + \begin{pmatrix} 1/C \\ 0 \end{pmatrix} u
\]

\[y = [0 \ R] x\]

and the transfer function is

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

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

\[
G(s) = \frac{k_1}{s^2 + k_2 s + k_3} \tag{5.67}
\]
and it is possible to make a constrained estimation of $\theta_1$, $\theta_2$, and $\theta_3$ using Eq. (5.67) and

$$
k_1 = \theta_1 \theta_2 \theta_3
$$

$$
k_2 = \theta_1 \theta_2
$$

$$
k_3 = \theta_2 \theta_3
$$

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

Another way to circumvent the problem with the sampling in Example 5.9 is to use the $\delta$-operator and fast sampling. The $\delta$-operator is defined by

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

where $q$ is the forward shift operator and $h$ the sampling interval. Let $B_h(\delta)$ and $A_h(\delta)$ be the numerator and denominator of the transfer function in the $\delta$-form when the sampling period is $h$. It can be shown that

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

is such that the coefficients in the $A_0$ and $B_0$ polynomials are the same as the coefficients in the continuous-time transfer function. This implies that the structure of the transfer function in the $\delta$-operator is essentially the same as that of the continuous-time transfer function, provided the sampling interval is sufficiently short. For instance, the following parameterizations can be used:

$$
H(\delta) = \frac{B_0(\delta)}{A_0(\delta)} \cdot \frac{B_1(\delta)}{A_1(\delta)}
$$

(5.68)

$$
H(\delta) = \frac{B_0(\delta)}{A_0(\delta)} + \sum \beta_i B_i(\delta) + \sum \alpha_i A_i(\delta)
$$

(5.69)

In Eq. (5.68) $B_0$ and $A_0$ are assumed to be known, and $B_1$ and $A_1$ are unknown. In Eq. (5.69) $B_i$ and $A_i$ are known, but $\beta_i$ and $\alpha_i$ are assumed to be unknown. The unknown parameters or polynomials in Eqs. (5.68) and (5.69) can be estimated, and the controller can be designed for the full process model.

For a direct STR it is possible to use the same methodology as for a direct MRAS discussed in Section 4.6. The problem illustrated in Example 5.9 can be avoided by using the $\delta$-operator formulation.
5.8 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 when making the design of the controller. It is sometimes of interest to include the uncertainties of the parameter estimates in the design. Such regulators 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 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 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 underlying control problem be carefully chosen when 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 interactor matrix of the system is known.

Problems

5.1 Show that Eq. (5.36) minimizes Eq. (5.35).

5.2 Consider the feedback system shown in Fig. 5.18 (a) and (b). Show that they are equivalent and that

\[ G'_R = \frac{G_R}{1 + \rho G_R} \]

Show that \(|G'_R| < 1/\rho\) when \(|G_R|\) is large.
5.3 Consider the process and controller in Example 5.2. 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.

5.4 Construct discrete-time direct and indirect self-tuning algorithms for the system given in Problem 4.7.

5.5 Construct continuous and discrete-time indirect self-tuning algorithms for the system in Problem 4.8.

5.6 Construct a continuous-time indirect self-tuning algorithm for the system in Problem 4.9.

5.7 Consider the basic direct self-tuning controller in Algorithm 5.4. Discuss different ways to incorporate reference values in the controller. (a) Use the difference \( y - u_c \) instead of \( y \) in the algorithm and introduce an integrator in the regulator. (b) Estimate the parameters using the model

\[
y(t + d) = R^*u + S^*y - T^*u_c + \varepsilon
\]

and let the controller be

\[
R^*u = -S^*y + T^*u_c
\]

What are the properties of the two methods?

5.8 Consider Problem 5.7 (a). What will happen if \( u_c - y \) is used instead?

5.9 Consider the process in Example 5.3. 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?

5.10 Consider the system in Example 5.4. 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 non-minimum-phase case).

*Hint:* Use Theorem 5.3 for the non-minimum-phase case.

5.11 Make the same calculations as in the previous problem, but for the moving-average controller with $d = 2$.

5.12 Consider the generalized minimum-variance controller of Eq. (5.36). 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 5.4 when $\tau = 0.6$?

5.13 Consider the process in Example 5.4 when $\tau = 0.6$ and $C = 0$. Use Eq. (5.64) to compute the closed-loop poles for different values of $N$ when $N_u = 1$.

5.14 Write computer programs to simulate a direct and an indirect self-tuning regulator. Notice that it can be useful to divide the algorithm into an estimation and a design part. Use the programs to verify the examples in this chapter.

5.15 Consider the system in Problem 2.7

(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 $\epsilon$ is discrete-time measurement noise. Determine the minimum-variance controller for the system.

(d) Simulate a self-tuning moving-average controller for different prediction horizons.

5.16 Make the same investigation as in Problem 5.15, but for the process in Problem 2.8.

5.17 Show that the moving average controller with $B^+ = 1$ and $d = n$ corresponds to a state deadbeat controller.

**References**

There are many papers, reports, and books written about self-tuning algorithms. Some fundamental references are given in this section. The first publication of the self-tuning idea is probably:

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:


The first thorough presentation and analysis of a self-tuning regulator was given in:


A revised version of this paper, in which the phrase “self-tuning regulator” was coined, is:


Different aspects of the basic self-tuning regulator described in Algorithm 5.4 is given in the thesis,


The generalized minimum-variance self-tuner was presented in:


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:


Pole placement and model-reference type self-tuners are treated in:


The problem of controlling non-minimum-phase plants is discussed in:


In the latter, the moving-average controller is presented. Algorithm 5.5 can be used to explain the pole/zero assignment regulator in Wellstead et al. (1979). 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. (5.30) is used, but with different filtering.

Continuous-time self-tuning regulators are discussed in:


The former also gives a unification of MRAS and STR. Adaptive feedforward is discussed further in Chapters 11 and 12.

Multivariable self-tuning regulators are treated in:


Adaptive predictive control is discussed in:


Linear quadratic Gaussian self-tuning regulators are treated in:


A detailed treatment of LQG self-tuners is given in:


It contains much information and many useful hints for practical applications.

The parameterization issues discussed in Section 5.7 are further treated in:


The δ-operator is discussed in:


Different ways to use a priori knowledge in discrete-time adaptive controllers are suggested in Wittenmark (1988), above, and in:

Clary, J. B., and G. F. Franklin, 1984. “Self-tuning control with a priori plant knowledge.” *Proc. 23rd IEEE Conference on Decision and
Chapter 5 Self-tuning Regulators

Control: pp. 369–374. Las Vegas, NV.


Parameter estimation is an important part of self-tuning regulators. Fundamental works are listed in the References in Chapter 3. Estimation of continuous-time models from discrete-time measurements is found in:


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


Digital design methods that are useful in self-tuning regulators are given in:


Chapter 6

STABILITY, CONVERGENCE, AND ROBUSTNESS

6.1 Introduction

Some theoretical problems have been discussed in earlier chapters in connection with description of specific algorithms. In this chapter we attempt to bring together theory of a more general character. The theory has several different goals, including

- To give insight into the properties of a specific algorithm.
- To describe behavior of algorithms in nonideal cases.
- To give ideas for new algorithms.

The behavior of specific algorithms can be understood through analysis of stability, convergence, and performance. Stability proofs require specific assumptions. It is then of considerable interest to have techniques for analyzing a particular algorithm to understand how it behaves in nonideal circumstances. 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 above. 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 due to their 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 results in this chapter are obtained by investigating the adaptive problem from different points of view. Stability is discussed in Section 6.2. A global stability proof is given for the standard direct algorithm. Convergence is discussed in Section 6.3. There is a drastic difference in convergence rates for different problems and different algorithms: understanding this is essential for the design of adaptive systems. The analysis is carried out from the viewpoint of parameter estimation. It shows that convergence is closely related to persistency of excitation. Nonlinear system theory is useful to get a more detailed insight into the behavior of adaptive systems. The fact that parameters change more slowly than the other variables of the system can be used to simplify the analysis. Averaging methods are good tools for this analysis, as developed in Sections 6.4 and 6.5. Several assumptions are required for the stability proof in Section 6.2. The consequences of violating some assumptions are discussed in Section 6.6. Averaging is used to explore the consequences of disturbances and of the fact that the adaptive system is based on a simplified model. The analysis gives further insight into the behavior of adaptive systems and gives suggestions for modifying the algorithms. The idea of averaging can also be applied to the stochastic case, which is explored in Section 6.7. Section 6.8 investigates whether it is necessary to know the sign of the high-frequency gain (as was assumed in Section 6.2). This leads to the notion of "universal stabilizers." The parameterization problem is discussed in Section 6.9, and different mechanisms that can create instability are discussed in Section 6.10.

### 6.2 Global Stability

Stability is a key requirement for a control system, but stability analysis of adaptive systems is difficult because the systems are nonlinear. The fundamental stability concept for nonlinear systems refers to the stability of a particular solution, i.e., Lyapunov stability. It may happen that one solution is stable and another is unstable. There can therefore be no
general notion of a stable nonlinear system. The behavior of a system may depend drastically on the command signal and the disturbances. The solution obtained for one command signal may be stable. Another command signal may give an unstable solution. Also, although a solution is stable, a perturbed motion may diverge if the perturbations are sufficiently large.

Stability was discussed in connection with model-reference adaptive system in Chapter 4. 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 gain can be used.

A stability proof for a direct 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 Process Model

A simple, direct, discrete-time adaptive control law based on pole placement will be investigated.

Consider a process described by the difference equation

$$A^*(q^{-1})y(t) = B^*(q^{-1})u(t - d)$$  \hspace{1cm} (6.1)

The model is written in the delay operator $q^{-1}$. For this purpose the reciprocal $A^*(z)$ of the polynomial $A(z)$ has been introduced. See Section 5.1. This polynomial is defined as

$$A^*(z) = z^nA(z^{-1})$$

If the parameters of the polynomials $A^*$ and $B^*$ are known a pole placement design, which gives the following response to command signals $u_c$

$$A_m^*(q^{-1})y(t) = t_0 u_c(t - d)$$

is obtained by solving the equation

$$A^*(q^{-1})R^*(q^{-1}) + q^dB^*(q^{-1})S^*(q^{-1}) = A_o^*(q^{-1})A_m^*(q^{-1})B^*(q^{-1})$$

where $A_o$ is the desired observer polynomial. Hence

$$A^*R^*y(t) + q^dB^*S^*y(t) = A_o^*A_m^*B^*y(t)$$

Using Eq. (6.1) to eliminate $y$ in the first term, we get

$$q^dB^*R^*u(t) + q^dB^*S^*y(t) = A_o^*A_m^*B^*y(t)$$
or asymptotically, if $B$ is stable,

$$A_o^* A_m^* y(t + d) = R^* u(t) + S^* y(t) = A_o^* A_m^* \phi^T(t) \theta^0$$

(6.2)

where

$$\theta^0 = [r_0 \ r_1 \ldots r_k \ s_0 \ s_1 \ldots s_l]^T$$

$$\varphi(t) = \frac{1}{A_o^* A_m^*} \begin{bmatrix} u(t) \ u(t - 1) \ldots u(t - k) \ y(t) \ y(t - 1) \ldots y(t - l) \end{bmatrix}^T$$

(6.3)

where deg $R = k$ and deg $S = l$. The fact that Eq. (6.2) is linear in the parameters can be used as a basis for an adaptive control law.

**Parameter Estimation**

The parameters are estimated using the following simple estimator:

$$\theta(t) = \theta(t - 1) + \frac{\gamma \varphi(t - d)}{\alpha + \varphi^T(t - d) \varphi(t - d)} e(t)$$

(6.4)

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

with $0 < \gamma < 2$ and $\alpha > 0$. This estimator is a simple normalized gradient scheme. For $\gamma = 1$ and $\alpha = 0$, it reduces to Kacmarcz’s projection algorithm.

**An Adaptive Control Law**

An adaptive control law can now be formulated as follows.

**Algorithm 6.1—DAPA (Direct Adaptive Pole Assignment)**

Repeat the following steps at each sampling instant.

1. Update the parameter estimates by Eq. (6.4).
2. Determine a control law such that

$$\hat{R}^* u(t) + \hat{S}^* y(t) = t_0 A_o^* u_c(t)$$

(6.5)

or equivalent by

$$\theta^T(t) (A_o^* A_m^* \varphi(t)) = t_0 A_o^* u_c(t)$$

(6.6)

where $u_c(t)$ is the desired setpoint.

**Remark.** Notice that it must be required that $\theta_1(t) = \hat{r}_0(t) \neq 0$; otherwise the control law is not causal.
6.2 Global Stability

Preliminaries

Since the proof consists of several steps, we will outline the basic idea. The properties of the estimator will first be explored. It will be proven that the estimates are bounded and that a normalized prediction error converges to zero. However, it cannot be shown 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.

The properties of the estimator will first be established.

Lemma 6.1—Estimator properties

Let the estimator of Eq. (6.4) be applied to the system of Eq. (6.2). Then

\[
\begin{align*}
(i) & \quad \|\theta(t) - \theta^0\| \leq \|\theta(t - 1) - \theta^0\| \leq \|\theta(0) - \theta^0\|, \quad t \geq 1 \\
(ii) & \quad \lim_{t \to \infty} \frac{e(t)}{\sqrt{\alpha + \varphi^T(t - d)\varphi(t - d)}} = 0 \\
(iii) & \quad \lim_{t \to \infty} \|\theta(t) - \theta(t - k)\| = 0 \quad \text{for any finite } k
\end{align*}
\]

Proof: The proof is only a minor modification of Theorem 3.9.

Remark 1. Notice that the result holds for all input sequences \(\{u(t)\}\). It applies to open-loop control as well as any closed-loop control.

Remark 2. Notice that the result does not imply that the estimates \(\theta(t)\) converge. \(\square\)

If the input and output signals of the system can be shown to be bounded, then \(\varphi\) given by Eq. (6.3) is bounded. If \(\dot{\varphi}(t - d)\) is bounded for all \(t\), it follows from Property (ii) of Lemma 6.1 that the prediction error \(e(t)\) goes to zero. Since \(\varphi\) is given by Eq. (6.3), it follows that \(\varphi\) is bounded. The following result is useful to establish this.

Lemma 6.2—Key technical lemma

Let \(\{s_t\}\) be a sequence of real numbers and \(\{\sigma_t\}\) a sequence of vectors such that

\[
\|\sigma_t\| \leq c_1 + c_2 \max_{0 \leq k \leq t} |s_k|
\]

Assume that

\[
z_t = \frac{s_t^2}{\alpha_1 + \alpha_2 \sigma_t^T \sigma_t} \to 0 \quad \text{ (6.7)}
\]

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
\[ |s_t| \leq s_t \quad \text{for } t \leq t_n. \] For this sequence, it follows that
\[
\left| \frac{s_t^2}{\alpha_1 + \alpha_2 \sigma_t^2 \sigma_t} \right| \geq \frac{s_t^2}{\alpha_1 + \alpha_2 (c_1 + c_2 |s_t|)^2} \geq \frac{1}{\alpha_3 c_2^2} > 0
\]
where \(0 < \alpha_3 < \alpha_2\). This contradicts Eq. (6.7) and proves the statements.

\[ \square \]

Main Result

The main result can now be stated as the following theorem.

**Theorem 6.1  Boundedness and convergence**

Consider a system described by Eq. (6.1). Let the system be controlled with the adaptive control algorithm DAPA, 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, and

(ii) \(\lim_{t \to \infty} |A_m^*(q^{-1})y(t) - t_0 u_c(t - d)| = 0\)

**Proof:** Introduce the filtered control error
\[
\varepsilon(t) = A_o^* (A_m^* y(t) - t_0 u_c(t - d)) = P^* y(t) - t_0 A_o^* u_c(t - d)
\]
\[
= P^* y(t) - \theta^T (t - d) (P^* \varphi(t - d))
\]
\[
= P^* e(t) + P^* (\theta^T (t - 1) \varphi(t - d)) - \theta^T (t - d) (P^* \varphi(t - d)) \quad (6.8)
\]
\[
= P^* e(t) + \sum_{i=0}^{\deg P} p_i (\theta(t - 1 - i) - \theta(t - d))^T \varphi(t - d - i)
\]

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. (6.6), the fourth from Eq. (6.4), and the last by expanding the expression.

It now follows from Properties (ii) and (iii) of Lemma 6.1 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. (6.8) 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
\[ |y(t)| \leq \alpha_1 + \beta_1 \max_{0 \leq k \leq t} |\varepsilon(k)| \]
Moreover, since the polynomial \( B \) is stable, it follows that
\[ |u(t - d)| \leq \alpha_2 + \beta_2 \max_{0 \leq k \leq t} |y(k)| \]
Hence
\[ |\varphi(t - d)| \leq \alpha_3 + \beta_3 \max_{0 \leq k \leq t} |\varepsilon(k)| \]
Applying Lemma 6.2, it now follows that \( \varphi(t) \) is bounded and that \( \varepsilon(t) \to 0 \) as \( t \to \infty \). Since the polynomial \( A_o^* \) is stable, (ii) also follows.

**Remark 1.** The proof can be modified to cover several modifications of the algorithm. For example, it also holds if the regressors are defined by
\[ \varphi(t) = [u(t) \ u(t-1) \ldots u(t-k) \ y(t) \ y(t-1) \ldots y(t-l)] \]
instead of by Eq. (6.3). As will be shown later, there is a significant advantage to using the filtered regressors in the algorithm.

**Remark 2.** Notice that 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 \( \hat{r}_0(t) = 0 \), to modify \( \varphi \) to give \( \hat{r}_0(t) \neq 0 \). Lemma 6.1 will still be valid with this modification of the algorithm. Since the estimator properties enter into the proof only via Lemma 6.1, 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 unique for 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. (6.8) and the augmented error introduced in Chapter 4.

\[ \square \]

**Discussion**

It has been established that Algorithm 6.1 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 $\gamma$, 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 regulator 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

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. Lemma 6.1 then needs to be generalized. Many types of least-squares-like algorithms can be covered by replacing the function $V = \hat{\theta}^T \hat{\theta}$ in Lemma 6.1 by $V = \hat{\theta}^T P^{-1} \hat{\theta}$ 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 when the control signal is bounded. Lemma 6.1 still holds but Theorem 6.1 does not, since Eq. (6.5) does not hold when the control signal saturates.

Effects of Disturbances

So far we have treated only the ideal case, in which there are no disturbances, but the results can be extended in different directions to cover
disturbances. Consider the case in which the process is described by

\[ A(q)y(t) = B(q)u(t) + v(t) \]  \hspace{1cm} (6.9)

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

**Example 6.1—Bounded disturbances**

Consider the system

\[ y(t + 1) + ay(t) = u(t) + v(t + 1) \]

Use an adaptive control law with \( A^*_o = A^*_m = 1 \). (The desired response is thus \( y_m(t + 1) = u_c(t) \).) The control law is

\[ u(t) = -\theta(t)y(t) + u_c(t) \]

where

\[ \theta(t + 1) = \theta(t) + \frac{y(t)}{1 + y^2(t)} e(t + 1) \]

\[ e(t + 1) = y(t + 1) - \theta y(t) - u(t) \]

Introduce

\[ \tilde{\theta} = \theta - \theta^0 \]

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

\[ \tilde{\theta}(t + 1) = \frac{1}{1 + y^2(t)} \tilde{\theta}(t) + \frac{y(t)v(t + 1)}{1 + y^2(t)} \]  \hspace{1cm} (6.10)

\[ y(t + 1) = -\tilde{\theta}(t)y(t) + u_c(t) + v(t + 1) \]

The key idea is now to find a disturbance \( v \) and a command signal \( u_c \) such that the parameter error goes to infinity. Assume that initial conditions are chosen such that \( \tilde{\theta}(1) = 0 \) and \( y(1) = 1 \). Define

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

for some large \( T \). Choose the following disturbance

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

and the following command signal

\[ u_c(t - 1) = \frac{1}{\sqrt{t}} - f(t), \quad t = 2, 3, \ldots, T - 5 \]
Chapter 6 Stability, Convergence, and Robustness

The signals $v$ and $u$ are bounded. A straightforward calculation gives

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

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

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

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

<table>
<thead>
<tr>
<th>$t$</th>
<th>$\tilde{\theta}(t)$</th>
<th>$y(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T - 4$</td>
<td>$\sqrt{T}$</td>
<td>$-1$</td>
</tr>
<tr>
<td>$T - 3$</td>
<td>$\frac{\sqrt{T}}{2}$</td>
<td>$\sqrt{T}$</td>
</tr>
<tr>
<td>$T - 2$</td>
<td>$\frac{1}{2\sqrt{T}}$</td>
<td>$-\frac{T}{2}$</td>
</tr>
<tr>
<td>$T - 1$</td>
<td>$\frac{1}{\sqrt{TT^2}}$</td>
<td>$\frac{\sqrt{T}}{4}$</td>
</tr>
<tr>
<td>$T$</td>
<td>$\frac{16}{\sqrt{TT^3}}$</td>
<td>$1$</td>
</tr>
</tbody>
</table>

Now choose $v(T+1)$ and $u_c(T)$ such that $\tilde{\theta}(T+1) = 0$ and $y(T+1) = 1$. The state vector of Eq. (6.10) is then equal to the initial state. By repeating the procedure for increasing values of $T$ a subsequence of $y(t)$ will increase as $-T/2$ and therefore is unbounded. \hfill \Box

The example shows that the algorithm may behave badly even if it is assumed that the disturbances are bounded.

**Theorem 6.2—Conditional updating**

Consider the plant

$$A^* y = B^* u + v$$

where $v$ is a disturbance that is bounded by

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

where \( R \) is the polynomial in the feedback law and \( C_1 \) is a constant. Assume that the adaptive control law DAPA is used, with the modification that parameters are updated only when the estimation error is such that

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

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

The proof is technical and quite complicated. See Egardt (1979). The modification of the algorithm is referred to as “conditional updating” or “introduction of a dead-zone in the estimator.”

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

\[
|e(t)| \geq 2 \sup |v(t)|
\]

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

Another modification of the algorithm also leads to bounded signals. The modification consists of using the updating law of Eq. (6.4) if the magnitude of the estimates is less than a given bound and to project into a bounded set if Eq. (6.4) gives estimates outside the bounds. We refer to Egardt (1979), Theorem 4.4, for details. This method will of course, require that the bounds on the parameters are known \textit{a priori}.

6.3 Convergence

When analyzing a direct algorithm in the previous section, it was shown that the control error converged although the parameter estimates did not necessarily converge. Another example is given in Example 4.3. This behavior is typical for direct algorithms. Parameter convergence is thus not necessary for error convergence in direct algorithms. However, for indirect algorithms it is essential to have parameter convergence. The performance will be very poor otherwise, since the regulator parameters are directly related to the process parameters. Analysis of parameter convergence also contributes to the understanding of adaptive systems.

Parameter convergence is closely related to system identification. The issues of identifiability and persistency of excitation will play an essential
role. The problem of convergence rates will also be discussed. The convergence rate depends on the algorithm used and the amount of excitation.

A Deterministic Case

Parameter convergence will first be discussed using the simple model

$$y(t) = \varphi^T(t)\theta^0$$

which is linear in the parameters; there 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

$$\theta(n) = \left( \sum_{k=1}^{n} \varphi(k)\varphi^T(k) \right)^{-1} \sum_{k=1}^{n} \varphi(k)y(k) = \theta^0$$  \hspace{1cm} (6.11)

If the estimate is instead calculated by recursive least squares, the following estimate is obtained:

$$\theta(n) = \left( P^{-1}(0) + \sum_{k=1}^{n} \varphi(k)\varphi^T(k) \right)^{-1} \left( \sum_{k=1}^{n} \varphi(k)y(k) + P^{-1}(0)\theta(0) \right)$$  \hspace{1cm} (6.12)

where \( \theta(0) \) 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.

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 that \( \sum \varphi(k)\varphi^T(k) \) is of full rank. When the parameters are changing, a least-squares estimator, where the covariance is regularly reset to \( \alpha I \), is a good implementation. This procedure is called covariance resetting. Such an estimator will give correct estimates with a settling time of at most \( 2n \) steps when the parameters are piece wise constant. The settling time can be decreased to \( n \) if \( n \) estimators are run in parallel.

Gradient Algorithms

The gradient algorithms are simpler than the least-squares algorithm, but they have considerably slower convergence rates. A typical gradient algorithm is

$$\theta(t) = \theta(t - 1) + \frac{\gamma \varphi(t)}{\alpha + \varphi^T(t)\varphi(t)}e(t)$$  \hspace{1cm} (6.13)
where
\[ e(t) = \varphi^T(t)(\theta^0 - \theta(t - 1)) \]

The estimation error is thus given by
\[ \tilde{\theta}(t) = A(t - 1)\tilde{\theta}(t - 1) \quad (6.14) \]

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. (6.14). 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 matrix \( A(t - 1) \) is the identity on the space orthogonal to \( \varphi(t) \).

The following lemma is useful to analyze Eq. (6.14).

**Lemma 6.3—Stability of a time-varying system**

Consider the time-varying system
\[
\begin{align*}
x(t + 1) &= A(t)x(t) \\
y(t) &= C(t)x(t)
\end{align*} \quad (6.15)
\]

Assume that there exists a symmetric matrix \( P(t) > 0 \) such that
\[ A^T(t)P(t + 1)A(t) - P(t) = -C^T(t)C(t) \]

Then Eq. (6.15) is stable. Moreover, if the system is uniformly completely observable, i.e., there exist \( \beta_1 > 0, \beta_2 > 0, \) and \( N > 0 \) such that
\[
\beta_1 I \leq \sum_{k=t}^{t+N-1} \Phi^T(k, t)C^T(k)C(k)\Phi(k, t) \leq \beta_2 I
\]

for all \( t \) and where \( \Phi(k, t) \) is the fundamental matrix, then Eq. (6.15) is also exponentially stable.

**Proof:** Introduce
\[ V(t) = x^T(t)P(t)x(t) \]
Hence
\[ V(t + 1) - V(t) = x^T(t)A^T(t)P(t + 1)A(t)x(t) - x^T(t)P(t)x(t) = -x^T(t)C^T(t)C(t)x(t) \leq 0 \]

The system is thus stable. Furthermore,
\[
V(t + N) - V(t) = - \sum_{k=t}^{t+N-1} x^T(k)C^T(k)C(k)x(k)
\]
\[
= -x^T(t) \left( \sum_{k=t}^{t+N-1} \Phi^T(k,t)C^T(k)C(k)\Phi(k,t) \right)x(t)
\]
\[
\leq -\beta_1 x^T(t)x(t) \leq -\frac{\beta_1}{\lambda_{max} P(t)} V(t)
\]

Hence
\[
V(t + N) \leq \left(1 - \frac{\beta_1}{\lambda_{max} P(t)} \right) V(t)
\]

Since
\[
P(t) = \sum_{k=t}^{\infty} \Phi^T(k,t)C^T(k)C(k)\Phi(k,t)
\]
it follows that $\beta_1 < \lambda_{max} P(t)$.  \hfill \Box

Applying this lemma to Eq. (6.14) we get the following theorem.

**Theorem 6.3—Global stability**

The difference equation (Eq. 6.14) is globally exponentially stable if for any $\varepsilon > 0$ there exist positive constants $\alpha$, $\beta$, and $N$ such that for all $k$
\[
0 < \alpha I \leq \sum_{k=t}^{t+N} \varphi(k)\varphi^T(k) \leq \beta I < \infty \tag{6.16}
\]

**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 $\varphi$ is suppressed. A straightforward calculation shows that Eq. (6.16) is satisfied, so the system is stable. To prove exponential stability, first observe that uniform observability of $(A(k), C(k))$ is
6.3 Convergence

Equivalent to uniform observability of \((A(k) - B(k)C(k)), C(k)\). Choosing

\[B(k) = C^T(k) = \sqrt{\frac{\gamma}{\alpha + \varphi^T \varphi}} \varphi\]

we find that \(A(k) - B(k)C(k) = I\), and uniform asymptotic stability then corresponds to Eq. (6.16).

Notice that Eq. (6.16) is equivalent to persistent excitation. 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.

Notice that there is a significant difference in convergence rate between the least-squares algorithm and a gradient algorithm. This is illustrated by an example.

**Example 6.2—Convergence rate**
Consider the MRAS for a first-order system in Example 4.6, with least-squares and gradient parameter updating. The control law is

\[u(t) = \theta_1 u_c(t) - \theta_2 y(t)\]

Figure 6.1 shows the parameter estimates obtained when the input is a square wave with period 40. Notice the drastic difference in the convergence rates. The least-squares method gives the true values with a tolerance of the graph after time 30. Also notice that the parameters change significantly only at the times when the command signal changes. This is the only time when there is significant excitation. The least-squares algorithm has converged after two level changes of the command signal as can be expected. Figure 6.2 shows the output signals in the time interval 0 to 80. Notice the superior performance of the least-squares algorithm, which is due to the fact that the parameters converge faster for that algorithm.

**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. (6.11). The covariance of the estimate for large \(t\) is

\[P(t) = \sigma^2 \left( \sum_{k=1}^{t} \varphi(k)\varphi^T(k) \right)^{-1}\]
Taking the covariance of the estimate as a measure of the rate of convergence, it is found that under uniform persistent excitation the convergence rate is at most $1/t$.

Summary

For indirect adaptive algorithms it is essential to have parameter convergence. Identifiability and persistency of excitation are important concepts. Analysis of 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. This will influence the excitation conditions and the convergence rate, as will be shown in the following sections.
Figure 6.2 Output and command signal for an MRAS with (a) gradient algorithm and (b) least-squares algorithm.

6.4 Averaging

The results in the previous sections do not admit 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. In an adaptive system it is natural to separate between 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.

Consider the system shown in Fig. 5.1. Assume that the system to be controlled is linear. Let \( \theta \) denote the regulator parameters and \( \nu \) the external driving signals. The signal \( \nu \) is typically composed of the command signal \( u_c \) and nonmeasurable disturbances acting on the process. With constant regulator parameters the closed-loop system can then be
written as

\[ \frac{d\xi}{dt} = A(\vartheta)\xi + B(\vartheta)\nu \]

\[ \eta = \begin{pmatrix} e \\ \varphi \end{pmatrix} = C(\vartheta)\xi + D(\vartheta)\nu \] \hspace{1cm} (6.17)

The state vector \( \xi \) includes the states of the system, the reference model, and the auxiliary state variables that may have to be introduced in order to calculate the error \( (e) \) and the regression vector \( (\varphi) \) used in the parameter adjustment mechanism. Since this system is linear, it can also be characterized by the differential operators \( G_{e\nu} \) and \( G_{\varphi\nu} \), which relate \( e \) and \( \varphi \) to \( \nu \). These operators depend on the regulator parameters \( (\vartheta) \).

Furthermore, let \( \vartheta \) denote the process parameters: We will consider a simple gradient scheme for adjusting the parameters:

\[ \frac{d\vartheta}{dt} = \gamma \varphi(\vartheta, \xi)e(\vartheta, \xi) \] \hspace{1cm} (6.18)

This equation can also be written as

\[ \frac{d\vartheta}{dt} = \gamma (G_{\varphi\nu}) (G_{e\nu}) \]

The control design can be represented by a nonlinear function \( \vartheta = \chi(\vartheta) \), which maps the estimated parameters into regulator parameters. This map becomes the identity for direct algorithms. The adaptive system is thus described by Eqs. (6.17) and (6.18). Notice that the equations have a very special structure. Equation (6.17) is linear in the states and the external driving signals, and nonlinearities appear as the product \( \varphi e \) in Eq. (6.18), in the design map \( \chi \), and in the functions \( A(\vartheta), B(\vartheta), C(\vartheta), \) and \( D(\vartheta) \) in Eq. (6.17). These functions are actually affine in \( \vartheta \).

**Equilibrium Analysis**

The first step in the analysis of a nonlinear differential equation is to analyze possible equilibria. In the case of adaptive systems described by Eqs. (6.17) and (6.18) it is usually not very useful to find proper equilibria, in the sense that both states \( \xi \) and parameters \( \vartheta \) are constant. Instead we will consider the case in which only the parameters are constant. This is simply to analyze the consequences of \( e \) in Eq. (6.18) being zero. If there exist parameters \( \vartheta \) such that \( e \) is identically zero for all exogeneous inputs \( \nu \), they can normally be found even for systems of high order. It frequently happens however, that such parameter values do not exist.
Averaging

The dynamic analysis 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 (2 states of the system, 4 parameters, and 2 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, there is, however, 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. (6.17) and (6.18). The rate of change of the parameter $\theta$ can be made arbitrarily small by choosing the adaptation gain $\gamma$ sufficiently small. Now consider Eq. (6.18). The product $\varphi e$ in the right-hand side depends on $\vartheta$ and $\xi$, where $\vartheta = \vartheta(\theta)$ varies slowly and $\xi$ varies fast. The key idea in the averaging method is to approximate the product $\varphi e$ by

$$G(\theta) = \text{avg} \{ \varphi(\vartheta(\theta), \xi(\vartheta(\theta), t)) e(\vartheta(\theta), \xi(\vartheta(\theta), t)) \}$$

where $\text{avg}$ denotes the average and $\xi(\vartheta(\theta), t)$ is computed under the assumption that the parameters $\theta$ are constant. The average can be computed in many ways. Typical examples are

$$\text{avg} \{ f(\theta, \xi(\theta, t), t) \} = \frac{1}{T} \int_0^T f(\theta, \xi(\theta, t), t) dt$$

$$\text{avg} \{ f(\theta, \xi(\theta, t), t) \} = \lim_{T \to \infty} \int_0^T f(\theta, \xi(\theta, t), t) dt$$

$$\text{avg} \{ f(\theta, \xi(\theta, t), t) \} = E f(\theta, \xi(\theta, t), t)$$

The first alternative is applicable when $f$ is periodic with period $T$, and the last equation applies when $\xi$ is a stationary stochastic process. The calculation of $\xi(\vartheta(\theta), t)$ is a straightforward exercise in linear system analysis. The expressions may, however, 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:

$$\frac{d\vartheta}{dt} - \gamma \text{avg} \{ \varphi(\vartheta(\theta), \xi(\vartheta(\theta), t)) e(\vartheta(\theta), \xi(\vartheta(\theta), t)) \} = 0 \quad (6.19)$$
This equation can also be written as

$$\frac{d\theta}{dt} - \gamma \text{avg} \{(G_{\varphi \nu}) (G_{\nu \nu})\} = 0 \quad (6.20)$$

Notice that the transfer functions $G_{\nu}$ and $G_{\varphi \nu}$ depend on the averaged parameter $\theta$. When the averaged equations are obtained, the behavior of the state variables $\xi$ can be obtained by linear analysis. The difference between $\theta$ and $\bar{\theta}$ can also be approximated.

The idea of averaging originated in the analysis of planetary motion. Several averaging theorems give conditions for $\theta$ being close to $\bar{\theta}$. The conditions typically require smoothness conditions of the functions involved and periodicity or near periodicity of the time functions. There are also stochastic averaging theorems. Notice that averaging analysis has already been used in Example 1.3 and in a more elaborate way in Theorem 5.2.

A significant advantage of averaging theory is that it reduces the dimensions of the problem. The theorems require that the adaptation gain should 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 well possess equilibria (i.e., solutions to $\text{avg}\{\varphi \epsilon\} = 0$), 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.

**Slow Perturbations**

The use of the averaging methods will be illustrated in a simple case. Consider the case in which external driving signals $\nu$ are changing so slowly that the closed-loop system can be approximated by static models. Let $G_{\varphi \nu}(\vartheta, p)$ and $G_{\nu \nu}(\vartheta, p)$ be the differential operators relating $\varphi$ and $\epsilon$ to $\nu$. The signals $\varphi$ and $\epsilon$ are then given by

$$\varphi(t) = G_{\varphi \nu}(\vartheta, 0) \nu(t)$$
$$\epsilon(t) = G_{\nu \nu}(\vartheta, 0) \nu(t)$$

Notice that regulator parameters $\vartheta$ depend on estimated parameters $\theta$. A true parameter equilibrium exists if the equation

$$G_{\nu}(\vartheta(\theta), 0) = 0$$
has a unique solution. A necessary condition is that $\theta$ and $\nu$ have equal dimensions. The averaged equation becomes

$$\frac{d\theta}{dt} = \gamma G_{\varphi\nu}(\vartheta(\theta), 0) R_{\nu} G^{T}_{e\varphi}(\vartheta(\theta), 0)$$  \hspace{1cm} (6.21)$$

where

$$R_{\nu} = \text{avg} \left( \nu \nu^{T} \right)$$

A necessary condition for Eq. (6.21) to have a unique parameter equilibrium is that $\nu$ and $\theta$ have equal dimension and that $R_{\nu}$ 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 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 6.1. The local stability of the equilibrium $\theta^{0}$ is given by the linearized equation

$$\frac{dx}{dt} = Ax$$

where $x$ denotes the deviation from the equilibrium $\theta - \theta^{0}$ and

$$A = G_{\varphi\nu}(\vartheta(\theta^{0}), 0) R_{\nu} \frac{\partial}{\partial \theta} G^{T}_{e\varphi}(\vartheta(\theta), 0)$$

### 6.5 An Example of Averaging Analysis

Use of averaging will now be illustrated by analysis of a standard MRAS with adaptation of a feedforward gain and a feedback gain.

Consider an MRAS designed for a process with the nominal transfer function

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

We wish to obtain a closed-loop system with the transfer function

$$G_{m}(s) = \frac{b_{m}}{s + a_{m}}$$

A model-reference adaptive control law was derived in Example 4.6 using Lyapunov theory (see Eq. (4.24)). The closed-loop system is described by
the equations

\[
\frac{d\theta_1}{dt} = -\gamma u_c e \\
\frac{d\theta_2}{dt} = \gamma y e
\]

\[e = y - y_m\]

\[y = G(p)u\]

\[y_m = G_m(p)u_c\]

\[u = -\theta^T \varphi = \theta_1 u_c - \theta_2 y\]  

(6.22)

where \(u_c\) is the reference signal, \(u\) the process input, \(y\) the process output, \(y_m\) the output of the reference model, \(e\) the error, \(\theta_1\) the adjustable feedforward gain, and \(\theta_2\) the adjustable feedback gain. Notice the change in notation compared to Chapter 4, where \(\theta_1\) and \(\theta_2\) were denoted \(t_0\) and \(s_0\) respectively. A block diagram of the system is shown in Fig. 6.3.

It is not possible to give a complete analysis of Eq. (6.22) for general reference signals; approximations must be made even in a simple case like this. The adaptive system described by Eq. (6.22) will now be investigated 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**

For ordinary differential equations it is customary to explore the equilibrium values first, and a similar path will be followed for adaptive sys-
tems. However, for time-varying reference signals there will not be proper equilibria in the sense that all variables are constant. Instead we will investigate equilibria such that the adjustable parameters are constant. It follows from Eq. (6.22) that the parameters $\theta_1$ and $\theta_2$ are constant when the error $e$ is zero. The conditions for $e$ to be zero will now be investigated. When $e = 0$ it follows from Eq. (6.22) that $\theta_1$ and $\theta_2$ are constant. The signal transmission from the command signal $u$ to the output $y$ is then described by the transfer function

$$G_c = \frac{\theta_1 G}{1 + \theta_2 G}$$

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

$$\theta_1 G(i\omega) = \theta_2 G_m(i\omega)G(i\omega) + G_m(i\omega)$$

(6.23)

This is a linear equation, which has a unique solution if $\text{Im}\{G(i\omega)\} \neq 0$. The solution is easily obtained by dividing Eq. (6.23) by $G_m G$ and $G$ respectively, and taking imaginary parts. This gives

$$\theta_1 - \frac{\text{Im}\{1/G(i\omega)\}}{\text{Im}\{1/G_m(i\omega)\}} - \frac{b_m}{b}$$

$$\theta_2 = -\frac{\text{Im}\{G_m(i\omega)/G(i\omega)\}}{\text{Im} G_m(i\omega)} = \frac{a_m - a}{b}$$

(6.24)

These equilibrium values do not depend on the frequency of the command signal. They also correspond to the desired feedback gains. Equation (6.23) is also valid for $\omega = 0$, i.e., when the input is a step. However, since $G(0)$ and $G_m(0)$ are real, the solution is not unique.

**Averaging**

To use the averaging method, it is first observed that the command signal $u_c$ is the only external signal; hence $\nu = u_c$. Further, $\varphi = [-u_c \ y \ T]$. To obtain the averaging equations, the transfer functions $G_{e\nu}$ and $G_{\varphi\nu}$ are first calculated. Straightforward calculations give

$$G_{\nu} = \frac{\theta_1 G}{1 + \theta_2 G} - G_m$$

$$G_{\varphi\nu}^T = \begin{pmatrix} -1 & \frac{\theta_1 G}{1 + \theta_2 G} \end{pmatrix}$$
Figure 6.4 Adaptation gains and their approximation by the averaging method.

The following result is useful when calculating the averages.

**Lemma 6.4—Averaging for sinoidal 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

\[
\text{avg}(vw) = \frac{u_0^2}{2} |G_v(i\omega)||G_w(i\omega)| \cos(\arg G_v(i\omega) - \arg G_w(i\omega))
\]

\[= \frac{u_0^2}{2} \text{Re}(G_v(i\omega)G_w(-i\omega))
\]

**Proof:** The signals $v$ and $w$ have the amplitudes $|G_v(i\omega)|$ and $|G_w(i\omega)|$; their phase angle is $\arg G_v(i\omega) - \arg G_w(i\omega)$.

The averaged equations can now be written as

\[
\frac{d\theta_1}{dt} = -\frac{\gamma u_0^2}{2} \text{Re} \left( \frac{\theta_1 G(i\omega)}{1 + \theta_2 G(i\omega)} - G_m(i\omega) \right)
\]

\[
\frac{d\theta_2}{dt} = \frac{\gamma u_0^2}{2} \text{Re} \left\{ \left( \frac{\theta_1 G(i\omega)}{1 + \theta_2 G(i\omega)} - G_m(i\omega) \right) \frac{\bar{\theta}_1 G(-i\omega)}{1 + \bar{\theta}_2 G(-i\omega)} \right\}
\]

(6.25)

**Example 6.3—Accuracy of averaging**

Consider the particular case of $a = 1$, $b = 2$, and $a_m = b_m = 3$. Let the adaptation gain $\gamma$ be 1 and the command signal $u_c \sin t$. The time histories of the adaptation gains $\theta_1, \theta_2$ and their approximations $\bar{\theta}_1, \bar{\theta}_2$ are shown in Fig. 6.4. 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. 6.5. 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.

Local Stability

The stability of the equilibrium of the averaged equations (Eq. 6.25) will now be investigated. Straightforward but tedious calculations give the following linearized equations:

$$\frac{dx}{dt} = Ax$$

(6.26)

where $x$ is a vector whose two components are the deviations of $\theta_1$ and $\theta_2$ from their equilibrium values, and the matrix $A$ is given by

$$A = \frac{\gamma u_0^2 |G_m|}{2\theta_1} \begin{pmatrix} -\cos \theta_m & |G_m| \cos 2\theta_m \\ |G_m| & -|G_m|^2 \cos \theta_m \end{pmatrix}$$

(6.27)

where $\theta_m = \arctan(\omega/a_m)$. The matrix $A$ has the characteristic equation

$$\lambda^2 + \alpha \lambda (1 + \cos^2 \theta_m) + \alpha^2 \sin^2 \theta_m = 0$$
Table 6.1  Eigenvalues $\lambda_1$, $\lambda_2$ and slopes $k_1$, $k_2$ of the eigenvectors of the matrix $A$ for different frequencies of the command signal.

<table>
<thead>
<tr>
<th>$\omega$</th>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
<th>$k_1$</th>
<th>$k_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-0.667</td>
<td>1.00</td>
<td>-1.00</td>
</tr>
<tr>
<td>1</td>
<td>-0.016</td>
<td>-0.554</td>
<td>1.18</td>
<td>-1.06</td>
</tr>
<tr>
<td>2</td>
<td>0.048</td>
<td>-0.343</td>
<td>2.06</td>
<td>1.26</td>
</tr>
<tr>
<td>3</td>
<td>0.083</td>
<td>-0.167</td>
<td>$\infty$</td>
<td>2.00</td>
</tr>
<tr>
<td>3.22</td>
<td>-0.155</td>
<td>-0.155</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-0.082 $\pm i0.051$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-0.056 $\pm i0.055$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-0.040 $\pm i0.044$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 averaged equation (Eq. 6.26) is thus stable for all $\omega \neq 0$.

**Local Behavior**

Additional insight into the behavior of the system can be gained by analysis of the linearized equations (Eq. 6.26). It follows from Eq. (6.27) that the matrix $A$ which describes the motion of the parameters locally in the neighborhood of the equilibrium, depends on the parameters in a complicated way. The parameters $\gamma$, $u_0$, and $b$ all appear as a scale factor, which determines the convergence rate. The parameters $\omega$ and $a_m$ influence the behavior of the parameters in a more complex way.

The effect of the frequency $\omega$ of the command signal on the properties of the linearized equation (Eq. 6.26) will now be explored. The other parameters are kept at their nominal values, i.e., $\gamma = 1$, $u_0 = 1$, $b = 2$, and $a_m = 3$. The eigenvalues and the slopes of the eigenvectors of the matrix $A$ are given in Table 6.1. The eigenvalues are real if $\omega < 3.22$. This means that the equilibrium is a node and that the solution is of the form

$$x(t) = e_1 e^{\lambda_1 t} + e_2 e^{\lambda_2 t}$$

(6.28)

where $\lambda_1$ and $\lambda_2$ are the eigenvalues and $e_1$ and $e_2$ the corresponding eigenvectors. For $\omega > 3.22$ the eigenvalues are complex, and the equilibrium is a focus.

For $\omega = 0$ the eigenvalues are $\lambda_1 = 0$ and $\lambda_2 = -0.667$, and the corresponding eigenvectors are $e_1 = (1 \ 1)^T$ and $e_2 = (1 \ -1)^T$. The
Figure 6.6 Local behavior of the averaged equations for different frequencies of the command signal. (a) $\omega = 1$; (b) $\omega = 2$; (c) $\omega = 3$; (d) $\omega = 6$.

eigenvectors are thus orthogonal. The solution is not stable. Since $\lambda_1 = 0$, it follows from Eq. (6.28) that the solution will settle somewhere along the slow eigenvector $e_1$.

For $\omega = 1$ the matrix $A$ has the eigenvalues $\lambda_1 = -0.016$ and $\lambda_2 = -0.554$. The corresponding eigenvectors are $e_1 = (1\ 1.18)^T$ and $e_2 = (1\ -1.06)^T$. Since there is a considerable difference between the eigenvalues, it follows from Eq. (6.28) that the solution will approach the subspace spanned by the slow eigenvector $e_1$ and move towards equilibrium at a rate determined by the slow eigenvalue. This eigenvalue corresponds to a time constant of about 60 s, which agrees well with the convergence rate seen in Fig. 6.4. Typical trajectories are shown in Fig. 6.6. As $\omega$ increases, the eigenvalues and the eigenvectors will approach each other as shown in Table 6.1. For $\omega = 3.22$ they will coincide. For larger values of $\omega$ the eigenvalues are complex, and the singularity changes from a node to a focus (see Fig. 6.6).

Table 6.1 shows that the magnitudes of the eigenvalues (and consequently the convergence rate) change significantly with the frequency $\omega$. It appears that the fastest convergence is obtained for frequencies around $\omega = 3.2$ rad/s. This is illustrated in Fig. 6.7, which shows parameter estimates for different $\omega$. The convergence rates agree well with the estimates obtained from the linearized equations. The goodness of the averaging
approximation can also be evaluated from Table 6.1. The approximation will work well if the parameter dynamics are considerably slower than the frequency of the command signal. For $\omega = 1$ the parameter dynamics have poles at $-0.016$ and $-0.554$. The slow pole is considerably slower than the frequency of the input signal. The fast pole ($-0.554$) is, however, of the same order of magnitude as $\omega$. For $\omega = 3$ the parameter dynamics are slower than the frequency by a factor 20. This explains why the approximation works better for $\omega = 3$ than for lower values of $\omega$.

**Summary**

The analysis performed gives significant insight into the behavior of the adaptive system. The parameter equilibria can be predicted, as well as the behavior of the estimates in the neighborhood of the equilibria. This makes it possible to estimate the convergence rate of the parameters. The analysis shows that it is not possible to derive simple rules of thumb for the convergence rate because it depends on the frequency of the command signal in a complicated way. It is also shown that small changes in the frequency of the command signal can cause significant changes in the convergence rate. It has also been observed that the tracking error can be quite small even if the parameters are far from their equilibrium values.
6.6 Robustness

The stability analysis of direct adaptive control in Section 6.2 is based on Assumptions A1 A4 and the premise that there are no disturbances. Assumption A2 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 regulators are based on simplified models. It is therefore of interest to investigate what happens when Assumption A2 is violated. An example illustrates that unexpected phenomena may occur if there are disturbances and if A2 does not hold. Adaptation of feedforward gain is first considered.

Adaptation of a Feedforward Gain

Consider a process with the transfer function \( G(s) \) and an adjustable feedforward gain. Find a feedforward gain \( \theta \) such that the input-output behavior matches the transfer function \( G_m(s) \) as well as possible. The case \( G_m = G \) was discussed in Examples 1.1 and 1.5 and in Chapter 4. Two different algorithms for updating the gain were proposed in Chapter 4: the MIT rule and the SPR rule. The algorithms are

\[
\frac{d\theta}{dt} = -\gamma y_m e \quad \text{(MIT)}
\]

\[
\frac{d\theta}{dt} = -\gamma u_c e \quad \text{(SPR)}
\]

where \( u_c \) is the command signal, \( y_m = \theta^0 G_m u_c \) the model output, and \( e \) the error defined by

\[
e(t) = y - y_m = G(p) (\theta(t)u_c(t)) - \theta^0 G_m(p)u_c(t)
\]

The analysis in Section 4.3 shows that the MIT rule gives a closed-loop system that is globally stable for any adaptation gain \( \gamma \) in the “ideal” use, when \( G = G_m \) and \( G \) is SPR. So far no stability result has been given for the MIT rule. In the presence of unmodeled dynamics, it is, of course, highly unrealistic to assume that a transfer function is SPR. Example 4.4 indicates, however, that the MIT rule will be unstable for sufficiently high adaptation gains if the system is not SPR.

The algorithms will now be investigated under nonideal conditions, using the tools of Section 6.4. Inserting the expressions for \( y_m \) and \( e \) into the equations for the parameters, we get

\[
\frac{d\theta}{dt} + \gamma (G_m u_c) (G(\theta u_c) - \theta^0 G_m u_c) = 0
\]

\[
\frac{d\theta}{dt} + \gamma u_c (G(\theta u_c) - \theta^0 G_m u_c) = 0
\]
where the first equation holds for the MIT rule and the second for the SPR rule. The corresponding averaging equations are

\[
\frac{d\theta}{dt} + \gamma \left( \theta \text{avg}\{(G_m u_c)(G u)\} - \theta^0 \text{avg}\{(G_m u_c)^2\} \right) = 0
\]

\[
\frac{d\theta}{dt} + \gamma \left( \theta \text{avg}\{u_c(G u_c)\} - \theta^0 \text{avg}\{u_c(G_m u_c)\} \right) = 0
\]

The equilibrium parameters are

\[
\theta_{\text{MIT}} = \theta^0 \frac{\text{avg}\{(G_m u_c)^2\}}{\text{avg}\{(G_m u_c)(G u_c)\}}
\]

\[
\theta_{\text{SPR}} = \theta^0 \frac{\text{avg}\{u_c(G_m u_c)\}}{\text{avg}\{u_c(G u_c)\}}
\]

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 (Eq. 6.32) are

\[
\gamma \text{avg}\{(G_m u)(G u)\} > 0 \quad \text{(MIT)}
\]

\[
\gamma \text{avg}\{u_c(G u_c)\} > 0 \quad \text{(SPR)}
\]

The MIT rule 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, for example, the case of a command signal composed of a constant and a sum of sinusoids:

\[
u_c(t) = a_0 + 2 \sum_{k=1}^{n} a_k \sin \omega_k t
\]

For positive \( \gamma \) the stability conditions become

\[
a_0^2 G_m(0) + \sum_{k=1}^{n} a_k^2 |G_m(i\omega_k)||G(i\omega_k)| \cos \{\arg G_m(i\omega_k) - \arg G(i\omega_k)\} \geq 0
\]

\[
a_0^2 G(0) + \sum_{k=1}^{m} a_k^2 \text{Re} G(i\omega_k) > 0
\]
The MIT rule gives a stable equilibrium if the phase lags of \( G_m \) and \( G \) differ at most by 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°. 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, where 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 Eq. (6.31) that the convergence rate of the parameters is strongly signal-dependent. The value of normalization as described in Section 4.3 is that the convergence rate becomes less dependent on the signal amplitudes. The calculations are illustrated by an example.

**Example 6.4 Sinusoid 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 frequency \( \omega \). Equation (6.32) gives the equilibrium values

\[
\theta_{\text{MIT}} = \frac{b^2 + \omega^2}{b^2}
\]

\[
\theta_{\text{SPR}} = \frac{b^2 + \omega^2}{b(ab - \omega^2)} \quad \omega < \sqrt{ab}
\]

The stability conditions show that the MIT rule is stable for all \( \omega \), but the SPR rule is stable only if \( \omega < \sqrt{ab} \). Figure 6.8 shows the behavior of the systems for \( a = 1 \) and \( b = 10 \) when the input signals have frequencies \( \omega = 3 \) 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 behavior is well predicted by the averaging analysis. Notice the difference in convergence rates. Initially, when \( \theta = 0 \), the rates of changes are given by

\[
\dot{\theta}_{\text{MIT}} = \gamma \theta^0 \text{avg}\{(G_m u_c)^2\}
\]

\[
\dot{\theta}_{\text{SPR}} = \gamma \theta^0 \text{avg}\{u_c(G_m u_c)\}
\]
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 $G_m u_c$.

Also notice the drastic difference in the input-output behavior with the equilibrium values of the parameters. For $\omega = 3$, $a = 1$, and $b = 10$, the MIT rule gives an equilibrium gain of 1.09, while the SPR rule gives a gain of 10.9. The unmodeled dynamics reduce the gain by $\sqrt{109}/10 = 1.04$ and introduce a phase lag of $17^\circ$ at $\omega = 3$.

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 the 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 the
presence of unmodeled dynamics the gain must be kept small to maintain stability.

**Analysis of a First-order MRAS**

The consequences of unmodeled dynamics will now be investigated for an MRAS that adjusts one feedforward gain and one feedback gain. The system shown in Fig. 6.3 will be investigated. This system was designed based on the assumption that the transfer function of the process has the form

\[ G(s) = \frac{b}{s + a} \]

We will investigate what happens if the process actually has more poles and zeros. Before going into details, a specific example will be investigated.

**Example 6.5—Unmodelled dynamics**

Assume that the nominal transfer function has \( a = 1 \) and \( b = 2 \), but the actual transfer function is

\[ G(s) = \frac{458}{(s + 1)(s^2 + 30s + 229)} \quad (6.33) \]

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 have been neglected in the model used to design the adaptive controller. Figure 6.9 shows the behavior of the parameters when the command signal is a step and there is a sinusoidal measurement error. Figure 6.10 shows the behavior of the parameters when the command signal is sinusoidal with different frequencies. The behavior shown in the figures is clearly not acceptable. We will therefore attempt to understand what happens and to find suitable remedies.

The example shows that the presence of unmodeled dynamics will drastically change the behavior of the adaptive system. Figure 6.10 shows that the equilibrium depends on the frequency of the command signal and that it may be unstable for certain frequencies. Assumption A2 of Theorem 6.1 is thus critical. We will now attempt to understand the mechanisms that change the behavior of the system so drastically.

**Step Commands**

First, the behavior illustrated in Fig. 6.9 will be analyzed. The case of step commands will first be investigated when there is no measurement noise. Possible equilibria will be determined. The equations describing the system are given by Eq. (6.22), from which it follows that the parameters are constant if \( e = 0 \). The analysis for a sinusoidal command signal in
Figure 6.9 Adjustable gains $\theta_1$ and $\theta_2$ when the adaptive control law of Eq. (6.22) is applied to the process of Eq. (6.33). The command signal is a step and there is sinusoidal measurement noise. The smooth curves show the behavior when there is no measurement noise.

Section 6.5 can be applied simply by observing that a step corresponds to $\omega = 0$. The equilibrium condition of Eq. (6.23) reduces to

$$\theta_2 = \frac{1}{G_m(0)} \theta_1 - \frac{G_m(0)}{G(0)} = \theta_1 - 0.5 \quad (6.34)$$

The equilibrium set is thus a straight line in parameter space. The line is uniquely determined by the steady-state gain $G(0)$ of the system. 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 $\theta_1$ and $\theta_2$. However the excitation used is a step that is persistently exciting of first order and thus admits determination of only one parameter.

Averaging will now be applied in order 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 $\theta_1$ and $\theta_2$ are constant. The closed-
6.6 Robustness

Figure 6.10 Adjustable gains \( \theta_1 \) and \( \theta_2 \) when the adaptive control law of Eq. (6.22) is applied to the process of Eq. (6.33) when the command signal is \( \sin \omega t \), with (a) \( \omega = 1 \); (b) \( \omega = 3 \); (c) \( \omega = 6 \); (d) \( \omega = 20 \).

loop eigenvalues are the zeros of the equation

\[
1 + \theta_2 G(s) = 0
\]

A necessary condition for stability is that \( 1 + \theta_2 G(s) = 0 \). 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 closed-loop system will typically become unstable when \( \theta_2 \) is sufficiently large.

Example 6.6—Step commands

With the transfer function of Eq. (6.33) used in Example 6.5, 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

\[-0.5 < \theta_2 < 17.03\]
The averaged equations for the parameter estimates are obtained by setting $\omega = 0$ in Eq. (6.25). If it is assumed that $G_m(0) = 1$, the equation becomes

\[
\frac{d\theta_1}{dt} = -\gamma u_0^2 \left( \frac{\theta_1 G(0)}{1 + \bar{\theta}_2 G(0)} - 1 \right)
\]

\[
\frac{d\bar{\theta}_2}{dt} = \gamma u_0^2 \frac{\theta_1 G(0)}{1 + \bar{\theta}_2 G(0)} \left( \frac{\bar{\theta}_1 G(0)}{1 + \bar{\theta}_2 G(0)} - 1 \right)
\]  \hspace{1cm} (6.35)

These differential equations have the equilibrium set of Eq. (6.34).

Close to the equilibrium set, the equations are described by the following linearized equations:

\[
\frac{dx}{dt} = \frac{\gamma u_0^2}{\theta_1} \begin{pmatrix}
-1 & 1 \\
1 & -1
\end{pmatrix} x
\]  \hspace{1cm} (6.36)

where $x_1 = \theta_1 - \theta_1^0$ and $x_2 = \theta_2 - \theta_2^0$. Consider a point away from the equilibrium line, i.e., $x_2 = x_1 + \delta$ or $\theta_2 = \theta_1 - 1/G(0) + \delta$. The velocity of the state vector at that point is $\dot{x}_1 = \gamma u_0^2 \delta / \theta_1$, $\dot{x}_2 = -\gamma u_0^2 \delta / \theta_1$. The vector field of the linearized equation is thus as shown in Fig. 6.11. The vector field thus pushes the parameter towards the equilibrium for $\theta_1 > 0$ and away from the equilibrium for $\theta_1 < 0$.

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. (6.34), the averaged equations (Eq. 6.35) can be divided to give

\[
\frac{d\bar{\theta}_2}{d\theta_1} = -\frac{G(0)\theta_1}{1 + \theta_2 G(0)}
\]
6.6 Robustness

Figure 6.12 Parameter trajectories (a) in the nominal case of $G(s) = 2/(s + 1)$ and (b) in the case of unmodeled dynamics.

This differential equation has the solution

$$\theta_2^2 + \frac{2}{G(0)} \theta_2 + \theta_1^2 = \text{const}$$

The parameters of the averaged equations will thus move along circular paths, with the center in $(0, -1/G(0))$. The motion is clockwise for $\theta_2 > \theta_1 + 1/G(0)$ and counter clockwise for $\theta_2 < \theta_1 + 1/G(0)$. The motion slows down and stops when the parameters reach the equilibrium set

$$\{\theta_1, \theta_2 | \theta_1 > 0, \theta_2 = \theta_1 + 1/G(0)\}$$

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 $-1/G(0) < \theta_2$. In the case of unmodeled dynamics, the stability region is defined by $-1/G(0) < \theta_2 < \theta_2^0$. This means that trajectories that start far away from the origin will escape from the stability region. Figure 6.12 shows the actual parameter paths in the nominal case and for the unmodeled dynamics given by the transfer function of Eq. (6.33) in Example 6.5. 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 $\gamma$ sufficiently small. The “jitter” in the trajectories in Fig. 6.12(b) is oscillations in the parameters, not numerical errors.

Measurement Noise

The effects of measurement noise will now be investigated. The simulation shown in Fig. 6.10 indicates that measurement noise may cause the
parameters to drift. As a starting point the simulation results of Fig. 6.9 will be represented in a different way. Figure 6.13 shows parameter $\theta_2$ as a function of parameter $\theta_1$. For comparison the simulation has also been executed without measurement noise, and the simulation indicates that the equilibrium is lost in the presence of measurement noise. The parameters will move towards 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 Eq. (6.22) that the error cannot be made identically zero by proper choice of the parameters. Hence there exists no true equilibrium such that the parameters are constant.

Possible equilibria of the averaged equations will now be investigated. To analyze the fast mode, assume that the regulator parameters are constant. Simple calculations show that the Laplace transform of the process output is given by

$$Y(s) = \frac{\theta_1 G(s)}{1 + \theta_2 G(s)} U_c(s) + \frac{1}{1 + \theta_2 G(s)} N(s)$$
where $U_c$ and $N$ are the Laplace transforms of the command signal and the measurement noise. If parameter $\theta_2$ is such that the system is stable, we get (after an exponentially decaying transient)

$$y(t) = \frac{\theta_1 G(0)}{1 + \theta_2 G(0)} u_0 + w(t)$$

The first term corresponds to the steady-state response to the command signal. The second term is the response to the measurement noise. The parameter motion is governed by Eq. (6.22). Assuming that $\gamma$ is small, so that the parameters change much more slowly than the measurement noise, the following averaged equations are obtained:

$$\frac{d\theta_2}{dt} = -\gamma u_0^2 \left( \frac{\theta_1 G(0)}{1 + \theta_2 G(0)} - 1 \right)$$

$$\frac{d\theta_1}{dt} = \gamma u_0^2 \frac{\theta_1 G(0)}{1 + \theta_2 G(0)} \left( \frac{\theta_1 G(0)}{1 + \theta_2 G(0)} - 1 \right) + \gamma v(\theta_2) \tag{6.37}$$

where

$$v(\theta_2) = \text{avg}\{w^2(t)\}$$

Notice that $v$ depends on $\theta_2$. Compare with Eq. (6.35). Setting the derivatives equal to zero, we find that the averaged equations do not have a solution if $v \neq 0$. This means that the averaged equations do not have an equilibrium in the presence of measurement noise. The behavior of the system is thus changed drastically. Equation (6.37) will be linearized to get more insight into the behavior of the system. Linearizing around the equilibrium set $\theta_2 = \theta_1 - 1/G(0)$, for the case $v = 0$ we get

$$\frac{dx}{dt} = \gamma u_0^2 \theta_1 \left( \begin{array}{cc} -1 & 1 \\ 1 & -1 \end{array} \right) x + \left( \begin{array}{c} 0 \\ \gamma v(\theta_2) \end{array} \right)$$

where $x_1 = \theta_1 - \theta_1^0$ and $x_2 = \theta_2 - \theta_2^0$. Addition and subtraction of the equation give

$$\frac{d}{dt} (x_1 - x_2) = -\frac{2\gamma u_0^2}{\theta_1} (x_1 - x_2) - \gamma v(\theta_2)$$

$$\frac{d}{dt} (x_1 + x_2) = \gamma v(\bar{\theta}_2)$$

If the mean square measurement noise is small, the behavior is as follows. For $\theta_1 > 0$ the parameters will move towards the line

$$x_2 = x_1 + \frac{\theta_1 v(\theta_2)}{2u_0^2}$$
or

\[
\theta_2 = \theta_1 - \frac{1}{G(0)} + \frac{\theta_1 v(\bar{\theta}_2)}{2u_0^2}
\]

The motion will then slowly drift along this line at the rate \(\gamma v(\bar{\theta}_2)\). The phase portrait of the averaged equations is shown in Fig. 6.14.

It is now clear what happens. The parameters will move towards the set, and when they come close to it, they will start to drift along the set. The gains will increase. In the presence of unmodeled dynamics, the system will finally become unstable. Notice the significant qualitative differences between the cases \(n = 0\) and \(n\) small. The analysis also gives quantitative results, as illustrated by the following example.

**Example 6.7—Sinusoidal measurement noise**

Consider the system in Example 6.5. Assume that the measurement noise is \(n(t) = n_0 \sin \omega_0 t\), with \(n_0 = 0.5\) and \(\omega_0 = 10\), and that the adaptation gain is \(\gamma = 0.1\). Hence \(v = n_0^2/2 = 0.125\) and Eq. (6.38) becomes

\[
\theta_2 = 1.0625\theta_1 - 0.5
\]

The drift rate along the set is \(\gamma n_0^2/2 = 0.0125\), which means that the average rate of change of the parameters is 0.00625. This agrees well with
the simulations in Fig. 6.9. The amplitude of the fluctuation is also easily estimated from Eq. (6.37). The amplitude is \( \gamma n_0/\omega_0 = 0.005 \) which agrees well with Fig. 6.9.

Intuitively, the results can be explained as follows. A step input is only persistently exciting of order 1 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.

**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 only one parameter 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. 6.10 indicates that there are some problems with sinusoidal command signals in combination with unmodeled dynamics.

As before, it is assumed that the adaptive regulator 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 has the transfer function

\[
G(i\omega) = \frac{b}{a + i\omega} r(\omega)e^{-i\vartheta(\omega)}
\]

The functions \( r \) and \( \vartheta \) represent the distortions of amplitude and phase due to unmodeled dynamics. It is assumed that the transfer function corresponding to \( r \) and \( \vartheta \) 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 \( \vartheta(\omega) + \tan^{-1}(\omega/a) \). This lag is \( \pi \) if

\[
\frac{\omega}{a} = \tan(\pi - \vartheta(\omega)) = -\tan \vartheta(\omega)
\]
or

\[ \omega \cos \theta(\omega) + a \sin \theta(\omega) = 0 \quad (6.40) \]

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

\[ \theta_2 = \bar{\theta}_2^0 = \frac{b r(\omega)}{\sqrt{a^2 + \omega^2}} \quad (6.41) \]

where \( \omega \) is the smallest value that satisfies Eq. (6.40).

**Equilibrium Analysis**

The possible equilibria of the parameters will first be determined, as
given by Eq. (6.40). Introducing the transfer function of Eq. (6.39) into
Eq. (6.24) gives (after straightforward but tedious calculations)

\[
\begin{align*}
\theta_1 &= \frac{b_m(a \sin \theta(\omega) + \omega \cos \theta(\omega))}{\omega b r(\omega)} \\
\theta_2 &= \frac{\omega(a_m - a) \cos \theta(\omega) + (\omega^2 + aa_m) \sin \theta(\omega)}{\omega b r(\omega)} \\
&= \frac{1}{b r(\omega)} \left( (\omega \sin \theta(\omega) - a \cos \theta(\omega)) + \frac{a_m}{\omega} (a \sin \theta(\omega) + \omega \cos \theta(\omega)) \right) \quad (6.42)
\end{align*}
\]

A comparison with Eq. (6.24) 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

\[ \omega \cos \theta(\omega) + a \sin \theta(\omega) = 0 \]

Comparing with Eq. (6.40), we find that this is precisely the frequency
where the process has a phase lag of 180°. The feedback gain for this
frequency is

\[ \theta_2 = \frac{1}{b r(\omega)} (\omega \sin \theta - a \cos \theta) = \frac{\sqrt{a^2 + \omega^2}}{b r(\omega)} \]

This implies that \( \theta_2|G(i\omega)| = 1 \), i.e., 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 such that the closed-loop system is unstable. This observation is illustrated by an example.

Example 6.8—Sinusoidal command signal
Consider the system in Example 6.5. 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 regulator gains are

\[ \theta_1 = \frac{3(259 - \omega^2)}{458} \]
\[ \theta_2 = \frac{2(137 + 7\omega^2)}{229} \]

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 regulator gains are \( \theta_1 = 0 \) and \( \theta_2 = 3900/229 = 17.03 \). The closed-loop system is unstable for this feedback gain. This explains the results shown in Fig. 6.10. It was required that the closed-loop response should be \( 3/(s + 3) \). At \( \omega = 3 \) the unmodeled dynamics correspond to \( r = 0.96 \) and \( \vartheta = 22° \). The unmodeled dynamics thus give an extra phase lag of 22°. In view of this, it is not surprising that a first-order model is marginal. The performance of the system is still quite reasonable (see Fig. 6.15).

A System Identification Viewpoint
The connection between direct and indirect adaptive control laws was discussed in Section 5.2. Additional insight into the robustness problem is obtained by interpreting the MRAS as an indirect scheme. With fixed controller parameters the closed-loop transfer function for the nominal system is

\[ \frac{\theta_1 G}{1 + \theta_2 G} = \frac{\theta_1 b}{s + a + \theta_2 b} \]
Requiring that this transfer function is equal to the desired closed-loop transfer function \( G_m = b_m/(s + a_m) \), we get

\[
\theta_1 = \frac{b_m}{b} \\
\theta_2 = \frac{a_m - a}{b}
\]  

The MRAS can thus be interpreted as if parameters \( a \) and \( b \) of the first-order model \( b/(s + a) \) were estimated and the regulator gains computed from Eq. (6.43). Solving Eq. (6.43) for the process parameters gives the estimates

\[
\hat{a} = a_m - \frac{b_m \theta_2}{\theta_1} \\
\hat{b} = \frac{b_m}{\theta_1}
\]

To investigate what happens with unmodeled dynamics, the expression in Eq. (6.42) for the regulator parameters is inserted in the above equations. This gives

\[
\hat{a} = \frac{\omega(a \cos \vartheta - \omega \sin \vartheta)}{a \sin \vartheta + \omega \cos \vartheta} \\
\hat{b} = \frac{\omega br}{a \sin \vartheta + \omega \cos \vartheta}
\]  

(6.44)

With no unmodeled dynamics \( (r = 1 \text{ and } \vartheta = 0) \), it follows that \( \hat{a} = a \) and \( \hat{b} = b \) for all frequencies. With unmodeled dynamics the estimates obtained depend on the unmodeled dynamics and on the frequency of the command signals. This is natural, because a first-order model is fitted to a complex transfer function. With a sinusoidal input it is possible to match at any frequency; the parameter estimates will then depend on the frequency.

Notice that \( \hat{a} \) is zero if

\[
\tan \vartheta = \frac{a}{\omega}
\]

The phase lag of the plant is then

\[
\vartheta + a \tan \frac{\omega}{a} = a \tan \frac{\omega}{a} + a \tan \frac{\omega}{a} = \frac{\pi}{2}
\]

This is logical, because the only way to obtain a phase lag of \( \pi/2 \) is to have \( a = 0 \). Also notice that the estimates become infinite if \( \tan \vartheta = -\omega/a \). This means that the phase lag of the plant is

\[
\vartheta + a \tan \frac{\omega}{a} = \pi - a \tan \frac{\omega}{a} + a \tan \frac{\omega}{a} = \pi
\]
Figure 6.15  Process output and model output of the adaptive system when $u_c(t) = \sin 3t$.

This is natural, because the only way to obtain a phase lag of $\pi$ is to have $a$ and $b$ infinitely large and with opposite signs.

Summary of Examples

The investigation of the first-order MRAS is summarized in the table below.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Exact Model Structure</th>
<th>Unmodeled Dynamics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step command</td>
<td>Equilibrium set is a half line.</td>
<td>Equilibrium set is a line segment.</td>
</tr>
<tr>
<td>Step command measurement noise</td>
<td>Solution will move towards a line and then drift along the line.</td>
<td>Solution will move towards a line and drift along the line until stability is lost.</td>
</tr>
<tr>
<td>Sinusoidal</td>
<td>Equilibrium set is a point that is independent of the frequency.</td>
<td>Equilibrium set is a point that depends on the frequency. The equilibrium is unstable for sufficiently high frequency.</td>
</tr>
</tbody>
</table>

Several interesting conclusions can be drawn from the examples. From a methodological point of view they show the 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 there exists a true equilibrium, in the sense that the parameters remain constant. Notice that the averaged equations may have an equilibrium even if the exact equations do not. It is, however, rarely the case that global analysis can be carried out.

The example also shows that Theorem 6.1, although it is of significant theoretical interest, has limited practical value. The theorem will clearly not hold if Assumption A2 is violated. This assumption will not hold in a practical case, in which there are always unmodeled dynamics. It is also not realistic to neglect disturbances. This raises the possibility that global stability can be established only under unrealistic assumptions.

Theorem 6.1 also gives poor guidelines for the choice of regulator complexity. To satisfy Assumption A2 it seems logical to increase the regulator complexity. This will, however, impose additional requirements on the input signal to maintain persistency of excitation.

**Improved Algorithms**

The examples indicate that the MRAS algorithm in Eq. (6.22) is incapable of dealing with unmodeled dynamics and disturbances. The insight given by the analysis also suggests various improvements of the algorithms.

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

**Projections, Leakage, and Dead Zones**

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

Several ideas have been proposed to modify the adaptive algorithms so as to avoid the difficulty. One possibility is to modify the algorithm so that the parameters are projected into a given fixed set. This will however
require that appropriate prior knowledge is available. For example, in Example 6.6 it is sufficient to project into a set such that $0 \leq \theta_2 \leq 17$. A convenient way to obtain a regulator with a finite gain is to provide a path parallel to the process with gain $\rho$. Compare Fig. 5.17. Let $G_R$ be the transfer function of the regulator. The arrangement with the parallel path is equivalent to a regulator with the transfer function

$$G_R^o = \frac{\rho G_R}{\rho + G_R}$$

This is clearly bounded by $\rho$ when $G_R$ has high gain.

Another possibility is to modify the parameter by updating the algorithm of Eq. (6.18) to

$$\frac{d\theta}{dt} = \gamma \varphi e + \alpha (\theta^0 - \theta)$$ (6.45)

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

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

$$\frac{d\theta}{dt} = \gamma \varphi e + \alpha |e| (\theta^0 - \theta)$$ (6.46)

A third way to avoid the difficulty is to switch off the parameter estimation if the input signal is not appropriate. There are several ways to determine when the estimates should be switched off. A simple way is to update only when the error is large i.e. to introduce a dead zone in the estimator. Such an approach is actually suggested by Theorem 6.2. However, it is necessary to have prior knowledge to select the dead zone.

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

**Filtering and Monitoring of Excitation**

From the system identification point of view the problem of unmodeled dynamics can be interpreted as follows. When fitting a low-order model to a system with complex dynamics, the results will depend critically on
the frequency content of the input signal. Precautions must thus be taken to ensure that the frequency content of the input signal is concentrated to the frequency range where the simple model is expected to fit well. This indicates that the signals should be filtered before they are entered into the parameter estimator or the parameter update law. However, filtering alone is not sufficient, since it may happen that the input signal only has frequencies outside the useful frequency range. (A typical case is the system in Example 6.8 with \( u_c(t) = \sin 16.09t \).) No amount of filtering can remedy such a situation. We are then left with only two options: to switch off the estimation or to introduce intentional perturbation signals.

**Normalization**

Various modifications of the adaptive algorithm will be discussed in more detail in Chapter 9. Theoretical analysis of the schemes is at the front line of research still a bit too complicated (and lengthy) to be included in this book.

Only a few sketchy remarks will therefore be given. Notice that Theorem 6.2 gives stability conditions for adaptive control applied to the model

\[
Ay = Bu + v
\]  
(6.47)

where \( v \) is a bounded disturbance. Unmodeled dynamics can, of course, be modeled by Eq. (6.47), but \( v \) will no longer be bounded, since it depends on the inputs and outputs. By introducing the signal defined by

\[
Cr(t) = \max (|u(t)|, |y(t)|)
\]

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

\[
\tilde{y} = \frac{y}{r}, \quad \tilde{u} = \frac{u}{r}, \quad \tilde{v} = \frac{v}{r}
\]

the model of Eq. (6.47) can be replaced by

\[
A\tilde{y} = B\tilde{u} + \tilde{v}
\]

where \( \tilde{v} \) now is bounded. By invoking Theorem 6.2, it can be established that adaptive control with a dead zone or projection gives a system with bounded signals. The detailed justification is complicated.

**Summary**

This section has analyzed the properties of a simple MRAS in the presence of unmodeled dynamics and disturbances. Examples of complicated behavior were presented, and it was shown that considerable insight could
be gained by using tools such as equilibrium analysis and averaging. The analysis revealed that the assumptions in Theorem 6.1 are restrictive. The analysis also suggested several ways to improve the algorithms by introducing leakage, filtering, dead zone, monitoring of excitation conditions, intentional perturbation signals, and normalization.

### 6.7 Stochastic Averaging

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 will be used on discrete time systems with stochastic inputs. Assume that the system is described by

\[
A^*(q^{-1})y(t) = B^*(q^{-1})u(t - d) + C^*(q^{-1})e(t)
\]  

(6.48)

where \( e(t) \) is zero mean Gaussian stochastic process. Depending on the specifications, different self-tuning regulators can be used to control the system (compare Chapter 5). For simplicity it is assumed that the basic direct self-tuning algorithm (Algorithm 5.4) is used. The controller parameters are then estimated from a model of the form

\[
y(t) = R^*(q^{-1})u(t - d) + S^*(q^{-1})y(t - d)
\]  

(6.49)

or

\[
y(t) = \varphi(t - d)^T \theta
\]

(6.50)

The parameters \( \theta \) are estimated using the recursive least-squares method. When applying averaging, it is appropriate to use the form

\[
\theta(t) = \theta(t - 1) + \gamma(t)R(t)^{-1}\varphi(t - d)(y(t) - \varphi^T(t - d)\theta(t - 1))
\]

\[
R(t) = R(t - 1) + \gamma(t)(\varphi(t - d)\varphi^T(t - d) - R(t - 1))
\]

(6.51)

where the covariance matrix \( P(t) \) is related to \( R(t) \) through

\[
P(t) = \gamma(t)R(t)^{-1}
\]

and

\[
\gamma(t) = \frac{1}{t}
\]

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

\[
r(t) = r(t - 1) + \gamma(t)(\varphi(t - d)^T\varphi(t - d) - r(t - 1))
\]

(6.52)
The controller is
\[ u(t) = -\frac{S^*(q^{-1})}{R^*(q^{-1})} y(t) \]  
(6.53)
or
\[ \varphi(t)^T \theta(t) = 0 \]

The self-tuning regulator is described by Eqs. (6.50) and (6.51). The control law of Eq. (6.53) is then used on the system of Eq. (6.48). 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. Using the averaging idea, it is possible to derive an associated deterministic differential equation. The convergence properties of the algorithm can then be determined using these equations. The method was suggested by Ljung (1977) and is sometimes called the ODE approach (Ordinary Differential Equation approach). Only a heuristic derivation and motivation is 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 Eq. (6.51) is small, and the correction in \( \theta(t) \) is small. As in Section 6.4, we can separate the states from the parameters and assume that the parameters are constant when evaluating the behavior of the closed-loop system. Both \( R(t) \) and \( \varphi(t) \) depend on the parameter estimates. Since \( \theta \) is assumed to change slowly, the behavior of the model can be approximated by

\[ y(t) = \varphi^T(t - d) \theta \]

where \( \theta \) is the averaged value of the estimates. Also, \( \varphi \) depends on the estimated variables through the feedback. The updating equation for \( R \) can be approximated by

\[ R(t) = R(t - 1) + \gamma(t)(G(\theta) - R(t - 1)) \]  
(6.54)

where

\[ G(\theta) = E \{ \varphi(t - d, \bar{\theta}) \varphi^T(t - d, \theta) \} \]  
(6.55)

The expectation is taken with respect to the underlying stochastic process in Eq. (6.48) and evaluated for the fixed value of the parameters \( \theta \). In the same way the parameter update is approximated by

\[ \theta = \theta(t - 1) + \gamma(t)R(t)^{-1} f(\theta) \]  
(6.56)
where
\[ f(\theta) = E \{ \varphi(t - d, \theta)(y(t) - \varphi^T(t - d, \theta)\theta) \} \quad (6.57) \]

Equations (6.56) and (6.54) 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
\[ \theta(t') - \theta(t) + \Delta \tau \bar{R}(t)^{-1} f(\theta(t)) \]
\[ R(t') = R(t) + \Delta \tau \left( G(\bar{\theta}(t)) - \bar{R}(t) \right) \]

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
\[ \frac{d\theta}{d\tau} = R(\tau)^{-1} f(\theta(\tau)) \quad (6.58) \]
\[ \frac{dR}{d\tau} = G(\theta(\tau)) \quad R(\tau) \quad (6.59) \]

If stochastic approximation is used, Eq. (6.59) is replaced by
\[ \frac{dr}{d\tau} = g(\theta(\tau)) - r(\tau) \]

where
\[ g(\theta) = E \{ \varphi^T(t - d)\varphi(t - d) \} \]

and \( R \) is replaced by \( r \) in Eq. (6.58). These equations are called the associated ordinary differential equations to Eqs. (6.51) and (6.52). 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. (6.59) and (6.58) have been heuristic. However, it can be rigorously shown that, provided the estimates \( \theta(t) \) are “sufficiently often in the domain of attraction of the associated differential equations, then

- Only stable stationary points of Eqs. (6.58) and (6.59) are possible convergence points for the estimates, and
- The trajectories \( \theta(\tau) \) are the “asymptotic paths” of the estimates \( \theta(t) \).
The associated ODE can be used to find possible convergence points of an adaptive algorithm, $\theta^0$ and $R^0$. The equations can then be linearized around these stationary points. It is easily seen that the linearized equations are

$$
\frac{d}{dt}\begin{pmatrix} \theta & \theta^0 \\ R & R^0 \end{pmatrix} = \begin{pmatrix} G(\bar{\theta})^{-1} \frac{\partial f(\theta)}{\partial \theta} & 0 \\ X & -I \end{pmatrix} \begin{pmatrix} \theta & \theta^0 \\ R & R^0 \end{pmatrix}
$$

where the element $X$ is not important for the local stability. The stationary point is thus stable if the matrix

$$
K = G(\bar{\theta})^{-1} \frac{\partial f(\theta)}{\partial \theta} \bigg|_{\theta = \theta^0}
$$

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 $\varepsilon(t) = y(t) - \varphi(t - d)^T \theta$ for a fixed value of $\theta$.
2. Compute the expected values $G(\theta)$ and $f(\theta)$.
3. Determine possible convergence points for Eqs. (6.58) and (6.59) and determine the local stability properties using Eq. (6.60).
4. Simulate the equations.

Even if Eqs. (6.58) and (6.59) 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 5.4 with $A_o^* = A_m^* = 1$). Let the algorithm be applied to a system described by Eq. (6.48). 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 (1977a)). 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 6.9  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

\[ y(t) - s_0 y(t - 1) = u(t - 1) + r_1 u(t - 2) \]

and use the controller

\[ u(t) = -s_0 y(t) - r_1 u(t - 1) \]

The closed-loop system is described by

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

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

In this case

\[ \varphi^T(t - 1) = \begin{bmatrix} u(t - 2) \\ y(t - 1) \end{bmatrix} \]

and

\[ \varepsilon(\ell) = y(t) \]

Thus

\[ f(\theta) = \begin{bmatrix} r_{yu}(2) \\ r_{y}(1) \end{bmatrix} \]

and

\[ G(\theta) = \begin{bmatrix} r_u(0) & r_{yu}(1) \\ r_{yu}(1) & r_y(0) \end{bmatrix} \]

where \( r_y(\tau), r_u(\tau), \) and \( r_{yu}(\tau) \) are the covariance functions of \( y \) and \( u \) and the cross-covariance between \( y \) and \( u \). These functions can be computed, since the systems are only of second order.

The stationary point is given by \( f(\theta) = 0 \), which gives \( r_{yu}(2) = 0 \) and \( r_y(1) = 0 \). This is exactly the result obtained in Theorem 5.2. Figure 6.16(a) shows the phase plane of the ODE when stochastic approximation is used. The stationary point corresponds to the minimum-variance regulator, and the triangle indicates the stability boundary for the closed-loop system. Figure 6.16(b) shows realizations of the estimates \( s_0 \) and \( r_1 \).
when stochastic approximation has been used. The estimator is started with a very small step size. The realizations follow the same gross behavior as the trajectories of the ODE.

Example 6.10 Moving-average self-tuner
Consider an integrator with a time delay $\tau$. (Compare Example 5.4.) For the sampling period $h > \tau$, the system is described by

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

where

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

Moving-average controllers of different orders will now be analyzed (see Section 5.3). The stability conditions are that the sampled system is nonminimum-phase (i.e., $|b| < 1$) and that all eigenvalues of $K$ have negative real parts.

Case 1 ($d = 1$)

The minimum-variance strategy is obtained through

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

Hence

$$
R(z) = z + b \\
S(z) = \frac{1 + c}{h - \tau} z
$$
The characteristic equation of $K$ in Eq. (6.60) is in this case

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

Since $b$ and $c$ are both less than 1, follows that the eigenvalues of $K$ are both negative. The condition $b < 1$ implies that $\tau < h$. 2.

**Case 2** \((d = 2)\)

Since $B$ is of first order, and $C$ of second order, we get the following combinations.

<table>
<thead>
<tr>
<th>Case</th>
<th>$B^+$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2(a)</td>
<td>$z + b$</td>
<td>Minimum variance</td>
</tr>
<tr>
<td>2(b)</td>
<td>$z + b$</td>
<td>Deadbeat</td>
</tr>
<tr>
<td>2(c)</td>
<td>1</td>
<td>Moving average</td>
</tr>
</tbody>
</table>

To investigate the equilibria, first notice that Cases 2(a) and 2(b) can give stable equilibria only if $b < 1$ (i.e., $\tau < h/2$). Case 2(a) corresponds to the minimum-variance controller. The characteristic equation of the matrix $K$ is

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

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

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

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

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

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

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

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

Summarizing, we find that if $d = 1$, there is only one equilibrium, which corresponds to the minimum-variance control. This equilibrium is locally stable only if $\tau < h/2$. When $d = 2$, there are three equilibria, corresponding to Cases 2(a), 2(b) and 2(c). Equilibrium 2(a) is always unstable; equilibrium 2(b) is stable if $b < c$; and 2(c) is stable if $b > c$. 
Figure 6.17  Simulation of the ODEs of the parameter estimates for the integrator when \( d = 2 \) and \( c = -0.8 \). (a) \( \tau = 0.6 \). (b) \( \tau = 0.4 \).

The phase portraits of the ODEs associated with the algorithm are shown in Fig. 6.17 for the case \( d = 2 \) and \( c = -0.8 \). When \( \tau = 0.4 \), there are three equilibria. They correspond to Case 2(a), which is a saddle point, Case 2(b), which is an unstable focus, and Case 2(c), which is a stable node. The stable node corresponds to the moving-average controller. The parameters are \( r_1 = 0.08 \) and \( s_0 = 0.20 \). For \( \tau = 0.6 \) there is only one equilibrium, which corresponds to the moving average controller with the parameters \( r_1 = 0.12 \) and \( s_0 = 0.20 \). It is also seen in Fig. 6.17 that there exist starting points for which the algorithm does not converge. The estimates are driven towards the stability boundary.

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

**Example 6.11—Local instability of a minimum-variance STR**

Consider a process described by

\[
y(t) - 1.6y(t - 1) - 0.75y(t - 2) = u(t - 1) + u(t - 2) + 0.9u(t - 3) + e(t) + 1.5e(t - 1) + 0.75e(t - 2)
\]

The \( B \) polynomial has zeros at

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

Furthermore,

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

The real part of \( C \) is thus negative at the zeros of \( B \). This implies that the parameters corresponding to minimum-variance control make an unstable
equilibrium for the ODEs. Furthermore, it follows from Theorem 5.2 that these parameter values are the only possible equilibrium point for the parameters. The following heuristic argument indicates that the estimates are bounded. If the parameters are such that the closed loop system is unstable, the inputs and the outputs will be so large that they dominate the stochastic terms in the model. The estimates will then quickly approach values that correspond to a deadbeat regulator for the system, which gives a stable closed-loop system. This argument can be made rigorous (see Johansson (1988)). The estimates will thus vary in a bounded area without converging to any point. Figure 6.18 shows the parameter estimates when a direct self-tuning regulator with the controller structure

\[ u(t) = \frac{s_0 + s_1q^{-1}}{1 + r_1q^{-1} + r_2q^{-2}} y(t) \]

is used. The simulation is initialized with values that corresponds to the minimum-variance controller. The simulation is done using RLS with a forgetting factor \( \lambda = 0.98 \). The figure shows that the estimates try to reach the optimal values but are repelled when they get close. The example shows that there exists a minimum-phase system, in which the parameters corresponding to the minimum-variance controller are not a stable equilibrium for the self-tuning algorithm. This particular example led, in fact, to an extensive research effort on the stability of stochastic self-tuners. \( \Box \)
6.8 Parameterization

The adaptive control systems discussed in this book center around the idea that a process is described as a linear system with parameters. It might therefore be expected that the way parameters are introduced is essential. Parameterization problems have already been encountered, for instance in the discussion of direct and indirect adaptive control in Chapters 4 and 5 but they will be pursued in more detail in this section. This is essential in order to get more insight into adaptive control and it will also lead to new adaptive algorithms.

Linearity in the Parameters

From the point of view of system identification it is highly desirable to have a parameterization that gives an error that is linear in the parameters i.e., of the form

\[ e = \varphi^T \theta \]  \hspace{1cm} (6.61)

where \( \varphi \) is a regression vector and \( \theta \) the parameters. The whole development of the error model that is the control theme of the MRAS is based on transformations that lead to Eq. (6.61). There are many parameter estimation schemes that can be applied to Eq. (6.61), such as the gradient method and the least-squares method. These estimation methods are also well understood.

Parameterization of Linear SISO Systems

The linear model

\[ Ay = Bu \]  \hspace{1cm} (6.62)

where \( A \) and \( B \) are polynomials in the differential operator or in the forward shift operator, is a representation that is commonly used in adaptive systems. The model has the advantage that it is linear in the parameters. A particular advantage of the representation in the discrete-time case is that it gives an exact representation of a rational transfer function with a time delay. It can be shown that the structure of the polynomials then is

\[ A(z) = z^{d+1} A_1(z) \]
\[ B(z) = B_1(z) \]

where

\[ d = \text{int} \left( \frac{\tau}{h} \right) \]  \hspace{1cm} (6.63)

and \( \tau \) is the time delay and \( h \) the sampling period. The polynomials \( A_1 \) and \( B_1 \) have the same degree. Notice that the model also allows fractional delay i.e. time delays that are not integer multiples of the
sampling period. The parameterization does, however, also have some significant drawbacks. The coefficients of the polynomials $A$ and $B$ may change drastically while maintaining a constant input-output behavior. This happens if $A$ and $B$ have a common factor or factors that are close. An example illustrates what may happen.

**Example 6.12  Common factor**
Consider a system with

$$A(s) = s^2 + as$$
$$B(s) = s + b$$

The system has the transfer function $G(s) = 1/s$ for all $a$ if $b = a$. If $b$ is close to $a$, the transfer function will be close to $1/s$ for a large range of $a$-values. \(\square\)

The opposite situation, in which the behavior may change drastically for small parameter variations, may also occur, as illustrated by the next example.

**Example 6.13—Sensitivity**
Consider a system with

$$A(z) - (z - 0.99)^4 + \epsilon$$
$$B(z) = 10^{-8}$$

The system can change from stable to unstable by changing $\epsilon$ from 0 to $2 \times 10^{-8}$. \(\square\)

These examples illustrate some of the difficulties. Other mechanisms that may give rise to difficulties are eigenvalues having different orders of magnitude.

Problems also appear with a sampled data model when the sampling period is small, because all the poles will be in a small neighborhood around $z = 1$. Furthermore, the sampled model will have “ghost zeros” outside the unit disc even if the continuous time system that is sampled is minimum phase. These zeros are given by the following theorem.

**Theorem 6.4—Limiting sampled-data zeros**
Let $G$ be a rational function

$$G(s) = \frac{(s - z_1)(s - z_2), \ldots, (s - z_m)}{(s - p_1)(s - p_2), \ldots, (s - p_n)} \quad (6.64)$$

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

\[
B_k(z) = b_1^k z^{k-1} + b_2^k z^{k-2} + \cdots + b_k^k
\]  \hspace{1cm} (6.65)

and

\[
b_i^k = \sum_{l=1}^{k-i} (-1)^i l^k \binom{k+1}{i-l}, \quad i = 1, \ldots, k
\]  \hspace{1cm} (6.66)

The first five polynomials \( B_k \) are

\[
\begin{align*}
B_1(z) &= 1 \\
B_2(z) &= z + 1 \\
B_3(z) &= z^2 + 4z + 1 \\
B_4(z) &= z^3 + 11z^2 + 11z + 1 \\
B_5(z) &= z^4 + 26z^3 + 66z^2 + 26z + 1
\end{align*}
\]

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

\[
\delta = \frac{q - 1}{h}
\]

or in Tustin’s operator

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

This yields parameterizations that give a much better resolution at \( q = 1 \). The \( \delta \) operator gives a description that is equivalent to the \( q \) operator description. The advantage of the transformation is that the \( \delta \) operator description has better numerical properties when the sampling is fast. All the poles of the \( q \) operator form are clustered around the point \( q = 1 \). This gives rise to numerical sensitivity (compare Example 6.13). For the \( \delta \) operator it can be shown that the limiting value

\[
\lim_{h \to 0} \frac{B_h(\delta)}{A_h(\delta)} = \frac{B_0(\delta)}{A_0(\delta)}
\]
is such that the coefficients in $B_0$ and $A_0$ are the same as the coefficients in the continuous-time transfer function. This implies that the structure of the transfer function in the $\delta$ operator is essentially the same as that of the continuous-time transfer function, provided the sampling period is sufficiently short.

When the pole excess is known, there is a simple, practical way to avoid difficulties with ghost zeros by simply using the standard direct algorithm with $d$ equal to the pole excess. Another difficulty with short sampling intervals for systems having time delays is that $d$ in Eq. (6.63) may become very large.

Several measures have been taken to avoid the problems with the representation in Eq. (6.62). One possibility is to use only models of low order and to choose the sampling period carefully. Another possibility is to choose $A^*(q^{-1}) = 1$ and a $B$ polynomial of high order, i.e., a finite impulse response function model (FIR). No cancellations will then be possible. However, the truncation of the impulse response may have severe drawbacks with systems having slow modes or systems with poorly damped oscillatory modes, since a large number of $b$ parameters are needed. A compromise is to use an $A$ polynomial of the form $A^*(q^{-1}) = 1 - aq^{-1}$. It is then easy to check if the $B$ polynomial has zero close to $z = a$. Another possibility is to parameterize the impulse response in functions that are orthogonal on $(0, \infty)$. One example is the Legendre polynomials, which are orthogonal on $(0, \infty)$ with the weighting function $\exp(-\alpha t)$.

**Partially Known Systems**

In specific applications it frequently happens that we know part of the system dynamics. It may then be advantageous to use this a priori knowledge and avoid estimating known parameters. Knowledge in terms of poles and zeros can easily be incorporated in the model of the process. Assuming that a factor $A_1$ of the $A$ polynomial and a factor $B_1$ of the $B$ polynomial are known, the model can be written as

$$A_2(A_1 y) = B_2(B_1 u)$$

If $\tilde{u} = B_1 u$ and $\tilde{y} = A_1 y$ are considered as new input and output signals, we obtain a standard problem with fewer parameters, and we can use the previous methods. This method is useful, for example, in connection with sensor and actuator dynamics which appear in cascade with the process dynamics. In many cases, however, the prior knowledge of parameters does appear as coefficients in differential equations. It is then necessary to use the continuous-time formulation, because the dependence of the parameters often becomes very complicated when the system is sampled. It may also be difficult, if not impossible, to obtain an error model that is linear in the parameters.
The Map $\chi$ from Process Parameters to Regulator Parameters

The design calculations are an important part of indirect adaptive systems. Theoretically the design procedure is represented by the function $\chi$, which maps process parameters $\theta$ to regulator parameters $\vartheta$. The properties of $\chi$ will, of course, depend on the parameterization of the model and the design procedure chosen. The function can often be quite complicated. The properties of the map will be discussed in some simple cases.

Consider the process model

$$Ay = Bu$$  \hspace{1cm} (6.67)

where it is assumed that $A$ has degree $n$ and $B$ degree $n-1$. The model thus has $2n$ parameters. Using pole placement design, the regulator parameters are given by

$$AR + BS = A_o A_m$$  \hspace{1cm} (6.68)

where $R$ and $S$ have the same degree $m$ as the observer polynomial $A_o$. In the generic case, $m = n - 1$, but it is possible to specify a higher-order observer. Without loss of generality $R$ can be monic. The regulator then has $2m + 1$ parameters. The function $\chi$ is thus a map from $R^{2n}$ to $R^{2m+1}$ where $m \geq n - 1$. Since Eq. (6.68) becomes singular when polynomials $A$ and $B$ have a common factor, it follows that the map $\chi$ has singularities. This is illustrated by an example.

**Example 6.14—Singularities for pole placement design**

Consider the model of Eq. (6.67), with

$$A(z) = z^2 + a_1 z + a_2$$

$$B(z) = b_1 z + b_2$$

Design a regulator with specified dynamics and observer polynomials. The design equation gives

$$A(z)(z + r_1) + B(z)(s_0 z + s_1) = A_o(z) A_m(z) = P(z)$$

Assume that $\deg P = 3$. With $z = -b_2/b_1$, the following equation is obtained when $b_1 \neq 0$:

$$r_1 = \frac{b_2}{b_1} + \frac{P(-b_2/b_1)}{A(-b_2/b_1)}$$

Equating coefficients for $z^2$ and $z^0$ gives

$$s_0 = \frac{p_1 - a_1 - r_1}{b_1} = \frac{1}{b_1} \left( p_1 - a_1 \frac{b_2}{b_1} - \frac{P(-b_2/b_1)}{A(-b_2/b_1)} \right)$$

$$s_1 = \frac{1}{b_2} (p_3 - a_2 r_1) = \frac{1}{b_2} \left( p_3 - a_2 \frac{b_2}{b_1} - a_2 \frac{P(-b_2/b_1)}{A(-b_2/b_1)} \right)$$
In this case the regulator parameters are

$$\vartheta = [r_1 \quad s_0 \quad s_1]$$

and the process parameters are

$$\theta = [a_1 \quad a_2 \quad b_1 \quad b_2]$$

The map $\chi$ from process parameters to regulator parameters is given by the above equations. Notice that the map is singular when $A(-b_2/b_1) = 0$ which means that there is a pole-zero cancellation in the process model. The regulator gains then become infinite. Notice that the singularities are not isolated points but an algebraic surface defined by

$$b_2^2 - a_1 b_1 b_2 + a_2 b_1^2 = 0$$

Singularities of the type in Example 6.14 will appear for practically all design methods based on state space. 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 $\chi$ discontinuous, which creates difficulties in the analysis. Another and perhaps better possibility is to find parameterizations and design procedures such that the mapping $\chi$ 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 6.15—Singularities for pole placement design**

Consider the system in Example 6.14 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 as $A_o = z$.

Figure 6.19 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 bounded. 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 6.14, and
the regulator 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. 6.20. In this particular case a factor is canceled if poles and zeros are so close that

\[ A \left( -\frac{b_2}{b_1} \right) \leq 0.01 \]  

(6.69)

The performance is now very good.

**The High-frequency Gain**

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

\[ A_o^* A_m^* y(t + d) = b_0 \left( R^* u(t) + S^* y(t) \right) \]
where $b_0$ is the coefficient of the first nonvanishing term in the $B$ polynomial. With some abuse of language, this coefficient is called the high-frequency gain, because it is the first nonvanishing coefficient of the impulse response. For continuous-time systems the transfer function of the process is approximately $G(s) = b_0 s^{-d_h}$. In Theorem 6.1 it was required that the sign of the coefficient $b_0$ be known. There are several ways to deal with the parameter $b_0$. It may be absorbed into $R$ and $S$ and estimated. The polynomial $R$ then has the form

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

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

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

and let the model be

$$y(t + 1) = r_0 \left( u(t) + \psi^T(t)\theta \right) = r_0 u(t) + \phi^T(t)\theta$$
With zero command signal the control law becomes

\[ u(t) = -\psi^T(t)\theta(t) \]

The equation for parameter updating is

\[ \hat{\theta}(t + 1) = \hat{\theta}(t) + P(t + 1)\varphi(t)e(t + 1) \]

where

\[ e(t + 1) = y(t + 1) = b_0 u(t) + b_0 \psi^T(t)\theta^0 \]

\[ = -b_0 \psi^T(t)\left(\hat{\theta}(t) - \theta^0\right) = -\frac{b_0}{r_0} \varphi^T(t)\left(\hat{\theta}(t) - \theta^0\right) \]

The estimation error is thus governed by

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

With a pure projection algorithm we have

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

In this case the matrix in large parentheses above has one eigenvalue \((1 - b_0/r_0)\) and the remaining eigenvalues 1. With least-squares updating the averaged equations for \(\hat{\theta}\) becomes

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

Hence, to remain stable, it must be required that

\[ 0 < \frac{b_0}{r_0} < 2 \]

If an algorithm with a fixed \(r_0\) is used, it is convenient to absorb \(r_0\) in the scaling of the signals. This is discussed in more detail in Chapter 11. When the parameter \(b_0\) is estimated, it can be treated as the other parameters. However, because of the special structure of the model it is useful to use special algorithms such as the ones discussed in Section 4.5.
6.9 Instability Mechanisms

It is important to understand the mechanisms that may cause instability in an adaptive system. Some insight into this has been developed previously, but this section will focus on the instability mechanisms.

Lack of Parameter Equilibrium

The notion of equilibrium is essential when dealing with nonlinear differential equations. For adaptive systems it is not particularly meaningful to talk about true equilibria, in the sense that all the states of a system are constant. Much more relevant are parameter equilibria, which correspond to the case in which the adjustable parameters are constant even when the other signals in the system are changing. The existence of true parameter equilibria depends on three elements: the nature of the external driving signal ($\nu$ in Section 6.3), the true system, and the model used to design the adaptive system. A true parameter equilibrium will always exist if the only excitation is the command signal and there are no unmodeled dynamics. The equilibrium is unique and independent of the external driving signal if the model used to design the adaptive system is compatible with the true system (i.e., there are no unmodeled dynamics). There will be a submanifold of equilibria if the system is overparameterized or if the external driving signal is not persistently exciting of sufficiently high order.

True parameter equilibria will not exist if there are unmodeled dynamics and the external driving forces are sufficiently complicated, involving a command signal that is persistently exciting of sufficiently high order, load disturbances, and measurement noise. If true parameter equilibria do not exist, there may still exist equilibria to the averaged equations. This corresponds to the physical situation in which parameters move around in the neighborhood of a fixed value.

Bad Local Properties

If parameter equilibria exist, they must be such that the parameters correspond to stable solutions of the closed-loop system. The averaged equations must also be locally stable at the equilibria. Having parameter equilibria outside the stability range is thus an obvious instability mechanism. This is illustrated by Example 6.5. Another instability mechanism is that the equilibrium is locally unstable, as in Example 6.4.

Parametric Excitation

Instabilities can be generated in linear systems by changing the parameters periodically, a phenomenon called parametric excitation. A classical
example is the Mathieu equation
\[
\frac{d^2 y}{dt^2} + \alpha \frac{dy}{dt} + (\beta + \gamma \cos \omega t)y = 0
\]
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. Parameter excitation can also occur in adaptive systems, as shown by the following example.

**Example 6.16—Parameter excitation**
Consider adjustment of a feedforward gain for a system with the transfer function \( G(s) = 1/(s+1) \), using the MIT rule. The parameter adjustment rule is
\[
\frac{d\theta}{dt} = -\gamma y_m(t)e(t)
\]
where
\[
e(t) = y(t) - y_m(t)
\]
The system output is given by
\[
\frac{dy}{dt} = \theta(t)u_c(t) - y(t)
\]
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
\[
\frac{d}{dt} \begin{pmatrix} \theta \\ y \end{pmatrix} = \begin{pmatrix} 0 & -\gamma y_m(t) \\ u_c(t) & -1 \end{pmatrix} \begin{pmatrix} \theta \\ y \end{pmatrix} + \begin{pmatrix} \gamma y_m^2(t) \\ 0 \end{pmatrix}
\]
This is a linear system with time-varying parameters. When the command signal is periodic, it may become unstable due to parametric excitation. The stability region for \( u_c(t) = \sin \omega t \) is shown in Fig. 6.21. The example indicates that even very simple adaptive systems can exhibit complex behavior. Notice, however, that the system is stable for low adaptation gains. Compare Example 6.4.

Linear differential equations with periodic equations like the one encountered in Example 6.16 are well understood. A central result is the following theorem.

**Theorem 6.5—Solution of periodic system**
The solution to the matrix differential equation
\[
\frac{d\Phi}{dt} = A(t)\Phi
\]
(6.70)
where $A(t)$ is periodic with period $T$ and continuous for all $t$, has the form

$$\Phi(t) = B(t)e^{Ct}$$

where $C$ is a constant matrix and $B$ periodic with period $T$.

The theorem, which is proven in standard textbooks on ordinary differential equations, e.g., Bellman (1953), can be used to analyze the stability of linear equations with periodic coefficients. To see how this should be done, consider a solution with

$$\Phi(0) = I$$

where $I$ is the identity matrix. It follows from the theorem that

$$\Phi(T) = e^{CT}$$

The differential equation is stable if the matrix $C$ has all its eigenvalues in the left half-plane, which means that the matrix $\Phi(T)$ should have all its eigenvalues inside the unit disc. Stability can thus be determined by numerical integration over one period. The use of the theorem is illustrated by an example.

Example 6.17—Parameter excitation (continued)

Consider the system in Example 6.16. Let the command signal be $u_c(t) = \sin \omega t$. After a transient, the model output becomes

$$y(t) = \frac{1}{\sqrt{1 + \omega^2}} \sin(\omega t - \text{atan} \ (\omega))$$
Choosing $\omega = 2$ and integrating to $T = \pi$ gives

$$\Phi(T) = \begin{pmatrix} 0.4373 & 0.7283 \\ 0.2389 & 0.4967 \end{pmatrix}$$

with eigenvalues 0.049 and 0.885 for $\gamma = 10$, and

$$\Phi(T) = \begin{pmatrix} 0.5609 & 0.9960 \\ 0.2642 & 0.5463 \end{pmatrix}$$

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 $\gamma = 11$. □

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. A high gain is a common way of generating instabilities in feedback systems. Results such as Example 4.6 and Theorem 6.1 indicate that arbitrarily high adaptation gain may be used in some cases. However, since these results are based on the assumption that certain transfer functions are strictly positive real, they are highly unrealistic. (Recall that SPR transfer functions will remain stable under feedback with arbitrarily large gain.) Under more realistic assumptions, it has already been shown that high adaptive gains may lead to instabilities.

Example 6.18 Instability at high pole excess and high gain
Consider a direct MRAS for adjustment of a feedforward gain, described by the equations

$$y = G(p)u$$
$$u = \theta u_c$$
$$\frac{d\theta}{dt} = -\gamma u_c e$$
$$e = y - y_m$$

Compare Example 4.7. The parameters are governed by

$$\frac{d\theta}{dt} + \gamma u_c (G(p)\theta u_c) = \gamma u_c y_m$$

Consider the case in which the command signal is constant $u_c^o$. The differential equation then reduces to

$$\frac{d\theta}{dt} + \gamma u_c^o (G(p)\theta) = \gamma u_c^o y_m$$
This is a linear equation with constant coefficients. Its characteristic equation is
\[ s + \gamma u_c^2 G(s) = 0 \tag{6.71} \]
If the transfer function \( G \) has a pole excess of 2 or more, the system will always be unstable for sufficiently high adaptation gains. A detailed picture of how the zeros of Eq. (6.71) vary with adaptation gain is easily obtained by a root locus argument.

This example illustrates clearly how large adaptation gains may generate instabilities. It has also been shown that high adaptation gain can generate chaotic behavior in an adaptive system.

The analysis in the example can also be used to provide an order of magnitude estimate of appropriate adaptation gains. For small adaptation gains the system has poles close to the poles of \( G \) and also a pole at
\[ s_a = -\gamma u_c^2 G(0) \]
This pole is associated with the parameter adjustment. As adaptation gain increases, this pole will interact with the poles of \( G \). To maintain a time separation between the parameter estimator and the process dynamics, we can require that
\[ |s_a| < \varepsilon |s_1| \]
where \( s_1 \) is the pole of \( G \) that has the largest real part and \( \varepsilon \) is a safety factor, say \( \varepsilon = 0.1 \). The adaptation gain should thus be chosen as
\[ \gamma = \frac{\varepsilon |s_1|}{u_c^2 G(0)} \]
Notice that it is signal-dependent. Compare the analysis in Example 4.4.

**Bounds on the Adaptation Gain**

The discussion of the instability mechanisms has clearly indicated that it is desirable to limit the adaptation gain. In this way it is possible to avoid the instability mechanisms associated with parametric excitation and high adaptation gain. This makes sense intuitively, because the very notion of parameters implies that they should be constant or slowly changing. It is therefore natural to use an averaging approach, as discussed in Section 6.3. An analysis of the local behavior of the averaged equation for the parameters gives an estimate of the parameter dynamics. The eigenvalues associated with parameter dynamics are proportional to the adaptation gain \( \gamma \). Compare Eqs. (6.19) or (6.20). The adaptation gain can then be chosen so that the eigenvalues of the estimator are smaller than the desired eigenvalues of the closed-loop system. Because of the complexity of
the problem, the given rule is, of course, merely a guideline. The final choice must be verified by simulation or experimentation. Notice that the adaptation gain will be signal-dependent for updating laws such as

\[
\frac{d\theta}{dt} = \gamma \varphi e
\]

In practice it is therefore highly desirable to use normalized updating, such as

\[
\frac{d\theta}{dt} = \frac{\gamma \varphi e}{\alpha + \varphi^T \varphi}
\]

or a least-squares method.

The procedure of choosing a reasonable adaptation gain is illustrated by an example.

**Example 6.19—Choice of adaptation gain**

Consider the MRAS discussed in Section 6.5. The closed-loop system has a specified pole at \( s_1 = -3 \). With a step input the largest eigenvalue of the averaged parameter equations is \( 2\gamma u_0^2 / \theta_1 = 2\gamma \). With a sinusoidal input the eigenvalues vary with the frequency of the command signal. The largest value is \( \gamma u_0^2 / \theta_1 = \gamma \). Requiring at least an order of magnitude difference between the eigenvalues, we get \( \gamma < 0.2 \). This is a conservative value compared with Fig. 6.4, where \( \gamma = 2 \). \( \Box \)

**Summary**

Three mechanisms that may lead to instability have been discussed. One mechanism is associated with lack of a parameter equilibrium or local instability of the equilibrium. The other mechanisms are parametric excitation and high adaptation gain. The last two mechanisms can be avoided by choosing a small adaptation gain. A simple rule of thumb for estimating reasonable adaptation gain has also been provided.

**6.10 Universal Stabilizers**

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

\[
\frac{dy}{dt} = ay + bu
\]

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

\[
u = f(\theta, y)\
\]

\[
\frac{d\theta}{dt} = g(\theta, y)
\]

(6.73)
that stabilizes the system for all values of $a$ and $b$? Morse (1983), suggested that there are no rational $f$ and $g$ that solve the problem. Morse’s conjecture was verified by Nussbaum (1983), who proved the following result.

**Theorem 6.6—Universal stabilizer**

The control law of Eq. (6.73), with

$$f(\theta, y) = y\theta^2 \cos \theta$$
$$g(\theta, y) = y^2$$

and $\theta(0) = 0$, stabilizes Eq. (6.73).

**Proof:** The closed-loop system is described by

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

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

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

Integration with respect to time gives

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

Hence

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

But

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

Hence

$$\lim_{t \to \infty} \inf \frac{1}{\theta} \int_0^\theta x^2 \cos x \, dx = -\infty$$
This gives
\[ \lim_{\theta \to \infty} \inf_{t} \frac{y^2(t)}{\theta(t)} = -\infty \]
which is a contradiction, because \( y^2 / \theta \) is nonnegative. It thus follows that
\[ \lim_{t \to \infty} \theta(t) = \theta^0 < \infty \]
Integration of the equation for \( \theta \) gives
\[ \theta(t) = \int_0^t y^2(t) dt \]
It then follows from Lemma 4.1 that
\[ \lim_{t \to \infty} y(t) = 0 \]
\[ \square \]
Notice that the control law of Eq. (6.74) can be interpreted as proportional feedback with the gain \( k = \theta^2 \cos \theta \). The behavior of the control law can be interpreted as follows. Sweep over all possible regulator gains and stop when a stabilizing gain has been found. The function \( g \) can be interpreted as the rate of change of the gain sweep. The rate is large for large errors and small for small errors. The form \( \cos \theta \) makes sure that the gains can be both positive and negative.

The control law of Eq. (6.74) is useful because it does not contain any parameters that relate to the system it stabilizes. It is therefore called a universal stabilizer. However, the control law is restricted to a first-order system. In attempting to generalize Theorem 6.6 to higher-order systems the following question was posed. How much prior information about an unknown system is required in order to stabilize it? This question was answered in a general setting by Martensson (1985), who showed that it is sufficient to know the order of a stabilizing fixed-gain controller. If a transfer function is given it is unfortunately a nontrivial task to find the minimal order of a stabilizing regulator.

Universal stabilizers may show very violent transient behavior. This is not surprising, since the system may be temporarily unstable during the sweep over the gains. The behavior of a universal stabilizer is illustrated in Fig. 6.22. A reference value is used in the simulations, and the control law is then modified to
\[ f(\theta, y) = (u_c - y)\theta^2 \cos \theta \]
\[ g(\theta, y) = (u_c - y)^2 \]
(6.75)
Figure 6.22  Simulation of the control law of Eq. 6.74 applied to the plants
(a) $G(s) = 1/(1 - s)$; (b) $G(s) = 1/(s - 1)$.

6.11 Conclusions

Analysis of adaptive systems is difficult because they are complicated. A number of different methods have been used to gain insight into the behavior of adaptive systems. A basic stability theorem has been derived, based on standard tools of the theory of difference equations. To show stability and convergence it is necessary to make quite restrictive assumptions about the system to be controlled. The consequences of violating these assumptions have been analyzed. It has been shown that analysis of equilibria and local properties around equilibria can be explored by the method of averaging. This method can also be applied to investigate global properties. Averaging can be applied in many different situations. For deterministic problems it can be used for steps and periodic signals. It can also be applied to stochastic signals. Averaging methods have also been applied to analyze what happens when adaptive systems are designed based on simplified models. Problems related to parameterization have also been discussed. To apply averaging methods, it is necessary to use small adaptation gains. There are unfortunately no good methods to determine analytically how small the gains should be. It is also demonstrated that adaptive systems may have very complex behavior for large adaptation gains.
6.1 Consider the system described by

\[ y_t = u_{t-d} \]

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

6.2 Consider Theorem 6.1. Generalize the results to cover the case in which the polynomial \( B^* \) has isolated zeros on the unit circle.

6.3 Consider the system

\[ y_t = u_{t-1} + a \]

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

6.4 Consider the system in Example 6.8. Interpret the results as if the adaptive algorithm tried to estimate parameters \( a \) and \( b \) in the transfer function \( G(s) = b/((s + a) \). Show that

\[ \hat{a} = \frac{229 - 31\omega^2}{259 - \omega^2} \]
\[ \hat{b} = \frac{458}{259 - \omega^2} \]

Compare Eq. (6.44). Determine the parameters for \( \omega = 2.72 \) and \( \omega = 17.03 \). Explain the results by evaluating \( G(s) \) for the corresponding frequencies.

6.5 Construct a proof analogous to Theorem 6.1 for continuous-time systems.

6.6 Formulate the averaging equation for a discrete-time algorithm.

6.7 Consider discrete-time adaptive control of the system

\[ y(t + 1) = ay(t) + bu(t) \]

Derive an MRAS that gives a closed-loop system

\[ y_m(t + 1) = a_m y_m(t) + b_m u_c(t) \]

Use averaging methods to analyze the system when the command signal is a step and a sinusoid.
6.8 Consider the previous example. Investigate the behavior of the system when the command signal is a step and when there is sinusoidal measurement noise.

6.9 Consider the MRAS given by Eq. (6.22). Investigate the local behavior of the closed-loop system when the command signal is a sinusoid and the gradient method

\[ \frac{d\theta}{dt} = \gamma \varphi e \]

is replaced with a least-squares method of the form

\[ \frac{d\theta}{dt} = P\varphi e \]

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

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

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

\[ G(s) = \frac{a}{s + a} \]

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

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

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

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

\[ u_c = a_1 \sin \omega_1 t + a_2 \sin \omega_2 t \]

and assume that the process has the transfer function

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

Derive conditions for the closed-loop system to be stable.
6.13 Consider the MRAS given by Eq. (6.22). Make a simulation study to investigate the consequences of introducing leakage as described by Eqs. (6.45) and (6.46) in the estimation algorithm. Study sinusoidal command signals as well as step commands and measurement noise.

6.14 Consider the MRAS in Problem 6.13. Make a simulation study to investigate the consequences of using conditional updating. Study sinusoidal command signals as well as step commands and measurement noise.

6.15 Consider the system in Problem 6.13. Let the input be sinusoidal with frequency $\omega$. Investigate the effects of sinusoidal measurement noise on the system.

6.16 Consider direct algorithms for control of the system

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

to give an input-output relation

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

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

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

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

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

6.20 Consider a system for adjustment of a feedforward gain based on the MIT rule. Let the command signal be $u_c(t) = \sin \omega t$, and let $G(s) = 1/(s+1)$. Simulate the parameter behavior for the MIT rule with adaptation gains $\gamma = 10$ and $\gamma = 11$. Compare the analysis in Example 6.4.

6.21 Consider the simulation shown in Fig. 6.4, which was performed with adaptation gain $\gamma = 1.0$. Repeat the simulation with different adaptation gains.

References

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

The proof of Theorem 6.1 follows the ideas in:


Equivalent results for continuous-time systems are presented in:


Many variations of the stability theorem are given in:


Related results are presented in:


A stability analysis for bounded disturbances is given in:


The case of mean square bounded disturbances was investigated in:

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


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


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


A simple presentation of the key ideas is given in:


More detailed treatments are given in:


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


The example of nonrobustness in Example 6.5 is based on:

This initiated the discussion of the robustness problem. The analysis in Sections 6.4 and 6.5 is largely based on:


The idea of introducing leakage is found in:


The idea of normalization was suggested by Praly. See e.g.,


It is further explored in:


Further discussion of robustness is given in:


Stochastic averaging was introduced in:

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


More recent proofs of the method are found in:


An accessible account is also given in:


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


Conditions for local stability of the equilibrium are given in:


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


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

The parameterization problem and adaptive control of partially known systems have not received much attention.

Work on universal stabilizers was initiated by a discussion of whether Assumption A4 of Theorem 6.1 is necessary. See:


The problem was solved for scalar systems in:


Universal stabilisers for multivariable systems is discussed in:


Martensson showed (to summarize roughly) that the order of a stabilizing regulator is the only information required for adaptive stabilization of a multivariable system.
Chapter 7

STOCHASTIC
ADAPTIVE CONTROL

7.1 Introduction

In Chapters 4, 5, and 6 the adaptive control problem was approached from a heuristic point of view. The unknown parameters of the process or the regulator were estimated using real-time estimation, and the estimated parameters were then used as if they were the true ones. The uncertainties of the parameter estimates were not taken into account in the design. This procedure gives a certainty equivalence controller. The model-reference adaptive controllers and the self-tuning regulators have been derived under the assumption that the parameters are constant but unknown. The estimation routines are then such that the uncertainties usually decrease rapidly after the estimation is started. However, the uncertainties can be large at the start-up or if the parameters are changing. In such cases it may be important to let the control law be a function of
the parameters as well as of the uncertainties.

It would be appealing to formulate the adaptive control problem from a unified theoretical framework. This can be done using nonlinear stochastic control theory, by which the process, its parameters, and the environment are described using a stochastic model. The difference compared with the treatment in the previous chapters is that the parameters of the process also are described using a stochastic model. The criterion is formulated so as to minimize the expected value of a loss function. It is difficult to find the controller that minimizes the expected loss function. Conditions for the existence of an optimal controller are not known. However, under the condition that a solution exists, it is possible to derive a functional equation using dynamic programming. This equation, called the Bellman equation, can be solved numerically only in very simple cases. The structure of the optimal regulator is shown in Fig. 7.1. The controller is composed of two parts: an estimator and a feedback regulator. The estimator generates the conditional probability distribution of the state from the measurements. This distribution is called the hyperstate of the problem. The feedback regulator is a nonlinear function that maps the hyperstate into the space of control variables.

The structural simplicity of the solution is obtained at the price of introducing the hyperstate, which can be a quantity of very high dimension. Notice that the structure is similar to that of the self-tuning regulator. The self-tuning regulator can be regarded as an approximation; the conditional probability distribution is replaced by a distribution with all mass at the conditional mean value. In Fig. 7.1 there is no distinction between the parameters and the other state variables of the process. The regulator can therefore handle very rapid parameter variations. Furthermore, the averaging methods based on separation of the states of the process and the parameters (used in Chapter 6) cannot be used to analyze the system.

The optimal control law has an interesting property. The control
Chapter 7  Stochastic Adaptive Control

attempts to drive the output to the desired value, but it will also introduce perturbations when the estimates are uncertain. This will improve the estimates and the future control. The optimal controller achieves a correct balance between maintaining good control and small estimation errors. This is called dual control and was first introduced by Feldbaum in 1960.

The chapter is organized in the following way. The problem formulation is given in Section 7.2, and Section 7.3 gives the derivation of the functional equation. The consequences of the structure of the solution are discussed, and the dual property is analyzed. Different ways to approximate the dual controller are discussed in Section 7.4. Only very simple examples of dual controllers are solved numerically, but the solutions give some useful indications of how suboptimal controllers can be constructed. Some examples are given in Section 7.5, and the stochastic adaptive approach is summarized in Section 7.6.

7.2 Problem Formulation

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

The Model

Consider the discrete-time single input single output system

\[ y(t) + a_1(t)y(t-1) + \cdots + a_n(t)y(t-n) = b_1(t)u(t-1) + \cdots + b_n(t)u(t-n) + e(t) \]  

(7.1)

where \( y, u, \) and \( e \) are output, input, and disturbance, respectively. The noise sequence \( \{e(t)\} \) is assumed to be Gaussian with zero mean and variance \( R_2 \). Further, it is assumed that \( e(t) \) is independent of \( y(t-1), y(t-2), \ldots, u(t-1), u(t-2), \ldots, a_i(t), a_i(t-1)\ldots, \) and \( b_1(t), b_i(t-1), \ldots \). It is further assumed that \( b_1(t) \neq 0 \) and that the system is minimum-phase for all \( t \). The time-varying parameters

\[ x(t) = \begin{bmatrix} b_1(t) & \ldots & b_n(t) & a_1(t) & \ldots & a_n(t) \end{bmatrix}^T \]  

(7.2)

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

\[ x(t + 1) = \Phi x(t) + v(t) \]  

(7.3)
where $\Phi$ is a known constant matrix and $\{v(t)\}$ is a sequence of independent, equally distributed normal vectors with zero mean value and known covariance $R_1$. The initial state of the system in Eq. (7.3) is assumed to be normally distributed with mean value

$$Ex(0) = m$$

and covariance

$$\text{cov} \ [x(0), x(0)] = R_0$$

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

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

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

where

$$\varphi^T(t - 1) = [u(t - 1) \ldots u(t - n) - y(t - 1) \ldots - y(t - n)]$$

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

The Criterion

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

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

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

Admissible Control Strategies

To specify the problem completely it is necessary to define the admissible control strategies. A control strategy is admissible if $u(t)$ is a function of all outputs observed up to and including time $t$, i.e., $y(t), y(t - 1), \ldots$ all applied control signals $u(t - 1), \ldots$ and the $a$ priori data. Let $Y_t$ denote all values of the output up to and including $y(t)$ or, more precisely, the $\sigma$-algebra generated by $y(t), \ldots, y(0)$ and $x(0)$.
Discussion of the Problem Formulation

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

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

7.3 Dual Control

The problem formulated in the previous section will now be analyzed. The problem of estimating the parameters of Eq. (7.1) is first considered. The control problem is first solved for the case in which the parameters are known. The problem will then be solved for the case when \( N = 1 \) in the criterion of Eq. (7.8). The solution of the complete problem is finally discussed. The control problem is solved using dynamic programming.

The Estimation Problem

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

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

**Theorem 7.1—Conditional distribution of the states**

Consider the model of Eq. (7.3) with the output defined by Eq. (7.6) where \( e(t) \) and \( v(t) \) are independent zero mean Gaussian variables with covariances \( R_2 \) and \( R_1 \) respectively. The initial state of the system is given by Eqs. (7.4) and (7.5).
The conditional distribution of $x(t)$ given $Y_{t-1}$ is Gaussian with mean $\hat{x}(t)$ and covariance $P(t)$ satisfying the difference equations

$$
\begin{align*}
\hat{x}(t + 1) &= \Phi x(t) + K(t)(y(t) - \varphi^T(t-1)x(t)) \\
P(t + 1) &= (\Phi - K(t)\varphi^T(t-1))P(t)\Phi^T + R_1 \\
K(t) &= \Phi P(t)\varphi(t-1)(R_2 + \varphi^T(t-1)P(t)\varphi(t-1))^{-1}
\end{align*}
$$

(7.9)

with the initial conditions

$$
\begin{align*}
x(0) &= m \\
P(0) &= R_0
\end{align*}
$$

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

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

and covariance

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

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

Remark. Notice that the conditional distribution of $y(t)$ given $Y_{t-1}$ is Gaussian even if $y(t)$ is not Gaussian.

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

$$
\xi(t) = \begin{bmatrix} \hat{\varphi}(t-1) & \hat{x}(t) & P(t) \end{bmatrix}
$$

(7.10)

where

$$
\hat{\varphi}^T(t-1) = \begin{bmatrix} 0 & u(t-2) & \ldots & u(t-n) & y(t-1) & \ldots & y(t-n) \end{bmatrix}
$$

(7.11)

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

**Systems with Known Parameters**

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

$$y(t + 1) = \varphi^T(t)x(t + 1) + e(t + 1)$$
$$= b_1(t + 1)u(t) + \tilde{\varphi}^T(t)x(t + 1) + e(t + 1)$$

The optimal feedback when $b_1(t + 1)$ and $x(t + 1)$ are known is then given by

$$u(t) = \frac{y_r(t + 1) - \tilde{\varphi}^T(t)x(t + 1)}{b_1(t + 1)} \quad \text{(7.12)}$$

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

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

and it minimizes Eq. (7.8), since $e(t + 1)$ is independent of $Y_t$ and $u(t)$. The minimal loss is given by

$$\min J_N = R_2$$

Notice that it is necessary to assume that $b_1(t + 1) \neq 0$ and that the system is minimum-phase at every instant of time. The control signal may otherwise be unbounded.

**Certainty Equivalence Control**

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

$$u(t) = \frac{y_r(t + 1) - \tilde{\varphi}^T(t)\hat{x}(t + 1)}{\hat{b}_1(t + 1)} \quad \text{(7.13)}$$

The true parameter values are replaced by the expected values given $Y_t$. The controller of Eq. (7.13) is called the *certainty equivalence controller*.

**Cautious Control**

The special case when $N = 1$ in Eq. (7.8) will now be considered. According to Theorem 7.1 the conditional distribution of $y(t + 1)$ given $Y_t$ is
Gaussian with mean $\varphi^T(t)\hat{x}(t+1)$ and covariance $R_2 + \varphi^T(t)P(t+1)\varphi(t)$. Then

$$E\left\{ (y(t+1) - y_r(t+1))^2 | Y_t \right\}$$

$$= (\varphi^T(t)\hat{x}(t+1) - y_r(t+1))^2 + R_2 + \varphi^T(t)P(t+1)\varphi(t)$$

$$= (\hat{\varphi}^T(t)\hat{x}(t+1) + \hat{b}_1(t+1)u(t) - y_r(t+1))^2 + R_2$$

$$+ \varphi^T(t)P(t+1)\varphi(t) + u^2(t)p_{b_1}(t+1)$$

$$+ 2u(t)\hat{\varphi}^T(t)P(t+1)\ell$$

(7.14)

The first equality is obtained by using the standard formula that

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

when $\zeta$ is a Gaussian variable with mean $m$ and variance $p$. The column vector $\ell$ selects the first column of the matrix $P(t)$, i.e.,

$$\ell^T = 1 \ 0 \ldots 0$$

Equation (7.14) is quadratic in $u(t)$. Minimization of Eq. (7.14) with respect to $u(t)$ gives the admissible one-step optimal controller

$$u(t) = \frac{b_1(t+1)y_r(t+1) - \hat{\varphi}(t)b_1(t+1)x(t+1) + P(t+1)\ell}{\hat{b}_1^2(t+1) + p_{b_1}(t+1)}$$

(7.15)

The minimum value of the loss function is

$$\min_{u(t)} E\left\{ (y(t+1) - y_r(t+1))^2 | Y_t \right\}$$

$$= (\hat{\varphi}^T(t)\hat{x}(t+1) - y_r(t+1))^2 + R_2 + \varphi^T(t)P(t+1)\varphi(t)$$

$$- \left( \frac{b_1(t+1)y_r(t+1) - \hat{\varphi}(t)\hat{b}_1(t+1)x(t+1) + P(t+1)\ell}{\hat{b}_1^2(t+1) + p_{b_1}(t+1)} \right)^2$$

(7.16)

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

7.3 Dual Control 309
Example 7.1—Integrator with time-varying gain
Consider an integrator where the gain is changing. Let the process be described by
\[ y(t) - y(t - 1) = b(t)u(t - 1) + e(t) \]
where
\[ b(t + 1) = \varphi_b b(t) + R_1 v(t) \]
The errors \( e \) and \( v \) are zero mean Gaussian white noise with the standard deviations \( R_2 \) and 1, respectively. Further, it is assumed that \( y_r = 0 \). The certainty equivalence controller is given by
\[ u(t) = -\frac{1}{b(t + 1)} y(t) \]
and the cautious controller is
\[ u(t) = -\frac{\hat{b}(t + 1)}{\hat{b}^2(t + 1) + p_b(t + 1)} y(t) \]
The gain in the cautious controller has been reduced by a factor
\[ \frac{\hat{b}^2}{\hat{b}^2 + p_b} \]
compared with the certainty equivalence controller. Notice that the gain approaches zero when the uncertainty increases.

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

It follows from a fundamental result of stochastic control theory (see Åström (1970), Lemma 8:3.2) that
\[
\min_{u_1(t) \ldots u_{N-1}} \mathbb{E} \left\{ \sum_{k=t}^{N} (y(k) - y_r(k))^2 \right\}
= \mathbb{E}_{Y_{t-1}} \left( \min \mathbb{E} \left\{ \sum_{k=t}^{N} (y(k) - y_r(k))^2 | Y_{t-1} \right\} \right)
\]
and it is assumed that the minimum exists. $E(\cdot|Y_{t-1})$ is a function of the hyperstate of Eq. (7.10) and $t$. Define

$$V(\xi(t), t) = \min_{u(t-1)\ldots u(N-1)} E \left\{ \sum_{k=t}^{N} (y(k) - y_r(k))^2 | Y_{t-1} \right\}$$

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

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

$$V(\xi(N), N) = (\hat{\phi}^T(N-1)\hat{x}(N) - y_r(N))^2 + R_2 + \hat{\phi}^T(N-1)P(N)\hat{\phi}(N-1)$$

$$- \left( \hat{b}_1(N)y_r(N) - \hat{\phi}^T(N-1)(\hat{b}_1(N)\hat{x}(N) + P(N)\ell) \right)^2$$

$$\hat{\delta}_1^2(N) + p_{b_1}(N)$$

At time $N - 1$ we get

$$V(\xi(N - 1), N - 1) = \min_{u} E \left\{ (y(N - 1) - y_r(N - 1))^2 + V(\xi(N), N) | Y_{N-2} \right\}$$

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

$$V(\xi(t), t) = \min_{u(t-1)} E \left\{ (y(t) - y_r(t))^2 + V(\xi(t+1), t+1) | Y_{t-1} \right\}$$

(7.17)

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

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

$$E \left\{ V(\xi(t+1), t+1) | Y_{t-1} \right\}$$
The average with respect to the distribution of \( y(t) \) given \( Y_{t-1} \) must be computed. According to Theorem 7.1 this distribution is Gaussian with mean \( m_y(t) \) and variance \( \sigma_y^2(t) \). This gives

\[
E \left\{ V(\xi(t+1), t+1) | Y_{t-1} \right\} = \frac{1}{\sigma_y \sqrt{2\pi}} \int_{-\infty}^{\infty} V(\phi(t), \hat{x}(t+1), P(t+1), t+1) e^{-(s-m_y)^2/(2\sigma_y^2)} ds
\]  

(7.18)

where

\[
\hat{x}(t+1) = \Phi \hat{x}(t) + K(t)(s - \varphi^T(t-1)\hat{x}(t))
\]

\[
P(t+1) = (\Phi - K(t)\varphi^T(t-1))P(t)\Phi^T + R_1
\]

\[
K(t) = \Phi P(t)\varphi(t-1)/\sigma_y^2(t)
\]

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

\[
\varphi_1(t) = u(t-1)
\]

\[
\varphi_i(t) = \varphi_{i-1}(t-1) \quad i = 2, \ldots, n, n+2, \ldots, 2n
\]

\[
\varphi_{n+1}(t) = s
\]

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

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

**Separation and Certainty Equivalence**

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

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

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

**Numerical Difficulties**

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

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

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

Another numerical difficulty is that the control law can become discontinuous in situations such as that shown in Fig. 7.2. The figure shows the loss function as a function of the control signal for three different but close values of the hyperstate. If there are several local minima, the control signal can become discontinuous when the global minimum changes from one local minimum to another.
7.4 Suboptimal Strategies

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

Cautious Controllers

Minimization over only one step leads to the one-step or cautious controller of Eq. (7.15). This controller takes the parameter uncertainties into account, in contrast to the certainty equivalence controller of Eq. (7.13). However, the gain of Eq. (7.15) will decrease if the variance of $b_1$ increases. This will give less information about $b_1$ in the next step, and the variance will increase further. The controller is then caught in a vicious circle, and the magnitude of the control signal becomes very small. This is called the turn-off phenomenon.

Example 7.2—Turn-off

Consider the integrator with unknown gain in Example 7.1 with $R_1 = 0.09$ and $\varphi_b = 0.95$. Figure 7.3 shows a representative simulation of the cautious controller. The control signal is small for periods of time, and the variance of the gain increases during the turn-off. After some time the control activity suddenly starts again.

The turn-off will generally start when the control signal is small and when the parameter $b_1$ is small. The problem with turn-off makes the cautious controller unsuitable for control of systems with quickly varying
parameters. The cautious controller can be useful if the parameters of the process are constant or almost constant, but the certainty equivalence controller with some simple safety measures can often be used in such cases also.

**Classification of Suboptimal Dual Controllers**

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

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

**Perturbation Signals**

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

**Constrained One-step Minimization**

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

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

One method is to choose the control as

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

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

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

\[ \text{tr } P^{-1}(t + 2) \geq M \]

\( P^{-1} \) is proportional to the information matrix. The constraint on the trace of \( P^{-1} \) means that the information about the parameters is always larger than some chosen value \( M \). A similar approach is to constrain only the variance of \( b_1 \) to

\[ p_{b_1}(t + 2) \leq \begin{cases} 
  \gamma b_1^2(t + 2) & \text{if } p_{b_1}(t + 1) \leq b_1^2(t + 1) \\
  \alpha p_{b_1}(t + 1) & \text{otherwise} 
\end{cases} \]

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

**Extensions of the Loss Function**

An approach that is similar to constrained minimization is to extend the loss function in order to prevent the shortsightedness of the cautious controller. One obvious way is to try to solve the two-step minimization problem. The derivation in the previous section shows that it is not possible to get an analytic solution when \( N = 2 \) in Eq. (7.8).
Another approach is to extend the loss function with a function of \( P(t+2) \), which will reward good parameter estimates. The following loss function can be used:

\[
\min_{u(t)} E \left\{ (y(t+1) - y_r(t+1))^2 + \lambda f(P(t+2)) \mid Y_t \right\} \tag{7.19}
\]

where \( \lambda \) is a fixed parameter. Since the crucial parameter is \( b_1 \), we can use

\[
f(P(t+2)) = p_{b_1}(t+2)
\]

or

\[
f(P(t+2)) = R_2 \frac{p_{b_1}(t+2)}{p_{b_1}(t+1)} \tag{7.20}
\]

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

Two similar approaches are to modify the loss functions to

\[
\min_{u(t)} E \left\{ (y(t+1) - y_r(t+1))^2 - \lambda \frac{\det P(t+1)}{\det P(t+2)} \mid Y_t \right\} \tag{7.21}
\]

and

\[
\min_{u(t)} E \left\{ (y(t+1) - y_r(t+1))^2 - \lambda \varepsilon^2(t+1) \mid Y_t \right\} \tag{7.22}
\]

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

\[
\varepsilon(t+1) = y(t+1) - \varphi^T(t)x(t+1)
\]

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

### Serial Expansion of the Loss Function

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

**Summary**

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

### 7.5 Examples

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

**Example 7.3—Optimal dual controller**

The first example is a numerically solved dual control problem from Åström and Helmersson (1983). Consider the integrator in Example 7.1. The gain is assumed to be constant but unknown, i.e., \( \varphi_b = 1 \) and \( R_1 = 0 \). It is assumed that the parameter \( b \) is a random variable with a Gaussian prior distribution; the conditional distribution of \( b \), given inputs and outputs up to time \( t \), is Gaussian with mean \( \hat{b}(t) \) and standard deviation \( \sigma(t) \). The hyperstate can then be characterized by the triple \( (y(t), \hat{b}(t), \sigma(t)) \). The equations for updating the hyperstate are given by Eq. (7.9).

Define the loss function

\[
V_N = \min_u \mathbb{E} \left\{ \sum_{k=t+1}^{t+N} y^2(k) \mid Y_t \right\}
\]

where \( Y_t \) denotes the data available at time \( t \) i.e., \( \{y(t), y(t-1), \ldots\} \). By introducing the normalized variables

\[
\eta = y/\sqrt{R_2}, \quad \beta = \hat{b}/\sigma, \quad \mu = -u\hat{b}/y
\]

it can be shown that \( V_N \) depends on \( \eta \) and \( \beta \) only. Further introduce the normalized innovation

\[
\varepsilon(t) = \frac{y(t+1) - y(t) - \hat{b}(t)u(t)}{R_2 + u(t)^2\sigma(t)}
\]
For $R_2 = 1$ the Bellman equation for the problem can be written as

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

where

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

and

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

where $\phi$ is the normal probability density with zero mean and unit variance and

$$y = \eta - \mu \eta + \varepsilon \sqrt{1 + \frac{\mu^2 \eta^2}{\beta^2}}$$

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

When the minimization is performed, the control law is obtained as

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

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

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

The original variables give

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

This control law is the one-step control or myopic control derived in Example 7.1.

For $N > 1$ the optimization can no longer be done analytically. Instead, we have to resort to numerical calculations. Figure 7.4 shows the dual control laws obtained for different time horizons $N$. The discontinuity of the control law corresponds to the situation that a probing signal is introduced to improve the estimates.
The certainty equivalence controller

\[ u(t) = -y(t)/\hat{b} \]  \hspace{1cm} (7.23)

can be expressed as

\[ \mu = 1 \]  \hspace{1cm} (7.24)

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

Example 7.4—Probing
An interesting feature of the dual control law is that it behaves quite differently from the heuristic algorithms. The most significant feature is the probing that takes place in order to gain more information about the unknown parameters. The effect of probing is most significant when the output $y$ is small. Probing can be illustrated using the results of Example 7.3. Both the cautious and the certainty equivalence control laws are continuous in $y$ and zero for $y = 0$. The dual control law is, however, very different. To show this, consider the control signal for $y = 0$. Figure 7.5 shows the control signal for $y = 0$—as a function of the normalized parameter precision, $\beta = \hat{b}/\sigma$, for different time horizons. All control laws give zero control signal when the parameter estimate is reasonably precise. However, for uncertain estimates the control signal is different from zero, and the transition is discontinuous. This discontinuity can be used to define a probing zone. Notice that the probing zone increases with increasing time horizon. For $N = 31$, probing occurs when $\beta \leq 1.5$, i.e., when $\hat{b} \leq 1.5\sigma$.

Example 7.5—Time-varying parameters
The system in Example 7.2 will now be controlled by a suboptimal dual controller that minimizes Eq. (7.19). See Wittenmark and Elevitch (1985)
(i.e., minimization of Eq. (7.19)). Figure 7.6 shows the same experiment using the same noise sequences as in Fig 7.3. With the suboptimal dual controller there is no tendency of turn-off. The simulation in Fig. 7.6 shows that the suboptimal dual controller is much better than the cautious controller. Comparisons using Monte Carlo simulations have also been done with the numerically computed optimal dual controller. The result is that the suboptimal dual controller is as good as the numerically computed optimal controller. A summary of some simulations is shown in Fig. 7.7 which shows mean values and standard deviations of the loss when the standard deviation of the parameter noise $R_1$ is changed. It is assumed that

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

For $\sqrt{R_1} = 0.003$ there is good agreement between the suboptimal controller and the optimal dual controller which was derived under the assumption that $R_1 = 0$. The optimal dual controller from Example 7.3 corresponds to $R_1 = 0$. The controller obtained has been used also for $\sqrt{R_1} = 0.003$.

In Eq. (7.20) $R_2/p_{b_1}(t + 1)$ is used as a normalization factor in the term added to the loss function. The reason is an attempt to preserve a property of the dual optimal controller. In Example 7.3 the loss $V_N$ was a function of the normalized variables $\eta$ and $\beta$. The loss function of
Figure 7.7 The mean values and standard deviations of the losses for Monte Carlo runs with different values of \( \rho = R_1 \) for the system in Example 7.5. (a) Cautious controller. (b) The suboptimal dual controller from Wittenmark and Craig (1985). (c) Numerically computed optimal dual controller from Åström and Helmersson (1982).

Eq. (7.19) with Eq. (7.20) will also have this property for the integrator example. Simulations indicate that the normalization in Eq. (7.20) also makes the choice of \( \lambda \) less crucial.

7.6 Conclusions

Optimal multistep controllers have been derived using stochastic control theory. The solution is defined through the Bellman equation. This functional equation is difficult to solve even for very simple systems. The optimal solution has some interesting properties; it makes a compromise between good control and good estimation by introducing probing actions. This dual effect is of great importance, since it introduces an active learning feature into the controller. It is important to preserve this dual feature when suboptimal controllers are considered. The cautious or one-step controller does not have any active learning; the control may instead be turned off when the parameter uncertainties become too large.

One important question is whether it is worth the effort to look at more elaborate control structures than the certainty equivalence controllers. The self-tuning regulators perform very well, as could be seen in Chapters 5 and 6. The extra computations are not too extensive in several of the suboptimal dual controllers discussed in Section 7.5. This indicates that active learning can easily be included.
Chapter 7  Stochastic Adaptive Control

There are two situations in which dual control can pay off. One is when the time horizon is very short and when it is important to get good parameter estimates immediately. Areas in which this is the case are economic systems and control of missiles. The other situation in which dual features are important is when the parameters are varying rapidly and when the $b_1$ parameter can change sign, as in the simulations given in Section 7.5.

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

Problems

7.1 Discuss possible difficulties of extending the problem given in Section 7.2 to the case when the system in Eq. (7.1) has an additional time delay.

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

7.3 Consider the process in Example 7.1, but with a constant but unknown gain $b$. Calculate and compare the minimum values of the loss function when (a) the parameter $b$ is known (i.e., the minimum-variance controller), (b) the certainty equivalence controller is used and (c) the cautious controller is used.

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

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

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

$$y(t) = \varphi(t)\theta_i + e(t) \quad i = 1, \ldots, m$$

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

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

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

$$y(t) = C(t)x(t) + e(t)$$
where $\Phi$, $\Gamma$, and $C$ contain some unknown parameters. For simplicity consider the case in which the system is given in controllable canonical form, i.e.,

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

$$
\Gamma^T(t) = \begin{bmatrix} b_1(t) \cdots b_n(t) \end{bmatrix}
$$

$$
C(t) = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}
$$

References

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


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


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


The difference between certainty equivalence and separation is treated in:

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


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


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


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


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


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


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


Chapter 8

AUTO-TUNING

8.1 Introduction

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

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

The chapter is organized as follows: The standard PID controller is discussed in Section 8.2, transient and frequency response methods for tuning are developed in Sections 8.3 and 8.4, and Section 8.5 is devoted to analysis of relay oscillations. Conclusions are presented in Section 8.6 and the chapter ends with problems and references.

## 8.2 PID Control

Many simple control problems can be handled very well by PID control, provided that the requirements are not too high. The PID algorithm is packaged in the form of standard regulators for process control and is also the basis of many tailor-made control systems. The textbook version of the algorithm is

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

(8.1)

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

\[
    u(t) = P(t) + I(t) + D(t)
\]  

(8.2)
where
\[ P(t) = K_c(u_c(t) - y(t)) \]
\[ \frac{dI}{dt} = \frac{K_c}{T_i}(u_c(t) - y(t)) + \frac{1}{T_i}(v(t) - u(t)) \]  
(8.3)
\[ \frac{T_d}{N} \frac{dD}{dt} + D = -K_cT_d \frac{dy}{dt} \]

The last term in the expression for \( dI/dt \) is introduced to keep the integral bounded when the output saturates. The variable \( v \) is a tracking signal which is equal to the output of the saturating actuator, and the parameter \( T_i \) is a time constant for the tracking action. The essential parameters to be adjusted are \( K_c, T_d \) and \( T_i \). The parameter \( N \) can be fixed; a typical value is \( N = 10 \). The tracking time constant is typically a fraction of the integration time \( T_i \).

### 8.3 Transient Response Methods

Several simple tuning methods for PID controllers are based on transient response experiments. Many industrial processes have step responses of the type shown in Fig. 8.1, in which the step response is monotonous after an initial time. A system with a step response of the type shown in Fig. 8.1 can be approximated by the transfer function
\[ G(s) = \frac{k}{1 + sT} e^{-sL} \]  
(8.4)


Table 8.1 The Ziegler-Nichols step response method.

<table>
<thead>
<tr>
<th>Controller</th>
<th>$K_c$</th>
<th>$T$</th>
<th>$T_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>$1/a$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI</td>
<td>$0.9a$</td>
<td>$3L$</td>
<td></td>
</tr>
<tr>
<td>PID</td>
<td>$1.2/a$</td>
<td>$2L$</td>
<td>$L$</td>
</tr>
</tbody>
</table>

where $k$ is the static gain, $L$ the apparent time delay, and $T$ the apparent time constant. Notice that

$$a = k \frac{L}{T}$$  \hspace{1cm} (8.5)

The Ziegler-Nichols Step Response Method

A simple way to determine the parameters of a PID regulator based on step response data was developed by Ziegler and Nichols and published in 1942. The method uses only two of the parameters shown in Fig. 8.1 namely $a$ and $L$. The regulator parameters are given in Table 8.1. The Ziegler-Nichols tuning rule was developed by empirical simulations of many different systems. The rule has the drawback that it gives closed-loop systems that are often too poorly damped. Systems with better damping can be obtained by modifying the numerical values in Table 8.1. By using additional parameters it is also possible to determine if the Ziegler-Nichols rule is applicable. If the time constant $T$ is also determined, an empirical rule is established that the Ziegler-Nichols rule is applicable if $0.1 < L/T < 1$. For large values of $L/T$ it is advantageous to use control laws that compensate for dead time. For small values of $L/T$ improved performance may be obtained with higher-order compensators. It is also possible to use more sophisticated tuning rules based on three parameters.

Characterization of a Step Response

The parameters $k$, $L$, and $T$ can be determined from a graphical construction such as the one indicated in Fig. 8.1. Such a method is difficult to automate. The parameter $k$ can be obtained from the ratio of static changes in input and output in steady state. There is a simple method, based on area measurements, to determine $L$ and $T$ (see Fig. 8.2). The area $A_0$ is first determined. Then

$$T + L = A_0 \frac{k}{k}$$  \hspace{1cm} (8.6)
The area $A_1$ under the step response up to time $T + L$ is then determined and $T$ is then given by

$$T = \frac{eA_1}{k} \quad (8.7)$$

where $e$ is the base of the natural logarithm. The essential drawbacks of the method are that it may be difficult to know the size of the step in the control signal and to determine whether a steady state has been reached. The step should be so large that the response is clearly noticeable above the noise but not so large that production is disturbed. Disturbances will also influence the result significantly.

**On-line Refinement**

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

**Use for Pre-tuning**

A model such as Eq. (8.14) is very useful in order to select sampling period and achievable closed-loop response for an MRAS or an STR.

## 8.4 Methods Based on Relay Feedback

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

**The Key Idea**

The basic idea is the observation that many processes will have limit cycle oscillations under relay feedback. The essential process properties can be determined from features of the limit cycle, and the parameters of a PID
8.4 Methods Based on Relay Feedback

regulator can be computed. Figure 8.3 shows a block diagram of a relay auto-tuner. When tuning is demanded, the switch is set to \( T \), which means that relay feedback is activated and the PID regulator disconnected. When a stable limit cycle is established, the PID parameters are computed, and the PID controller is then connected to the process. An approximative method will first be used to get some insight into the information that can be derived from an experiment with relay feedback.

The Method of Harmonic Balance

An approximative method called the *method of harmonic balance* or the *describing function method* will be described. Consider a simple feedback system composed of a linear part with the transfer function \( G(s) \) and feedback with an ideal relay. The block diagram is shown in Fig. 8.4. We assume in this section that \( u_c = 0 \). An approximative condition for oscillation can be determined as follows: Assume that there is a limit cycle with period \( T_u \) and frequency \( \omega_u = 2\pi/T_u \) such that the relay output is a periodic symmetric square wave. If the relay amplitude is \( d \), a simple Fourier series expansion of the relay output shows that the first harmonic component has the amplitude \( 4d/\pi \). Assume further that the process dynamics are of low-pass character and that the contribution from the first harmonic dominates the output. The error signal then has the amplitude

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

The condition for oscillation is thus that

\[
\arg G(i\omega_u) = -\pi \quad \text{and} \quad K_u = \frac{4d}{\pi a} = \frac{1}{|G(i\omega_u)|}
\]  

(8.8)

where \( K_u \) can be regarded as the equivalent gain of the relay for transmission of sinusoidal signals with amplitude \( a \). The condition is thus that the
linear system in Fig. 8.4 has a Nyquist curve that intersects the negative axis. The amplitude $a$ and the frequency of the oscillation $\omega_u$ are easily obtained from Eq. (8.8). The frequency of the limit cycle is thus automatically adjusted to the frequency $\omega_u$ at which the open-loop process dynamics has a phase lag of 180°. The corresponding period is called the ultimate period, for historical reasons. The parameter $K_u$ is called the ultimate gain. Physically, it is the gain that brings the system to the stability boundary under pure proportional control. An experiment with relay feedback will thus give the period and the amplitude of the open-loop transfer function of the process at the frequency at which the phase lag is 180°. Notice also that an input signal whose energy content is concentrated at $\omega_u$ is generated automatically in the experiment.

Several refinements are useful. The amplitude of the limit cycle oscillation can be specified by introducing a feedback that adjusts the relay amplitude.

A hysteresis in the relay is useful to make the system less sensitive to noise. It will now be shown how to determine the parameters of a PID regulator from $T_u$ and $k_u$. The method can be made insensitive to disturbances by comparing and averaging over several periods of the oscillation.

**The Ziegler-Nichols Closed-loop Method**

A very simple rule for choosing the parameters of PID regulators is ideally matched to the determination of $K_u$ and $T_u$ by the relay method. The controller settings are given in Table 8.2. These parameters give a closed-loop system with quite low damping. Systems with better damping can be obtained by slight modifications of the numbers in the table.

**Improved Estimates**

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

The data obtained from the relay experiment can also be used to estimate a discrete-time transfer function using standard system identification methods.

### Application to Pre-tuning

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

### An Example

The properties of a relay auto-tuner is illustrated by an example. The process to be controlled consists of three cascaded tanks. The level of the lower tank is measured, and the control variable is the voltage to the amplifier driving the pump for the inlet. The signals are noisy. The relay in the auto-tuner has a hysteresis, which is determined automatically based on measurements of the process disturbances. The relay amplitude is also adjusted automatically to keep a specified amplitude of the limit cycle. The limit cycle is judged to be stationary by measuring the periods and amplitudes of two positive half-periods. Figure 8.5 shows the process inputs and outputs in one experiment, illustrating the effect of amplitude adjustment. When the tuning is finished the regulator is switched to PID
control automatically. A change of the setpoint shows that the tuning has been successful.

8.5 Relay Oscillations

Since limit cycling under relay feedback is a key idea of relay auto-tuning, it is important to understand why a linear system oscillates under relay feedback and when the oscillation is stable. It is also important to have methods for determination of the period and the amplitude of the oscillations. Consider the system shown in Fig. 8.4. Introduce the following state space realization of the transfer function $G(s)$:

$$
\frac{dx}{dt} = Ax + Bu
$$

$$
y = Cx
$$

(8.9)

The relay can be described by

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

(8.10)

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

**Theorem 8.1—Limit cycle period**

Assume that the system defined in Fig. 8.4 and by Eqs. (8.9) and (8.10) has a symmetric limit cycle with period $T$. The period $T$ is then the smallest value of $T > 0$ that satisfies the equation

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

(8.11)
where
\[ \Phi = e^{AT/2} \]
and
\[ \Gamma = \int_0^{T/2} e^{As} ds B \]

**Proof:** Let \( t_k \) denote the times when the relay switches. Since the limit cycle is symmetric, it follows that
\[ t_{k+1} - t_k = T/2 \]

Assume that the control signal \( u \) is \( d \) over the interval \((t_k, t_{k+1})\). Integration of Eq. (8.9) over the interval gives
\[ x(t_{k+1}) = \Phi x(t_k) + \Gamma d \]

Since the limit cycle is symmetric, it also follows that
\[ x(t_{k+1}) = -x(t_k) \]

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

Since the output \( y(t) \) must be zero at \( t_k \), it follows that
\[ y(t_k) = C x(t_k) = -C(I + \Phi)^{-1} \Gamma d = 0 \]

which gives Eq. (8.11).

**Remark 1.** Notice that the condition of Eq. (8.11) can also be written as
\[ H_{T/2}(-1) = 0 \quad (8.12) \]

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

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

**Remark 3.** Similar conditions can also be derived for relays with hysteresis.
Comparison with Describing Function

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

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

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

$$H_h(-1) = \sum_{n=-\infty}^{\infty} \frac{2}{i(\pi + 2n\pi)} G\left(\frac{i\pi + 2n\pi}{h}\right)$$

$$= \sum_{n=0}^{\infty} \frac{4}{\pi(1+2n)} \text{Im}\left(G\left(\frac{i\pi + 2n\pi}{h}\right)\right) = 0$$

The first term of the series gives

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

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

**Example 8.1—Limit cycle period**

Assume that the linear part has the transfer function

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

Simple calculations show that

$$\arg G(i\omega_u) = -\frac{\pi}{2} - \tan^{-1}\omega_u - \tan^{-1}\frac{\omega_u}{a}$$

$$= -\frac{\pi}{2} - \tan^{-1}\frac{\omega_u(a+1)}{a - \omega_u^2} = -\pi$$

This implies that $\omega_u = \sqrt{a}$. The approximative analysis thus gives the following estimate of the period:

$$T_u = \frac{2\pi}{\sqrt{a}} = 6.28 \frac{1}{\sqrt{a}}$$
To apply Theorem 8.1, the system is sampled with period $h$. The pulse transfer function is

$$H_h(z) = \frac{h}{(z-1)} - \frac{a(1-e^{-h})}{(a-1)(z-e^{-h})} + \frac{1-e^{-ah}}{a(a-1)(z-e^{-ah})}$$

Hence

$$H_h(-1) = -\frac{h}{2} + \frac{a(1-e^{-h})}{(a-1)(1+e^{-h})} - \frac{1-e^{-ah}}{a(a-1)(1+e^{-ah})}$$

$$= -\frac{h}{2} + \frac{a}{a-1} \left( \frac{1-e^{-h}}{1+e^{-h}} - \frac{1}{a^2} \frac{1-e^{-ah}}{1+e^{-ah}} \right) = 0$$

A series expansion in $1/a$ gives, for large $a$,

$$H_h(-1) \approx -\frac{h}{2} + \left( 1 + \frac{1}{a} \right) \frac{h}{2} \left( 1 - \frac{h}{2} + \frac{h^2}{6} \right)$$

$$\approx -\frac{h}{2} + \frac{h}{2} \left( 1 + \frac{1}{a} \right) \left( 1 - \frac{h^2}{12} \right) = 0$$

This gives

$$h \approx 2\sqrt{3/a}$$

This means that the period is approximately

$$T_u = 4\sqrt{3/a} = \frac{6.93}{\sqrt{a}}$$

which should be compared with the value given by the approximate formula.

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

8.6 Conclusions

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

**Problems**

8.1 Consider a process characterized by the transfer function

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

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

8.2 Consider a process with the transfer function

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

Show that Eq. (8.6) gives

\[ T + L = \Sigma T_k + L \]

8.3 Consider a process described by the transfer function

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

Determine a proportional regulator that gives an amplitude margin \( A_m = 2 \). Show that it is identical to the setting obtained by applying the Ziegler-Nichols rule in Table 8.1.

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

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

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

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

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


The Ziegler-Nichols tuning rules were presented in:


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


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


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


The transient response method for automatic tuning of PID regulators is used in products from Foxboro, Yokogawa, Eurotherm, and Honeywell. It is used for pre-tuning in adaptive controllers from Leeds and Northrup and Turnbull control. The relay autotuner was presented in:


It is also patented:


A comprehensive discussion of automatic tuning of simple regulators is found in the monograph:

The relay auto-tuner is used in products from SattControl AB and Fisher Controls Inc. These regulators are very easy to use, since tuning is achieved simply by pushing the tuning button.

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

Chapter 9

GAIN SCHEDULING

9.1 Introduction

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

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

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

### 9.2 The Principle

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

A main problem in the design of systems with gain scheduling is to find suitable scheduling variables. This is normally done based on knowledge of the physics of a system. In process control the production rate can often be chosen as a scheduling variable, since time constants and time delays are often inversely proportional to production rate. Compare Example 2.2.
Figure 9.1  Block diagram of a system in which influences of parameter variations are reduced by gain scheduling.

When scheduling variables have been determined, the regulator parameters are calculated at a number of operating conditions, using some suitable design method. The regulator is thus tuned or calibrated for each operating condition. The stability and performance of the system are typically evaluated by simulation; particular attention is given to the transition between different operating conditions. The number of entries in the scheduling tables is increased if necessary. Notice, however, that there is no feedback from the performance of the closed-loop system to the regulator parameters. Gain scheduling should thus not be regarded as an adaptive regulator but rather as a special type of nonlinear regulator.

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

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

Gain scheduling has the advantage that the regulator parameters can be changed very quickly in response to process changes. Since no estima-
Figure 9.2 Compensation of nonlinear actuator using an approximate inverse.

tion of parameters occurs, the limiting factors depend on how quickly the auxiliary measurements respond to process changes.

9.3 Design of Gain Scheduling Regulators

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

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

The ideas are illustrated by some examples.

Example 9.1—Nonlinear actuator
Consider the system with a nonlinear valve in Example 2.1. The nonlinearity is assumed to be

\[ v = f(u) = u^4 \quad u \geq 0 \]

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

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

Assume that \( f(u) = u^4 \) is approximated by two straight lines: one connecting the points \((0, 0)\) and \((1.3, 3)\), and the other connecting \((1.3, 3)\) and \((2, 16)\). See Fig. 9.3. Then

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

Figure 9.4 shows step changes in the reference signal at three different operating conditions when the approximation of the inverse of the valve characteristic is used between the regulator and the valve. Compare with the uncompensated system in Fig. 2.2. There is a considerable improvement in the performance of the closed-loop system. By improving the inverse it is possible to make the process even more insensitive to the nonlinearity of the valve.

The example above shows a simple and very useful idea to compensate for known static nonlinearities. In practice it is often sufficient to approximate the nonlinearity by a few line segments. There are several commercial single-loop controllers that can make this kind of compensation. DDC packages usually include functions that can be used to implement nonlinearities. The resulting controller is nonlinear and should (in its basic form) not be regarded as gain scheduling. In the example there is no measurement of any operating condition apart from the regulator output. In other situations the nonlinearity is determined from measurement of several variables.

Gain scheduling based on an auxiliary signal is illustrated in the following example.
Example 9.2—Tank system
Consider a tank where the cross section $A$ varies with height $h$. The model is

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

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

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

where

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

A good PI control of the tank is given by

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

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

and

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

This gives a closed-loop system with natural frequency \( \omega \) and relative damping \( \zeta \). Introducing the expressions for \( \alpha \) and \( \beta \) gives the following gain schedule:

\[ K = 2\zeta \omega A(h^0) - \frac{g_{in}^0}{2h^0} \]

\[ T_i = \frac{2\zeta}{\omega} - \frac{g_{in}^0}{2A(h^0)h^0\omega^2} \]

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

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

\[ T_i = \frac{2\zeta}{\omega} \]

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

\[ \Box \]

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

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

**Example 9.3—Concentration control**

Consider the concentration control problem in Example 2.2. The process is described by Eq. (2.1). Assume that we are interested in manipulating the concentration in the tank, \( c \), by changing the inlet concentration, \( c_{in} \). For a fixed flow the dynamics can be described by the transfer function

\[ G(s) = \frac{1}{1 + sT} e^{-s\tau} \]
where \[ T = \frac{V_m}{q} \quad \tau = \frac{V_d}{q} \]

If \( \tau < T \), then it is straightforward to determine a PI controller that
performs well when \( q \) is constant. However, it is difficult to find universal
values of the controller parameters that will work well for wide ranges of \( q \).
This is illustrated in Fig. 2.5, which shows the step responses of a fixed
gain controller for varying flows.

Since the process has a time delay, it is natural to look for sampled
data controllers. Sampling of the model with sampling period \( h = \frac{V_d}{(dq)} \),
where \( d \) is an integer gives

\[ c(kh + h) = ac(kh) + (1 - a)u(kh - dh) \]

where \[ a = e^{-qh} \quad V_m = e^{-V_d/(V_mD)} \]

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

The gain scheduling is realized simply by having a regulator with
constant parameters in which the sampling rate is inversely proportional
to the flow rate. This will give the same response, independent of the flow
when looking at the sampling instants, but the transients will be scaled in
time. Figure 9.5 shows the output concentration and the control signals
for three different flows. To implement this gain-scheduling controller it is
necessary to measure not only the concentration but also the flow. Errors
in the flow measurement will result in jitter in the sampling period. To
avoid this it is necessary to filter the flow measurement.

In Examples 9.1 and 9.2 it was possible to determine the schedules
exactly. The behavior of the closed-loop system does not depend on the
operating conditions. In other cases it is possible to obtain only approxi-
mate relations for different operating conditions. The design then has to
be repeated for several operating conditions to create a table. It is also
necessary to interpolate between the values of the table to obtain a smooth
behavior of the closed-loop system. This can lead to extensive calculations
and simulations before the full gain schedule is obtained.

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

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

\[ t_s = \frac{V_d}{q} t \]

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

A system of the form

\[ \dot{x}(t) = f(x(t)) + g(x(t))u(t) \]

can also be transformed into a linear system, provided that all states of the system can be measured and a generalized observability condition holds.

The design is done by first transforming the system into a fixed linear system. The transformation is usually nonlinear and depends on the states
of the process. A regulator is then designed for the transformed model and the control signals of the model are retransformed into the original control signals. The result is a special type of nonlinear controller which can be interpreted as a gain-scheduling controller. Knowledge about the nonlinearities in the model is built into the controller. The method with nonlinear transformations is illustrated by an example.

**Example 9.4—Nonlinear transformation of a pendulum**

Consider the system

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -\sin x_1 + u \cos x_1 \\
y &= x_1
\end{align*}
\]  

(9.1)

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

\[ u'(t) = -\sin x_1(t) + u(t) \cos x_1(t) \]

This gives the linear equations

\[
\dot{x} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u'
\]

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

\[ u'(t) = -l'_1 x_1(t) - l'_2 x_2(t) + m' u_c(t) \]

The transfer function from \( u_c \) to \( y \) is

\[
\frac{m'}{s^2 + l'_2 s + l'_1}
\]

Let the desired characteristic equation be

\[
s^2 + p_1 s + p_2
\]  

(9.2)

which can be obtained with

\[
l'_1 = p_2 \quad l'_2 = p_1 \quad m' = p_2
\]

Transformation back to the original control signal gives

\[
u(t) = \frac{u'(t) + \sin x_1}{\cos x_1} = \frac{1}{\cos x_1} \left( -p_2 x_1 - p_1 x_2 + p_2 u_c + \sin x_1 \right)
\]  

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

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

$$u(t) = -l_1 x_1(t) - l_2 x_2(t) + m u_c(t)$$  \hspace{1cm} (9.4)

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

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

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

Example 9.5—Nonlinear transformation of second-order system
Consider the system

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

$$\frac{dx_2}{dt} = f_2(x_1, x_2, u)$$
Assume that the state variables can be measured and that we want to find a feedback such that the response of the variable \( x_1 \) to the command signal is given by the transfer function

\[
G(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}
\]

(9.5)

Introduce new coordinates \( z_1 \) and \( z_2 \), defined by

\[
z_1 = x_1
\]

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

and the new control signal \( v \), defined by

\[
v = F(x_1, x_2, u) = \frac{\partial f_1}{\partial x_1} f_1 + \frac{\partial f_1}{\partial x_2} f_2
\]

(9.6)

These transformations result in the linear system

\[
\frac{dz_1}{dt} = z_2
\]

\[
\frac{dz_2}{dt} = v
\]

(9.7)

It is easily seen that the linear feedback

\[
v = \omega^2(u_c - z_1) - 2\zeta \omega z_2
\]

(9.8)

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

\[
v = F(x_1, x_2, u) = \omega^2(u_c - x_1) - 2\zeta \omega f_1(x_1, x_2)
\]

Solving this equation for \( u \) gives the desired feedback. It follows from the implicit function theorem that a condition for local solvability is that the partial derivative \( \frac{\partial F}{\partial u} \) is different from zero.

The generalization of the example requires a solution to the general problem of transforming a nonlinear system into a linear system by nonlinear feedback. Conditions and examples are given in the references in the end of this chapter. A simple version of the problem also occurs in
control of industrial robots. In this case the basic equation can be written as

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

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

### 9.5 Applications of Gain Scheduling

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

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

#### Ship Steering

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

The ship dynamics are obtained by applying Newton's equations to the motion of the ship. For large ships the motion in the vertical plane can be separated from the other motions. It is customary to describe the horizontal motion using a coordinate system fixed to the ship (see Fig. 9.7). Let \( V \) be the total velocity, \( u \) and \( v \) the \( x \) and \( y \) components of the velocity, and \( r \) the angular velocity of the ship.

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

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

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

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

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

where

$$\begin{align*}
K &= K_0 \frac{u}{l} \\
T_i &= T_{i0} \frac{l}{u} \quad i = 1, 2, 3 \tag{9.11}
\end{align*}$$

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

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

where

$$\begin{align*}
b &= b_0 \left(\frac{u}{l}\right)^2 = b_2 \left(\frac{u}{l}\right)^2 \\
a &= a_0 \left(\frac{u}{l}\right) \tag{9.13}
\end{align*}$$
Table 9.1 Parameters of models for different ships.

<table>
<thead>
<tr>
<th>Ship</th>
<th>Mine-sweeper</th>
<th>Cargo</th>
<th>Tanker</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Length m</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>161</td>
<td>350</td>
</tr>
<tr>
<td>$a_{11}$</td>
<td>$-0.86$</td>
<td>0.77</td>
<td>$-0.45$</td>
</tr>
<tr>
<td>$a_{12}$</td>
<td>$-0.48$</td>
<td>$-0.34$</td>
<td>$-0.43$</td>
</tr>
<tr>
<td>$a_{21}$</td>
<td>$-5.2$</td>
<td>$-3.39$</td>
<td>$-4.1$</td>
</tr>
<tr>
<td>$a_{22}$</td>
<td>$-2.4$</td>
<td>$-1.63$</td>
<td>$-0.81$</td>
</tr>
<tr>
<td>$b_1$</td>
<td>0.18</td>
<td>0.17</td>
<td>0.10</td>
</tr>
<tr>
<td>$b_2$</td>
<td>$-1.4$</td>
<td>$-1.63$</td>
<td>0.81</td>
</tr>
<tr>
<td>$K_0$</td>
<td>2.11</td>
<td>$-3.86$</td>
<td>0.83</td>
</tr>
<tr>
<td>$T_{10}$</td>
<td>8.25</td>
<td>5.66</td>
<td>$-2.88$</td>
</tr>
<tr>
<td>$T_{20}$</td>
<td>0.29</td>
<td>0.38</td>
<td>0.38</td>
</tr>
<tr>
<td>$T_{30}$</td>
<td>0.65</td>
<td>0.89</td>
<td>1.07</td>
</tr>
<tr>
<td>$a_0$</td>
<td>0.14</td>
<td>0.19</td>
<td>$-0.28$</td>
</tr>
<tr>
<td>$b_0$</td>
<td>$-1.4$</td>
<td>$-1.63$</td>
<td>$-0.81$</td>
</tr>
</tbody>
</table>

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

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

where $D$ (in m$^3$) is the displacement, $A$ (in m$^2$) is the rudder area, and $c$ is a parameter whose value is approximately 0.5. The parameter $a$ will depend on trim, speed, and loading. Its sign may change with the operating conditions.

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

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

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

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

**Example 9.6—Ship steering**

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

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

is used. The loop transfer function is

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

The characteristic equation of the closed-loop system is

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

The relative damping is

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

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

$$a = a_{nom} \frac{u}{u_{nom}}$$

$$b = b_{nom} \left( \frac{u}{u_{nom}} \right)^2$$
This gives the damping
\[
\zeta = \frac{1}{2} \left( \frac{a_{\text{nom}}}{\sqrt{Kb_{\text{nom}}}} + \frac{u}{u_{\text{nom}}} T_d \sqrt{Kb_{\text{nom}}} \right)
\]

Consider an unstable tanker with
\[
\begin{align*}
&c_{\text{nom}} = -0.3 \\
b_{\text{nom}} = 0.8 \\
&K = 2.5 \\
&T_d = 0.86
\end{align*}
\]

This gives $\zeta = 0.5$ and $\omega = 1.4$ at the nominal velocity. Furthermore,
\[
\begin{align*}
&\omega = 1.4u/u_{\text{nom}} \\
&\zeta = -0.11 + 0.61u/u_{\text{nom}}
\end{align*}
\]

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

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

\[\square\]

**pH Control**

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

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

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

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

\[
\text{H}_2\text{O} \rightleftharpoons \text{H}^+ + \text{OH}^-
\]

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

\[
\frac{[H^+][OH^-]}{[\text{H}_2\text{O}]} = \text{constant} \quad (9.16)
\]

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

\[
[H^+][OH^-] = K_w \quad (9.17)
\]

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

**Example 9.7 Titration curve for a strong acid-base pair**

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

\[
\text{HCl} + \text{NaOH} \rightarrow \text{H}^+ + \text{OH}^- + \text{Na}^+ + \text{Cl}^{-}
\]

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

\[
[\text{Cl}^-] = x_A = m_A/V
\]

and the concentration of sodium ions is given by

\[
[\text{Na}^+] = x_B = m_B/V
\]

because the acid and the base are completely ionized. Since the number of positive ions equals the number of negative ions it follows that

\[
x_A + [\text{OH}^-] = x_B + [H^+]
\]
The concentration of hydroxyl ions can be related to the hydrogen ion concentration by Eq. (9.17). Hence

\[ x = x_B - x_A = [\text{OH}^-] - [\text{H}^+] = \frac{K_w}{[\text{H}^+]} - [\text{H}^+] = 10^{p\text{H}-14} - 10 \ p\text{H} \] (9.18)

Solving for $[\text{H}^+]$ gives

\[ [\text{H}^+] = \sqrt{x^2/4 + K_w} - x/2 \]
\[ [\text{OH}^-] = \sqrt{x^2/4 + K_w} + x/2 \]

This gives

\[ p\text{H} = f(x) = -\log\left(\sqrt{x^2/4 + K_w} - x/2\right) \] (9.19)

The graph of the function $f$ is called the titration curve. It is the fundamental nonlinearity for the neutralization problem. An example of the titration curve is shown in Fig. 9.8, which shows that there is considerable variation in the slope of the titration curve. The abscissa of the titration curve in Fig. 9.8 is given in terms of the concentration difference $x_B - x_A$. The $x$-axis can also be recalibrated into the amount of the reagent.

The derivative of the function $f$ is given by

\[ f'(x) = \frac{10 \log e}{2\sqrt{x^2/4 + K_w}} = \frac{10 \log e}{10^{p\text{H}-14} + 10^{-p\text{H}}} \] (9.20)

The derivative has its largest value $f' = 2.2 \cdot 10^6$ for $p\text{H} = 7$. It decreases rapidly for larger and smaller values of $p\text{H}$. For $p\text{H} = 4$ and 10 we have $f' = 4.3 \cdot 10^3$. The gain can thus vary by several orders of magnitude.

Figure 9.8 shows that the pH of a strong acid that is almost neutralized may change very rapidly if only a small amount of base is added. The
reason for this is that strong acids and bases are completely dissociated. A weak acid is not completely dissociated, so it can absorb hydrogen ions by converting them to undissociated acid. It can also create hydrogen ions by dissociating acid molecules. This means that a weak acid or a weak base has an ability to resist changes in pH. This property is called buffering. The titration curve of a solution that contains weak acids or bases will therefore be less steep than the titration curves of strong acids or bases.

Example 9.7 shows that there will be a severe nonlinearity in the system, due to the titration curve. An additional example illustrates the difficulties in controlling such a system.

Example 9.8  pH control

Consider the problem of controlling the pH of an acid effluent, which is fed to a stirred tank with volume \( V \) (in l) and neutralized with NaOH. Let \( c_A \) (mol/l) be the concentration of acid in the influent stream and \( q \) (l/s) the flow of the effluent. Let \( c_B \) (mol/l) be the concentration of the reagent. Assume that the reagent concentration is so high that the reagent flow \( u \) (l/s) is negligible in comparison with \( q \). The system is modeled by a linear dynamic model which describes the mixing dynamics as if there were no reactions, and a static nonlinear titration curve, which gives pH as a function of the concentrations. Let \( x_A \) and \( x_B \) be the concentrations of acid and base in the tank if there were no chemical reactions. Mass balances then give

\[
\frac{dx_A}{dt} = \frac{q}{V} (c_A - x_A) \\
\frac{dx_B}{dt} = \frac{u}{V} c_B - \frac{q}{V} x_B
\]

(9.21)

The pH is given by Eq. (9.19). It is further assumed that the dynamics of the pH sensor and the pump together can be described by the transfer function

\[
G(s) = \frac{1}{(1 + sT)^2}
\]

A simple calculation indicates the difficulties in the control problem. Assuming proportional control with gain \( k \), the linearized loop transfer function from the error in pH to pH becomes

\[
G_0(s) = \frac{c_B k f'}{q(1 + sT_m)(1 + sT)^2}
\]

where \( T_m \) is the mixing time constant

\[
T_m = \frac{V}{q}
\]
Figure 9.9  Output pH and control signal when controlling the process in
Example 9.8 using a PI controller when pH  \( f \) is (a) 7; (b) 8; (c) 9.

and  \( f' \) is the slope of the titration curve given by Eq. (9.20). The critical
gain for stability is

\[
k_c = \frac{q}{f'c_B T} \left(2 + \frac{T}{T_m}\right) \left(1 + \frac{T}{T_m}\right) \approx \frac{2q}{f'c_B T}
\]

where the approximation holds for \( T \ll T_m \). Since the slope of the titration
curve varies drastically with pH, the critical gain will vary accordingly.
Some values are listed below for different values of the pH of the mixture.

<table>
<thead>
<tr>
<th>pH</th>
<th>Critical gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>0.009</td>
</tr>
<tr>
<td>8</td>
<td>0.046</td>
</tr>
<tr>
<td>9</td>
<td>0.46</td>
</tr>
<tr>
<td>10</td>
<td>4.6</td>
</tr>
</tbody>
</table>

To make sure that the closed-loop system is stable for small perturbations
around an equilibrium of pH = 7 the gain should thus be less than 0.009.
A reasonable value of the gain for operation at pH = 8 is  \( k = 0.01 \), but
this gain will give an unstable system at pH = 7 and is too low for a
reasonable response at pH = 9. Figure 9.9 shows PI control with gain
0.01 and reset time 1. The process is started at equilibrium pH = 4. The
reference value is then changed to 7, 8, and 9, respectively.
Figure 9.10  Control configuration for the pH control problem in Example 9.8.

The calculations and the simulation illustrate the key problems with pH control. The difficulties are compounded by the presence of time delays and flow variations. One way to get around the problem is to use the concentration \( x \) as the output rather than \( \text{pH} \). Figure 9.10 shows a possible control scheme where the measured \( \text{pH} \) and the reference value of \( \text{pH} \) are transformed into equivalent concentrations. This means that the variable \( x \) is computed for the measured \( \text{pH} \) by the formula

\[
x = f^{-1}(\text{pH}) = 10^\text{pH-14} - 10^{-\text{pH}}
\]

(9.22)

The transfer function from \( u \) to \( x \) is

\[
\frac{c_B}{q(1 + sT_m)(1 + sT)^2}
\]

which is independent of the operating point. Figure 9.11 shows the same experiments as in Fig. 9.9, but with the control modification shown in Fig. 9.10. It should be noted that the nonlinear compensation with Eq. (9.18) can be used, since a strong acid-base pair is controlled. The more general problem of mixtures of many weak acids and bases does not have an easy linearizing transformation. It is then necessary to measure the concentrations of the components or to make an on-line measurement of the titration curve. Some form of adaptation can then be reasonable.

\[\square\]

Combustion Control

In combustion control of a boiler it is important to adjust the oxygen content of the flue gases. The flow of combustion air depends on the burn rate in the boiler. The measurement signal is the oxygen content in the exhaust stack, and the control signal is the trim position, which controls the flow of combustion air. There is a significant time delay between the input to the burner and the oxygen sensor in the exhaust stack. With a
Figure 9.11 The same experiment as in Fig. 9.9, but with the controller structure in Fig. 9.10. The gain of the controller is 1000 and the reset time is 1. (a) $\text{pH}_{\text{ref}} = 7$; (b) $\text{pH}_{\text{ref}} = 8$; (c) $\text{pH}_{\text{ref}} = 9$.

conventional controller there is then a loss of efficiency before the correct trim position is reached after a change in the burn rate. One configuration based on adaptive feedforward and gain scheduling is shown in Fig. 9.12. The working range of the boiler is divided into regions. For each region

Figure 9.12 Adaptive feedforward and gain-scheduling an oxygen trim controller.
there is a memory (digital integrator). All integrators are zero initially. When the boiler starts to operate the trim control will adjust the oxygen setpoint. When the setpoint level is achieved, the appropriate integrator is set to the correct trim position. A trim profile will be built up as the boiler works over its range. When the boiler returns to a position at which the integrator is set, the stored trim value is instantly fed to the trim drive actuator thus eliminating the lag from the control loop. If the fuel changes the trim profile is updated automatically. The controller thus works with an adaptive feedforward compensation from the burn rate. There is also a gain scheduling of the loop gain of the controller in order to get tight control under all firing conditions. This gain schedule is built up when commissioning the controller.

**Fuel-Air Control in a Car Engine**

A schematic drawing of a microcomputer control system for a car engine is shown in Fig. 9.13. The accelerator is connected to the throttle valve. The fuel injection is governed by a table look-up controller. The control variable, which is the opening time for the fuel injection valve, is controlled by a combination of feedforward and feedback. The feedforward signal is a nonlinear function of engine speed and load. The load is represented by the air flow, which can be measured using a hot wire anemometer. In one common system the table has $16 \times 16$ entries with linear interpolation.
There is also feedback in the system from an exhaust oxygen sensor. The fuel-air ratio is measured using a zirconium oxide catalytic sensor called the lambda sond. This sensor gives an output that changes drastically when the fuel-air ratio is 1. A typical sensor characteristic is shown in Fig. 9.14. The lambda sond is positioned after the exhaust manifold in an excess oxygen environment, where the exhaust gas from all the cylinders is mixed. This creates a delay in the feedback loop. Notice the feedforward path via the table discussed above. The feedback has a special form; continuous control cannot be used because of the strongly nonlinear characteristics of the lambda sond. The error signal is formed by normalizing the output of the lambda sensor as follows

\[
e = \begin{cases} 
1 & \text{if } V > 0.5 \\
-1 & \text{if } V \leq 0.5 
\end{cases}
\]

The error signal is thus positive if the fuel-air ratio is low (lean mixture) and negative when the ratio is high (rich mixture). The error signal is sent to a PI regulator whose gain and integration time are set from the scheduling table. The values are set based on load (air flow) and engine speed. The gain schedule is implemented simply by adding entries for the gain and integration time to the table used for feedforward of the nominal control variable. Because of the relay characteristic, there will be an oscillation in the fuel-air ratio. This is beneficial, because the catalytic sensor needs a variation to operate properly. The amplitude and the frequency of the oscillation are determined by the parameters of the regulator.
Figure 9.15 Simplified block diagram of the pitch control of the autopilot for a supersonic aircraft.

Flight Control Systems

Figure 9.15 shows a block diagram of the pitch channel of a flight control system for a supersonic aircraft. There are three scheduling variables: height $H$, indicated airspeed $V_{IAS}$, and Mach number $M$. The parameters of the regulator that are scheduled are drawn as boxes, with arrows indicating the scheduling variables. The schedule for the gain $K_{QD}$ is given by

$$K_{QD} = K_{QD_{1A}} + (K_{QD_{H}} - K_{QD_{1A}})MF$$

where $K_{QD_{1A}}$ is a function of indicated airspeed $V_{IAS}$ (shown in Fig. 9.16) and $K_{QD_{H}}$ is a function of height (also shown in Fig. 9.16). The variable $MF$ is given by

$$MF = \frac{1}{s+1}K_{MF}$$

where $K_{MF}$ is a function of the Mach number and $s$ is the Laplace transform variable.
9.6 Conclusions

Gain scheduling is a good way to compensate for known nonlinearities. With such a scheme the regulator reacts quickly to changing conditions. One drawback of the method is that the design may be time-consuming if it is not possible to use nonlinear transformations or auto-tuning. Another drawback is that the controller parameters are changed in open loop without feedback from the performance of the closed-loop system. This makes the method impossible to use if the dynamics of the process or the disturbances are not known accurately enough.

Problems

9.1 Simulate the tank system in Example 9.2. Let the tank area vary as

\[ A(h) = A_0 + h^2 \]

Further assume that \( a = 0.1A_0 \).

(a) Study the behavior of the closed-loop system when the full gain schedule and when the modified gain schedule is used.

(b) Study the sensitivity of the system to changes in the parameters of the process.

(c) Study the sensitivity of the closed-loop system to noise in the measurement of the level.

9.2 Consider the stirred tank reactor in Example 9.3. Design a fixed sampled-data controller with a fixed sampling period for the system. Compare with a regulator based on the time-scaled model.

Figure 9.16 Scheduling functions. The function \( K_{QDIA} \) is also different for different flight modes.
9.3 A model of a ship is given in Section 9.5. Show that the two scalings suggested in Example 9.6 correspond to the time invariance and the path invariance behavior of the ship.

9.4 The simulations in Fig. 9.9 are done using the model of Eqs. (9.21) and (9.19) with \( q = 1000, V = 1000 \), \( T = 0.1 \), and \( K_w = 10^{-14} \). The controller is a PI controller with gain 0.01 and reset time 1. Verify the simulations in Fig. 9.9 and Fig. 9.11.

9.5 Consider the ship steering problem in Example 9.6. Simulate the closed-loop system and determine the sensitivity with respect to the speed of the ship.

9.6 The controller in Example 9.3 gives a control that is equal when measured in terms of the number of sampling intervals, but not in terms of time. Suggest and test possibilities to get the same time responses independent of the flow through the tank.

References

The use of gain scheduling in aircraft control is discussed in:


A typical application of gain scheduling and compensation of nonlinearities in the process industry is given in:


Nonlinear transformations in a general context were originally discussed using geometric control theory in:


Necessary and sufficient conditions under which transformations from nonlinear to linear systems exist are given in:


A neat application for design of a flight control system for a helicopter was made by:


Applications of the same idea in simpler setting are given in:


Additional examples are given in Chapter 12.
Chapter 10

ALTERNATIVES TO
ADAPTIVE CONTROL

10.1 Why Not Adaptive Control?

In previous chapters it has been shown that adaptive control can be very useful and give good closed-loop performance. However, that does not mean that adaptive control is the universal tool that should always be used; a control engineer should be equipped with a variety of tools and the knowledge of how to use them. A good guideline is to use the simplest control algorithm that satisfies the specifications. Gain scheduling and robust high-gain control should definitely be considered as alternatives to adaptive control algorithms. Gain scheduling was discussed in Chapter 9. This chapter presents other alternatives to parameter adaptive control systems. Section 10.2 treats robust high-gain control. The self-oscillating adaptive system (SOAS) is presented in Section 10.3. This is a special class of adaptive systems with strong ties to high-gain control and auto-tuning.
Relay feedback is a key ingredient of the SOAS. Another class of switching systems, variable structure systems, is discussed in Section 10.4. Variable structure systems have been developed mainly in the Soviet Union and can be regarded as a generalization of the SOAS. Conclusions are given in Section 10.5.

10.2 Robust High-gain Feedback Control

Some design methods deal explicitly with process uncertainties. One powerful method has been developed by Horowitz. This procedure, which has its origin in Bode's classical work on feedback amplifiers, is based on several ideas. The specifications are expressed in terms of the transfer function from command signal to process output. The plant is characterized by its nominal transfer function. For each frequency it is also assumed that the process uncertainty is known in terms of variations in amplitude and phase. A solution is determined in terms of a regulator with a feedback $G_{fb}$ and a feedforward $G_{ff}$, as shown in Fig. 10.1. Such a configuration is called two-degree-of-freedom system, because there are two transfer functions to be determined.

Several other design methods can be used to design robust controllers. One technique is based on LQG design. By adjusting the weighting matrices in the LQG problem, a loop transfer recovery (LTR) is achieved. This design procedure can cope with phase uncertainty at high frequencies. The key idea is to keep the loop gain less than 1 at high frequencies, where the phase error is large.

In Horowitz's procedure the feedback transfer function $G_{fb}$ is first determined such that the closed-loop uncertainty is within the specified limits. The nominal value of the transfer function is then modified by the feedforward compensation $G_{ff}$. It is based on graphical constructions using the Nichols chart. It gives a high-order linear compensator that can cope with the specified plant uncertainty. The procedure attempts to keep the loop gain as low as possible. A key idea in the Horowitz design method
is the observation that a system in which the Nyquist curve is close to a straight line through the origin can tolerate a significant change of gain. The response time will change with the gain, but the shape of the response will remain invariant. For minimum-phase systems with a known pole excess it is always possible to find a frequency range in which the phase is constant. By proper compensation it is then possible to obtain a loop gain at which the Nyquist curve is close to a straight-line segment. The assumption that the pole excess is known implies that the phase of the system is known for high frequencies. This is not always a realistic assumption. The Horowitz design method was originally developed for structured perturbations but has also been extended to unstructured uncertainties.

The main step in the procedure is to determine the tolerances for the gain in the closed-loop transfer function. The plant uncertainties are specified as gain and phase variations of the plant transfer function at different frequencies. The given tolerances and uncertainties are used to calculate constraints for the open-loop transfer function. The feedback compensator $G_{fb}$ is then designed such that the compensated open-loop system satisfies the tolerances. This is usually an iterative procedure, which can conveniently be done graphically using a Nichols chart. Finally, the prefilter $G_{ff}$ is designed such that the closed-loop specifications are fulfilled. This may be done using the Bode diagram.

The major drawback of the method is that it is impossible to know a priori whether the desired closed-loop specifications are attainable. It is thus a trail-and-error method, but the iterations give the designer insight into the tradeoffs between different specifications, such as closed-loop sensitivity, complexity of the regulator, and measurement noise amplification.

The properties of a two-degree-of-freedom system are illustrated by an example.

**Example 10.1 An industrial robot arm**

A simple model of a robot arm is used in this example. The transfer function from the control input (motor current $I$) to measurement output (motor angular velocity $\omega$) is

$$G_p(s) = \frac{k_m(J_0s^2 + ds + k)}{J_0J_ms^3 + d(J_a + J_m)s^2 + k(J_a + J_m)s}$$

with $J_a \in [0.0002, 0.002]$, $J_m = 0.002$, $d = 0.0001$, $k = 100$, and $k_m = 0.5$. The moment of inertia $J_a$ of the robot arm varies with the arm angle. Bode plots of the plant gain for the extreme values of the arm inertia $J_a$ are given in Fig. 10.2. The purpose of the control system is to control the angular velocity step responses at various arm angles. The aim is to get a closed-loop system with a bandwidth between 15 and 40 Hz. The disturbance rejection specification has been set to 6 dB. A feedback compensator that
satisfies the specifications is
\[ G_{fb}(s) = \frac{125(1 + s/50)(1 + s/300)}{s(1 + s/800)(1 + s/5000)} \]

This compensator is essentially a PI regulator with a lead filter. The final prefilter has the transfer function
\[ G_{ff}(s) = \frac{1 + s/1000}{(1 + s/26)(1 + s/200)(1 + s/200)} \]

Simulated responses are shown in Fig. 10.3 and Fig. 10.4.

To make a comparison an adaptive controller is also designed for the process. In this particular problem the essential uncertainty is in one parameter only, the moment of inertia. It is then natural to try to make a special adaptive design in which only this parameter is estimated.

The adaptive regulator is designed based on a simplified model. Neglecting the elasticity in the robot arm, the system can be described by

\[ J \frac{d\omega}{dt} = k_m I \]  \hspace{1cm} (10.1)

where \( J = J_a + J_m \) is the total moment of inertia and \( k_m \) the current gain of the motor. The plant of Eq. (10.1) can be controlled adequately with a PI controller. Compare Example 2.4. The regulator parameters can be chosen as in Eq. (2.10). This gives the following characteristic equation for the closed-loop system:
\[ s^2 + 2\zeta_0\omega_0 s + \omega_0^2 = 0 \]
Figure 10.3 Simulation of the step response with the arm inertia $J_a = 0.0002$ for the robust system.

Figure 10.4 Simulation of the step response with the arm inertia $J_a = 0.002$ for the robust system.
The regulator parameters are thus related to the model by simple equations. Notice that the integration time \( T_i \) does not depend on the moment of inertia and that the regulator gain should be proportional to the moment of inertia.

A root-locus calculation indicates that the design based on the simplified model will work well if

\[
\omega_0 < \omega_{\text{crit}} = \zeta_0 \left( \frac{k J_m}{J_a^2} \right)^{1/2}
\]

The most critical case occurs for \( J_a = 0.002 \). It implies that \( \omega_0 \) must be less than 200 rad/s.

The fact that the design is based on a simplified model limits the closed-loop bandwidth. A fast response to command signals can still be obtained by use of feedforward compensation. For this purpose let the desired response to angular velocity commands be given by

\[
G_m(s) = \frac{\omega_m^2}{s^2 + 2\zeta_0 \omega_m s + \omega_m^2}
\]

The feedforward controller can now be designed such that the closed-loop system gets the desired response.

An adaptive system can be obtained simply by estimating the total moment of inertia by applying recursive least squares to the model of Eq. (10.1) and feeding the estimate into the above design equation. To estimate the parameters of the continuous-time model of Eq. (10.1), it is necessary to introduce filtering. This is done by integrating Eq. (10.1) over the time interval \((t, t + h)\):

\[
\omega(t + h) - \omega(t) = \frac{k_m}{J} \int_t^{t+h} I(s) \, ds
\]

A least-squares estimator of \( J \) is easily constructed from this equation. This estimate is then used in the PI control law. A simulation of the system is shown in Figs. 10.5 and 10.6. The parameter \( h \) was chosen to 0.1 s in the simulations. The figures show that the system adapts to a good response after two transients. Notice the magnitudes of the control signal for the cases of low and high inertia.

The regulator structures for the robust and the adaptive cases are quite similar by design. The feedback part of the robust regulator is essentially a PI regulator with a lead-lag filter. The parameters are \( K = 2.5 \) and \( T_i = 0.02 \). The lead-lag filter increases the regulator gain to 6.7 at frequencies around 500 rad/s. The feedback part of the adaptive regulator is also a PI regulator, but the parameters are adjustable. They range from \( K = 0.15 \) and \( T_i = 0.07 \) for \( J_a = 0.0002 \) to \( K = 1.05 \) and \( T_i = 0.07 \) for
Figure 10.5 Simulation of the tailored adaptive systems response, with the arm inertia $J_a = 0.0002$. The regulator parameters are initially tuned for $J_a = 0.002$. The feedback gain in the adaptive regulator is thus 40 times smaller than the gain of the robust regulator. This means that the effects of measurement noise are also much smaller for the adaptive regulator.

Both systems are designed to give the same response time to command signals. Notice, however, that feedforward is used in very different ways in the two systems. In the robust design it is used to increase the response time to command signals, while it is used to decrease the response time in the adaptive design. The reason is that the bandwidth of the closed inner loop is large in the robust design, to take care of the plant variations, while the adaptive design allows a low closed-loop bandwidth since the uncertainty is eliminated.

The responses of the adaptive system are better over the full parameter range when the parameters are adapted, but it will take some time for the parameters to adapt. The robust regulator will have a better response when the parameters of the process are changing rapidly from one constant value to another.

Comparison between Robust and Adaptive Control

The robust design method will generally give systems that respond faster when the parameters change, but it is important that the range of param-
eter variation be known. The adaptive regulator responds more slowly but can generally handle larger parameter variations. The adaptive controller will give better responses to command signals and load variations when regulator parameters have converged, provided that the model structure is sufficiently correct. The controllers designed by Horowitz's method will generally have high-loop gains, which make them more sensitive to noise.

10.3 The Self-oscillating Adaptive System

A system that is insensitive to parameter variations can be obtained by using a two-degree-of-freedom configuration with a high-gain feedback and a feedforward compensator (compare Section 10.2). This section introduces an adaptive technique to keep the gain in the feedback loop high using a relay feedback. Relays combine the properties of high gain and inexpensive implementations. However, relays often introduce oscillations into the system.

The idea of the self-oscillating adaptive system (SOAS) originated in work at Honeywell on adaptive flight control in the late 1950s. The inspiration came from work on nonlinear systems by Flügge-Lotz at Stanford.
Systems based on the idea were flight-tested in the F-94C, the F-101, and the X-15 aircraft. The idea has also been applied in process control, but the SOAS has not found widespread use. One reason is that substantial modifications of the basic scheme are necessary to make the systems work well. A characteristic feature of the SOAS is that there is a limit cycle oscillation. The system thus represents a type of adaptive control in which there are intentional perturbations, which excite the system all the time. The SOAS is one of the simplest systems with this property.

The SOAS is based on three useful ideas: model-following, automatic generation of test signals, and use of a relay with a dither signal as a variable gain. The key result is that the loop gain is automatically adjusted to give an amplitude margin $A_m = 2$.

**Principles of the SOAS**

Since we want to emphasize the ideas we will limit the discussion to the basic version of the system. A block diagram of an SOAS is shown in Fig. 10.7. This is a two-degree-of-freedom system. There is a high-gain feedback loop around the process. The desired response to command signals is obtained by the reference model. Ideally, the high-gain loop will make the process output $y$ follow the model output $y_m$. The response of the closed-loop system will be relatively insensitive to the variations in process dynamics because of the high loop gain. The system is thus a typical model-following design. The special feature is that the high-gain loop is nonlinear.

**The High-gain Loop**

The feedback compensator contains a lead filter $G_f(s)$ and a relay. The relay is motivated by the desire to have as high a gain as possible. Because
of the relay, there will be a limit cycle, whose amplitude is kept at a tolerable limit by adjusting the relay amplitude by a separate feedback loop. The relay gives a high gain for small inputs and the gain decreases with increasing input amplitude. The key difficulty in the design of an SOAS is to find a suitable compromise between the limit cycle amplitude and the response speed. A low relay amplitude gives a limit cycle with a low amplitude but also a slow response speed. A large relay amplitude gives a rapid response but also a large amplitude of the limit cycle oscillation. The relations can to some extent be influenced by the lead filter.

Properties of the Basic SOAS

The limit cycle in a system with relay feedback was discussed in Section 8.5. This will now be used to analyze the self-oscillating adaptive system. Consider the system shown in Fig. 10.7 without the gain changer.

The relay is used in order to introduce a limit cycle oscillation in the system. The period and the amplitude of the oscillation can be determined by the methods discussed in Section 8.5. When the reference signal is changed, or when there are disturbances, there will also be other signals in the system, which will be superimposed on the limit cycle oscillations. The signals that appear in the system will thus be of the form

\[ s(t) = a \sin \omega t + b(t) \]

where \( a \sin \omega t \) denotes the limit cycle oscillation. The key to understanding the SOAS is to find out how signals of this type propagate in the system. It is straightforward to determine the transmission of the signal through the linear subsystems; the signal propagation through the relay is the main difficulty. This analysis will be simplified considerably if it is assumed that \( b(t) \) varies much more slowly than \( \sin \omega t \). Furthermore, assume that \( b(t) \) is smaller than \( a \). This should be true at least in steady state, since \( b(t) \) is the difference between the model output and the process output.

The Dual-input Describing Function

It is assumed that \( b(t) \) varies so slowly that it can be approximated by a constant. The input signal to the relay is thus of the form

\[ u(t) = a \sin \omega t + b \]

The relay input and output are shown in Fig. 10.8. The relay output can be expanded in a Fourier series

\[ y(t) = bN_R + aN_A \sin \omega t + aN_{A2} \sin 2\omega t + \cdots \]  \hspace{1cm} (10.2)
where the numbers $N_A$ and $N_B$ are given by

$$N_B = \frac{1}{2\pi b} \int_0^{2\pi} y(t) \, dt = \frac{d(\pi + \alpha) - d(\pi - 2\alpha) + \alpha d}{2\pi b}$$

$$= \frac{4\alpha d}{2\pi b} = \frac{2\alpha d}{\pi b} = \frac{2d}{\pi b} \sin^{-1}\left(\frac{b}{a}\right)$$

$$N_A = \frac{1}{\pi a} \int_0^{2\pi} y(t) \sin \omega t \, dt = \frac{2d}{\pi a} \int_{\alpha}^{\pi - \alpha} \sin \omega t \, dt$$

$$= \frac{4d}{\pi a} \cos \alpha = \frac{4d}{\pi a} \sqrt{1 - (b/a)^2}$$

Small values of $b$ give the approximations

$$N_A \approx \frac{4d}{\pi a} \quad N_B \approx \frac{2d}{\pi a}$$

Notice that

$$N_A \approx 2N_B \quad (10.3)$$

The transmission of the constant level $b$ and of the first harmonic $\sin \omega t$ are thus characterized by the equivalent gains $N_B$ and $N_A$. Since the linear parts will normally attenuate high frequencies more than low frequencies, a reasonable approximation is often obtained by considering only the constant part and the first harmonic. The number $N_B$, which describes the propagation of a constant signal, is called the dual-input describing
function, by analogy with the ordinary describing function that describes the propagation of sinusoids through static nonlinearities. Notice that the describing function $N_B$ depends on $a$ (the amplitude of the sinusoidal oscillation). This dependence is the key to understanding how the SOAS works.

The dual-input describing function can be used to characterize the transmission of slowly varying signals. A detailed analysis of the accuracy of the approximation is fairly complicated. Let it therefore suffice to mention some rules of thumb for using the approximation. It is recommended that the ratio $a/b$ should be greater than 3 and that the ratio of the limit cycle frequency to the signal frequency should also be greater than 3. It is strongly recommend that the analysis be supplemented by simulation.

**Main Result**

The tools for explaining how the SOAS works are now available. Consider the system in Fig. 10.7. From Section 8.4, the period of the limit cycle is given by Eq. (8.8) when using the describing function method. The amplitude of the limit cycle at the relay input is also given by Eq. (8.8):

$$N_A|G(i\omega_u)| = 1$$  \hspace{1cm} (10.4)

The transmission of a sinusoidal signal through a relay can thus be approximately described by an equivalent gain, which is inversely proportional to the signal amplitude at the relay input. The amplitude thus automatically adjusts so that the loop gain is unity at the frequency $\omega_u$.

Now consider the propagation of slowly varying signals superimposed on the limit cycle oscillations. The propagation of the signals through the linear parts of the system can be described by the transfer function $G(s)$. If the signals vary slowly in comparison with the limit cycle oscillations, the propagation through the relay is approximately described by the dual-input describing function $N_B$. The propagation of slowly varying signals is thus approximately described by the loop transfer function

$$G_0(s) = N_B(a)G(s)$$

It follows from Eqs. (10.3) and (10.4) that

$$|G_0(i\omega_u)| = N_B(a)|G(i\omega_u)| = \frac{1}{2}N_A|G(i\omega_u)| = 0.5$$

We thus obtain the following important result which describes the operation of the SOAS.
Result 10.1  Amplitude margin of SOAS
The SOAS automatically adjusts itself so that the response to reference signals is approximately described by the closed-loop transfer function

\[ G_c(s) = \frac{kG(s)}{1 + kG(s)} \]

where the gain \( k \) is such that the amplitude margin is 2. \( \square \)

This result explains the adaptive properties of the SOAS. The result can also be stated in the following way: The relay acts as a variable gain. The magnitude of the gain depends on the amplitude of the sinusoidal signal at the relay input. This gain is automatically set by the limit cycle oscillation to such a value that the loop gain becomes 0.5 at the frequency of the limit cycle.

The result is illustrated by an example.

Example 10.2—Basic SOAS
Assume that the linear parts are characterized by the transfer function

\[ G(s) = \frac{ka}{s(s + 1)(s + a)} \]

From Example 8.1, the period of the limit cycle is approximately given by

\[ \omega_u = \sqrt{a} \]

The magnitude of the transfer function at this frequency is

\[ |G(i\omega_u)| = \frac{k}{a + 1} \]

If the relay amplitude is \( d \) it follows that the amplitude of the limit cycle oscillation at the relay input is approximately given by

\[ e_0 = \frac{kd}{1 + a} \]

The limit cycle amplitude is thus inversely proportional to \( a \). A simulation of the system is shown in Fig. 10.9. The feedforward transfer function is a second-order system with the damping 0.7 and the natural frequency 1 rad/s. The nominal values of the parameters are \( k = 3 \), \( d = 0.35 \), and \( a = 20 \). The approximate analysis gives a limit cycle with period \( T = 1.4 \) and amplitude 0.05. The process gain is suddenly increased by a factor of 5 at \( t = 25 \). Notice the rapid adaptation. The amplitude of the oscillation
will, however, also increase by a factor of 5. If the value of $d$ is chosen such that the error would be 0.05 for the higher value of $k$, then the system becomes too slow for small $k$.

Design of an SOAS

The self-oscillating adaptive system is a simple nonlinear feedback system that is capable of adapting rapidly to gain variations. The system has a continuous limit cycle oscillation. This is not suitable when valves or other mechanical parts are used as actuators. An SOAS may, however, conveniently be used with thyristors as actuators. The presence of the limit cycle oscillation may also cause other inconveniences. Since the system will automatically adjust to an amplitude margin $A_m = 2$, it is also necessary that the characteristics of the process be such that this design principle gives suitable closed-loop properties. The key problem in the design of the SOAS is the compromise between the limit cycle amplitude and the response speed. This compromise is influenced by the selection of the linear compensator, $G_f(s)$, and of the relay amplitude. Compare Fig. 10.7. The design for an SOAS can be described by the following procedure.

Step 1: The relay amplitude is first determined such that the desired control authority (tracking rate, force, speed, etc.) is obtained. This can be estimated by analyzing the response of the process to constant control signals.
Step 2: When the relay amplitude is specified, the desired limit cycle frequency can be determined from the condition

\[ d |G_p(i\omega_u)| = \epsilon_0 \]

where \( \epsilon_0 \) is the tolerable limit cycle amplitude in the error signal and \( G_p(s) \) is the transfer function of the process. It is necessary to check that the frequency obtained is reasonable. For example, the frequency \( \omega_u \) may become so high that the process dynamics become uncertain.

Step 3: The final step is to determine the transfer function \( G_f \) of the linear compensator such that

\[ \arg G_f(i\omega_u) + \arg G_p(i\omega_u) = -\pi \]

A large phase lead may be necessary, but this may not be realizable because of noise sensitivity.

Step 4: Check that the linear closed-loop system with the loop gain \( G_0 = kG_fG_p \) will work well when the gain is adjusted so that the amplitude margin is 2. If this is not the case, the compensator \( G_f \) must be modified.

Notice that it is necessary to have an estimate of the magnitude of the process transfer function in Steps 1 and 2. Knowledge of the phase curve of the process transfer function is necessary in the third step. Also notice that it may not be possible to resolve the compromises in all steps. It is then necessary to add additional loops for changing the gain.

Gain Changers

External feedback loops, which adjust the relay amplitude, may be used to resolve the compromise between a high tracking rate and a small limit cycle amplitude. The so called up-logic used in the first SOAS can be described as follows

\[ d = \begin{cases} 
  d_1 & \text{if } |e| > e_l \\
  d_2 + (d_1 - d_2)e^{-(t-t_0)/T} & \text{if } |e| < e_l 
\end{cases} \]

The time \( t_0 \) is the last time that \( |e| < e_l \). The relay amplitude is thus increased to \( d_1 \) when the error exceeds a limit \( e_l \). The relay amplitude then decreases to a lower level \( d_2 \) when the error is less than \( e_l \). This gain changer increases the relay amplitude and the response rate when large reference signals are applied.

Another type of gain changer has been used to control the amplitude of the limit cycle. The limit cycle amplitude at the process output is measured by a band-pass filter and a rectifier. The relay amplitude is
then adjusted to keep the limit cycle amplitude constant at the process output.

**Dither Signals**

In some applications it is desirable to avoid the limit cycle. One idea that has been used successfully is to introduce a variable gain after the relay. The gain is adjusted so that the limit cycle vanishes. In the early applications it was difficult to implement multiplications. A trick that was used to implement the multiplication is illustrated in Fig. 10.10. A high-frequency triangular wave is added to the signal before the relay. With low-pass filtering the average effect of the additive triangular signal is the same as a multiplication by a constant. The constant is inversely proportional to the amplitude of the triangular wave. The triangular wave is called a *dither signal*. Use of a dither signal is an illustration of the idea that an oscillation may be quenched by another high-frequency oscillation.

**Example 10.3—SOAS with lead network and gain changer**

The relay control in Example 10.2 gave an error amplitude of about $e_0 = 0.03$. Assume that we want to decrease the amplitude by a factor of 3 while maintaining $d = 0.35$. This gives a new oscillation frequency $\omega'_u$ such that

$$d \ G(i\omega'_u) = 0.01$$

or $\omega'_u = 10$ rad s. To get this oscillating frequency a lead network $G_f$ is
added such that
\[
\arg G_f(i\omega'_u) + \arg G_p(i\omega'_u) = -\pi
\]

Figure 10.11 shows a simulation of the system in Example 10.2 with the compensation network
\[
G_f(s) = 1.2 \frac{s + 5}{s + 15}
\]

As in Fig. 10.9, the gain is increased by a factor of 5 at \( t = 25 \). It is seen that the lead network decreases the amplitude of the oscillation while maintaining the response speed. To speed up the response we can introduce the up-logic for the gain. Figure 10.12 shows a simulation in which \( d_1 = 0.5 \), \( d_2 = 0.1 \) and \( e_t = 0.1 \). The error signal is decreased, but there is still an oscillation. The behaviour of the closed-loop system can be sensitive to the choice of the parameters in the gain changer. Too large a value of \( d_1 \) will cause the error to be larger than \( e_0 \), and there will not be any decrease in \( d \) nor in the amplitude of the error. The oscillation can be quenched by adding a dither signal at the input of the process.

**Externally Excited Adaptive Systems**

A system closely related to the SOAS is obtained by injecting a high-frequency sinusoid to measure the gain of the process and to set the regulator gain. Such a system is called an *externally excited adaptive system*.
(EEAS) and gives the designer more freedom than the SOAS, because the frequency of the excitation can be chosen more easily. This system is used for track-keeping in compact disc recorders. The main source for the parameter variation is a gain variation in the laser diode system.

Summary

The basic SOAS is simple to implement and can cope with large gain changes in the process. Result 10.1 shows that the SOAS will automatically adjust itself so that the amplitude margin is 2. However the limit cycle in the SOAS is noticeable and can be disturbing. The introduction of lead network, gain changer, and dither can decrease the amplitude of the oscillation. The EEAS is a similar system in which a high frequency signal is introduced externally.

10.4 Variable-structure Systems

In Section 10.2 it was shown how fixed robust controllers can be obtained by increasing the complexity of the controller. A special class of relay control, called a variable structure system (VSS), is another way to achieve robust control. Such systems are extensions of relay controllers discussed
Figure 10.13 The trajectories of Eq. (10.7) at the switching surface.

previously in this chapter. The name "variable-structure" alludes to the fact that the controller structure may be changed.

Consider a linear single input, single output system described by

$$\dot{x} = Ax + Bu$$  \hspace{1cm} (10.5)

A conventional constant-gain state feedback controller has the structure

$$u(t) = -Lx(t)$$  \hspace{1cm} (10.6)

In a variable structure system the feedback gain $L$ is different in different parts of the state space. Such systems have interesting properties and can in certain circumstances be made insensitive to variations in the process dynamics.

**Sliding Modes**

Systems with switching gains will exhibit interesting behavior. Consider the case in which the feedback law switches on the surface

$$\sigma(x) = 0$$

Assume that the closed-loop system is described by

$$\dot{x} = \begin{cases} f & \text{if } \sigma(x) > 0 \\ f^- & \text{if } \sigma(x) < 0 \end{cases}$$  \hspace{1cm} (10.7)

where the switching is such that the functions $f$ and $f^-$ have the property indicated in Fig. 10.13. The behavior of the system can be described as follows. The vector field will drive the state towards the surface $\sigma(x) = 0$. The control will change rapidly from one value to another on the switching surface. This is called *chattering*. The net effect is that the state will move towards the surface $\sigma(x) = 0$ and then slide along the surface. This sliding motion can be described as follows. Let $f_n$ denote the projection
of $f$ on the normal of the surface $\sigma(x) = 0$. Introduce a number $\alpha$, such that
\[
\alpha f^+_n + (1 - \alpha) f^-_n = 0
\]
The sliding motion is then given by
\[
\dot{x} = \alpha f^+ + (1 - \alpha) f
\]

**Robust Control**

In a variable-structure system we attempt to find a switching surface such that the closed-loop system behaves as desired. There are three important questions that must be answered for VSS:

- Will the trajectories starting at any point hit the switching line?
- Is there a sliding mode?
- Is the sliding mode stable?

There are partial answers to these questions in the literature on VSS.

A sliding mode that is insensitive to the parameters of the system can be constructed as follows. Assuming that the system is controllable and has one input, it can then be transformed to the controllable canonical form

\[
\dot{x} = \begin{pmatrix}
-a_1 & -a_2 & \cdots & -a_n & 1 & -a_n \\
1 & 0 & 0 & 0 & 0 \\
\vdots & & & & & \\
0 & 0 & \cdots & 1 & 0 \\
\end{pmatrix} x + \begin{pmatrix}
1 \\
0 \\
\vdots \\
0 \\
\end{pmatrix} u \quad (10.8)
\]

Let the switching surface be
\[
\sigma(x) = p_1 x_1 + p_2 x_2 + \cdots + p_n x_n = p^T x = 0
\]

On this we have
\[
\frac{d^{n-1}}{dt^{n-1}} \sigma(x) = p_1 x_1^{(n-1)} + p_2 x_2^{(n-1)} + \cdots + p_n x_n^{(n-1)}
\]

It follows from Eq. (10.8) that
\[
x_2^{(n-1)} = x_1^{(n-2)} \\
x_3^{(n-1)} = x_1^{(n-3)} \\
\vdots \\
x_n^{(n-1)} = x_1
\]
The sliding mode is thus characterized by

$$p_1 \frac{d^n}{dt^n} x_1 + p_2 \frac{d^{n-2}}{dt^{n-2}} x_1 + \cdots + p_n \frac{dx_1}{dt} + p_n x_1 = 0$$

The dynamic behavior on the sliding surface can be specified by a proper choice of the numbers $p_i$. The motion is determined by a differential equation of order $n - 1$. It will be stable if the polynomial

$$P(s) = p_1 s^{n-1} + p_2 s^{n-2} + \cdots + p_n$$

has all its roots in the left-half plane. The sliding mode is insensitive to variations in $A$ and $B$ of Eq. (10.5).

One way to design a variable-structure system is to first transform to controllable canonical form and choose a stable switching surface in the transformed variables. The system and the switching criteria can then be transformed back to the original state variables.

It remains to show that the switching of the feedback law can be constructed so that the sliding regime $p^T x = 0$ will be reached. This can be done for Eq. (10.8) by introducing the Lyapunov function

$$V(x) = \frac{1}{2} \sigma^2(x) = \frac{1}{2} (p^T x)^2$$

The stability condition is

$$\dot{V} = \sigma(x) \dot{\sigma}(x) = x^T pp^T (A - BL)x < 0$$

Consider the right-hand side

$$x^T p (p^T A - p^T BL)x = x^T p (p^T B) \left( \frac{p^T A}{p^T B} - L \right)x = \sigma(x)(p^T B)(a - L)x$$

where

$$a = \frac{p^T A}{p^T B} \quad (10.9)$$

This gives sufficient conditions for switching the gain so that the derivative of the Lyapunov function is negative. The following conditions on the feedback gain $L$ are sufficient for existence of a sliding mode:

$$l_i > a_i \quad \text{if } \sigma(x)(p^T B)x_i > 0$$

$$l_i < a_i \quad \text{if } \sigma(x)(p^T B)x_i < 0$$
This implies that the feedback gains \( L \) switch on switching surface \( \sigma(x) = 0 \) and on \( n \) planes whose normals are the coordinate axes. Notice that the switching function is nonlinear in the states of the system. In the literature there are also sufficient conditions given; see the references in the end of this chapter. In special cases it is also possible to get fewer switching planes. The method of variable structure is illustrated by the following example.

**Example 10.4—Second-order VSS**

Consider the unstable system

\[
\begin{align*}
\dot{x}_1 &= x_1 + u \\
\dot{x}_2 &= x_1 \\
y &= x_2
\end{align*}
\]

which has the transfer function

\[
G(s) = \frac{1}{s(s - 1)}
\]

Choose the switching

\[
\sigma = p_1 x_1 + p_2 x_2 = x_1 + x_2
\]

which means that the motion on the switching line is governed by \( \dot{\xi} + \xi = 0 \) where \( \xi \) is a coordinate along the line. The parameter \( a \) given by Eq. (10.9) is

\[
a = \frac{p^T A}{p^T B} = \begin{pmatrix} 2 & 0 \end{pmatrix}
\]

One controller that gives a stable sliding mode is

\[
u(t) = -l_1 x_1(t) - l_2 x_2(t)
\]

where

\[
l_1 = \begin{cases} 
2.5 & \text{if } \sigma(x)x_1 > 0 \\
-2.5 & \text{if } \sigma(x)x_1 < 0
\end{cases}
\]

and

\[
l_2 = \begin{cases} 
0.5 & \text{if } \sigma(x)x_2 > 0 \\
-0.5 & \text{if } \sigma(x)x_2 < 0
\end{cases}
\]

The phase plane of the system is shown in Fig. 10.14. The input and the output for one initial value are shown in Fig. 10.15. Notice that the
Figure 10.14 Phase portrait of the system in Example 10.4.

Figure 10.15 One input output pair for the system in Example 10.4. The initial conditions are $x_1(0) = 1.5$ and $x_2(0) = 0$. 
trajectories hit the switching line $\sigma = 0$ and stay on it. This implies that the control signal will chatter. Along the sliding line $\sigma = 0$ we have

$$\frac{d\sigma}{dt} = \dot{x}_1 + \dot{x}_2 = \dot{x}_1 + x_1 = 0$$

Since the system is in controllable form, the closed-loop behavior is independent of the system parameters at the sliding mode.

\[ \square \]

**Summary**

Variable-structure systems are related to the SOAS. In variable-structure systems we want the system to get into a sliding mode in order to obtain insensitivity to parameter variations. The control signal of variable-structure systems will chatter in the sliding mode. The chatter can be avoided by introducing a relay with a hysteresis, but this will transform the chattering into a limit cycle. The amplitude of the control signal is determined by the magnitude of the state variables or the error. With this modification the variable-structure system can be regarded as an SOAS in which the relay amplitude depends on the states. The switching condition is a linear function of the error in the SOAS, while in variable-structure systems it is a nonlinear function of the states.

The theory on VSS can be extended to controllers, in which the feedback is done from a reduced number of state variables. The conditions will, however, become more complex than those discussed in this chapter. There will be more constraining conditions on the choice of the switching plane. Since the conditions for the existence of a sliding mode depend on the process and the switching plane, there have been attempts to make adaptive VSS by adaptation on $\sigma(x)$.

One main drawback of variable-structure systems is the problem of choosing the switching plane. It also requires measurement of all state variables. Another drawback is the chatter in the control signal in the sliding mode.

**10.5 Conclusions**

Robust high-gain control can be very effective for systems with structured parameter variations, where the range of the variations is known. If the parameter bounds are uncertain, high-gain design methods will lead to a complex and conservative design. Relay feedback is an extreme form of high-gain systems. This chapter has described different ways to use relay feedback to obtain systems that are insensitive to parameter variations. Self-oscillating adaptive systems and variable-structure systems are two
applications of this idea. The SOAS can be designed to work quite well, but it requires engineering effort and some knowledge about the process in order to get a satisfactory performance of the closed-loop system. These drawbacks have resulted in lack of interest in the SOAS. The ideas behind SOAS have, however, become useful in connection with auto-tuning of simple regulators as discussed in Chapter 8.

Problems

10.1 Determine whether the following plants can be stabilized by a linear fixed parameter compensator when \( a \in [-1, 1] \).

(a) \( a/s \);  (b) \( 1/(s + a) \);  (c) \( 1/(1 + as) \);  (d) \( a/(1 + s) \);  (e) \( a/(1 - s) \).

10.2 Consider the process

\[ G_p(s) = e^{-sT} \quad T \in [0, 1] \]

(a) Show that it can be controlled by a controller of the structure in Fig. 10.1 with

\[ G_{fb}(s) = \frac{0.6(1 + s/1.3)}{s(1 + s/2)} \]

\[ G_{ff}(s) = \frac{1 + s}{1 + s/3} \]

(b) Simulate the behavior for changes in the command signal and step disturbances at the output.

(c) Discuss how to make a self-tuning regulator based on pole placement for the process.

10.3 Consider the linear closed-loop system shown in Fig. 10.16 with the same \( G(s) \) as in Example 10.2 and with \( a = 20 \). Determine the gain \( k \) so that the amplitude margin is \( A_m = 2 \). Simulate the system and determine its step response. Compare this with the step response of the corresponding SOAS in Example 10.2.
10.4 Consider a linear plant with the transfer function

\[ G(s) = \frac{k}{s(s + 1)^2} \]

where the gain \( k \) may vary in the range \( 0.1 \leq k \leq 10 \). Determine the relay amplitude \( d \) and a suitable lead network so that the limit cycle amplitude at the process output is less than 0.05 and so that the rise time to a step of unit amplitude is never less than 0.5 s. Simulate the resulting design and verify the results.

10.5 Consider the system in Example 10.2. Experiment with a gain changer of the “up-logic” type. Investigate how a dither signal will influence the performance of the closed-loop system.

10.6 Consider the system in Problem 10.4. Design a gain changer that keeps the limit cycle amplitude at 0.01 for the whole operating range.

10.7 Consider a system with the transfer function

\[ G(s) = \frac{k}{s + 1} \]

where the gain \( k \) may change in the range 0.1 to 10. Design a servo using the SOAS principle so that the closed-loop transfer function is

\[ G(s) = \frac{1}{s^2 + s + 1} \]

independent of the process gain.

10.8 Consider the system in Example 10.4.
(a) Investigate the stability of the closed-loop system if the control signal is fixed to \( u = -\alpha x_1 \) or to \( u = \alpha x_1 \). Sketch the phase plane for the two cases and compare with Fig. 10.14.
(b) Determine the necessary and sufficient conditions for the existence of a sliding mode.
(c) Discuss the influence of \( p_i \) and \( l \cdot \) on the speed of the trajectory along the sliding mode.

10.9 Assume that the system in Eq. (10.5) is in companion form and that the control law is of the form

\[ u = -\sum_{i=1}^{m} l_i x_i \cdot \]

Derive the necessary and sufficient conditions for the existence of a sliding mode. When will the sliding mode be stable?
10.10 Consider the process in Problem 2.7. Design a robust controller for the system. Investigate the disturbance rejection of the closed-loop system.

10.11 Consider the process in Problem 2.8. Design a robust controller for the system. Investigate the disturbance rejection of the closed-loop system.

10.12 Design an SOAS for the system in Problem 2.7 and investigate its properties.

10.13 Design an SOAS for the system in Problem 2.8 and investigate its properties.

References

The robust high-gain design is closely related to early ideas on feedback amplifiers. See:


The basic robust design method for SISO systems, for specifications in the frequency domain is discussed in:


This last paper presents criteria for determining whether a given set of performance specifications are achievable (i.e., when the plant is non-minimum-phase) and, if so, a synthesis procedure is included for deriving the optimum design, defined as that with an effectively minimum-loop transmission bandwidth. Example 10.1 is based on:

Robust design of processes with unstructured uncertainties is treated in:


Research on relay systems was very active in the 1950s and 1960s. An authoritative treatment by one of the key contributors is:


The method of harmonic balance and describing function is extensively treated in:


This book also contains applications of SOAS. The background of SOAS and more about the design rules can be found in:


A modified version of the SOAS was flight-tested extensively in the experimental rocket plane X15. Experiences from that are summarized in:

Thompson, M. O., and J. R. Welsh, 1970. “Flight test experience with adaptive control systems.” *Proc. Agard Conf. on Advanced Control*

The general conditions for stability of the limit cycle in relay systems are still unknown. Some guidance is given by the stability conditions in:


Additional results on relay oscillations are found in:


A procedure for designing externally excited adaptive systems (EEAS) is given in:


Variable-structure systems are treated in:


The sufficient conditions of the existence of a sliding mode is found in:


Adaptive variable-structure systems are discussed in:


An interesting overview of variable-structure systems is given in the survey paper:


It also presents several industrial applications.
Chapter 11

PRACTICAL ASPECTS
AND IMPLEMENTATION

11.1 Introduction

The previous chapters have mainly been devoted to the theoretical aspects of adaptive control algorithms. Since adaptive systems are complex it is essential to make simplifications when developing theory. Considerable insight can be obtained by investigating idealized situations. Theory can also make clear the limits of what can be achieved under idealized situations. However, when developing a practical adaptive system we are faced with many situations that are not covered by the theory. To make a successful system the designer must have considered all situations and combinations of conditions that may occur. The design of the controller is done automatically on-line, so everything must be covered in the algorithm. It is also important to have a "safety net" that makes it possible to cover situations such as start-up shut-down switching between manual and automatic modes etc.
Much of the practical work is not done on a firm theoretical basis but consists of ad hoc solutions which often depend on the application under consideration. They are often verified by extensive experimentation and simulation, because the problems are complex and appropriate theory is not yet available. Some of the key issues of adaptive control are:

- How to use prior information about the process
- How to determine realistic specifications for the closed-loop system
- How to achieve robust estimation
- Unmodeled high-frequency dynamics
- Signal conditioning
- Numerics
- Start-up and bumpless transfer
- Process and actuator nonlinearities

Several of these points are not specific to adaptive control but are also important for implementation of controllers in general. Section 11.2 discusses the implementation of the estimator. Forgetting of old data and filtering of signals are considered. Various aspects of the controller design are covered in Section 11.3. Only discrete-time controllers are discussed in this chapter, since computers are the main tool for implementing adaptive controllers. Some topics are numerical aspects, reset action, anti-reset windup, and choice of sampling interval. Sections 11.2 and 11.3 cover estimation and controller design as two separate tasks. The interaction between them which is very important in adaptive controllers, is discussed in Section 11.4. Section 11.5 presents prototype algorithms and program skeletons that will be very useful for the implementation of adaptive algorithms.

There is a great difference between an academic or theoretical algorithm and an industrially useful algorithm. The implementation must include many features that have proved to be good from a practical point of view but are perhaps not fully covered by the present theory. These practical features can also be hints for improving the theory of adaptive controllers. Examples of such features are also given in Chapter 12 which presents some commercial adaptive systems.

11.2 Estimator Implementation

There are many books and papers discussing parameter estimation; see the references in Chapter 3 and at the end of this chapter. This section concentrates on the practical and numerical aspects. The motto of this section is:
“Use only good data for the updating and don’t throw away useful information.

It will be assumed that the process is described by the discrete-time model

\[ y(t) = G_o(q)u(t) + H_o(q)e(t) + d(t) \]  \hspace{1cm} (11.1)

Notice that possible anti-aliasing filters can be allowed as a part of the process \( G_o(q) \). The transfer function operators \( G_o \) and \( H_o \) depend on the sampling period \( h \). Pre- and postsampling filters are also included in \( G_o \). The disturbance \( e(t) \) is assumed to be a zero mean white noise disturbance. The second disturbance \( d(t) \) is a purely deterministic signal of known form but unknown magnitude. It may be a load, a ramp, or a sinusoidal signal.

Both direct and indirect adaptive algorithms need a recursive estimation scheme. Most common are the method of least squares and its modifications. The basic least-squares algorithm with exponential forgetting was introduced in Chapter 3.

Different aspects and modifications of the basic algorithm will be discussed, guided by the motto of the section. The keys to getting good models are good data and appropriate model structures. The models used will invariably be simplified (e.g., linear and of low order). The physical process will thus have dynamics that is not accounted for in the model used in the adaptive system. This is referred to as unmodeled dynamics. It is known from the theory of system identification that the estimates obtained in the presence of unmodeled dynamics will depend crucially on the properties of the input signal. When determining the structure and complexity of a controller, we always have some more or less well-defined prior knowledge of how the process can be described: for instance, its complexity, amount of time delay, and types of disturbances. The purpose of the estimator part of the adaptive controller is to find the parameters of this prejudice model. In this section we will discuss how to adjust the measured data to the prejudice model.

**Elimination of Known Types of Disturbances**

Disturbances of known types are represented by \( d(t) \) in Eq. (11.1). They can be thought of as generated by pulses into known dynamic systems. It is assumed that it is not known a priori when the pulses occur or their amplitude. Such signals are called piecewise deterministic signals and can be generated by

\[ d(t) = H_d(q)\delta_s(t) = \frac{D_n(q)}{D_d(q)}\delta_s(t) \]

where \( \delta_s(t) \) is a sequence of pulses and \( H_d(q) \) a filter. For instance, if \( d(t) \)
is a piecewise constant signal, then

$$H_d(q) = \frac{q}{q - 1}$$

The signal $d(t)$ can be eliminated (except for occasional pulses or finite length disturbances) by filtering the input and the output with $D_d(q)$.

The disturbance annihilation filter $D_d(q)$ is of band-pass or notch type. It can then be advisable to use a filter of the form

$$H_f(q) = \frac{D_d(q)}{D_d'(q)} \quad (11.2)$$

where $D_d'(q)$ is a stable polynomial chosen such a way that $H_f$ has a high attenuation at high frequencies. Filtering with $H_f$ will then remove both low and high frequencies in the signal and eventually some other frequencies within a narrow band. The filter $H_f(q)$ should thus have an amplitude curve as in Fig. 11.1. The lower break frequency $\omega_{fl}$ depends on the desired crossover frequency of the closed-loop system. As a rule of thumb, $\omega_{fl}$ should be at least one decade below the desired crossover frequency. Further, $\omega_{fh}$ should be about 2 10 times the crossover frequency. If $\omega_{fh}$ is too high, the estimator may attempt to fit the model at too high frequencies.

**Example 11.1—Choice of disturbance annihilation filter**

When $d(t)$ is a piecewise constant signal, then $D_d(q) = q - 1$. If $D_d' = 1$, then $H_f$ corresponds to taking the differences of the inputs and outputs of the process. To eliminate the influence of the high frequencies it is advisable to use

$$D_d'(q) = \frac{q - \alpha}{1 - \alpha}$$

where $|\alpha| < 1$. The parameter $\alpha$ should be chosen such that it corresponds to 2 10 times the closed-loop crossover frequency. The filter $H_f$ then
becomes
\[ H_f(q) = \frac{(1 - \alpha)(q - 1)}{q - \alpha} \]

With the filter \( H_f \) the disturbance \( d(t) \) will be removed from the data, except for components in the frequency interval \((\omega_{fl}, \omega_{fh})\). The estimator will thus not be confused by low frequency drift or disturbances at specific frequencies. These disturbances are instead taken into consideration when the controller is designed by using the internal model principle. This implies that a level is removed by having an integrator in the controller, a ramp by having two integrators, and so on. Notice that the regulator is still using the unfiltered sampled signal.

One way to eliminate offset or mean values in the signals is to use the CARIMA model of Eq. (5.59)
\[ A^*(q^{-1}) \Delta y(t) = B^*(q^{-1}) \Delta u(t - d_o) + C^*(q^{-1}) e(t) \quad (11.3) \]
where
\[ \Delta = 1 - q \]
This is the same as the filtering of the signals discussed above, but the model will accentuate the high-frequency properties of the process. The differentiation of the signals should thus also be combined with a filter with high-frequency roll-off as in Eq. (11.2).

Another way to take care of offset is to model the process as
\[ A^*(q^{-1}) y(t) = B^*(q^{-1}) u(t - d_o) + C^*(q^{-1}) e(t) + d \quad (11.4) \]
where \( d \) represents the offset. The offset can now be treated as a parameter that can be estimated. The parameter vector is then augmented by \( d \) and the regression vector by a constant equal to unity.

Notice that the removal of bias in the signals used for the estimation should not be mixed up with the reset action in the controller. This will be discussed in Sections 11.3 and 11.4.

**Unmodeled Dynamics**

Real physical processes may have complicated dynamics. They may be nonlinear or infinite dimensional. A key feature of adaptive controllers is that they attempt to model dynamics by simple linear models. The parameters obtained when a simple model is fitted depend strongly on the properties of the input signal. A linear model can describe a nonlinear system well if the input signal is a square wave with fixed amplitude but large discrepancies may appear for other signal amplitudes. The frequency content of the input signal is also important as is illustrated in Example 11.2.
Example 11.2—Fitting low order models to high order systems
Consider a process with transfer function \( G(s) \). Assume that it is attempted to model the system by a first order system with transfer function

\[
G_m(s) = \frac{b}{s + a}
\]

If the input signal is sinusoidal with frequency \( \omega_o \) it is possible to get a perfect fit if \( \text{Im}\{G(i\omega_o)\} \neq 0 \). Straightforward calculations show that \( G_m(i\omega_o) = G(i\omega_o) \) if the parameters are chosen as

\[
a = \frac{\omega b \text{Re}\{G(i\omega_o)\}}{\text{Im}\{G(i\omega_o)\}}
\]

\[
b = \frac{\omega |G(i\omega_o)|^2}{\text{Im}\{G(i\omega_o)\}}
\]

This shows that it is possible to get a perfect fit but that the parameters obtained depend on \( \omega_o \). \( \square \)

An interesting property of adaptive systems is that the parameters are estimated in closed loop. This implies that the simple model in the adaptive controller is fitted with the actual signals generated by the feedback. It explains intuitively the self-tuning property. However, there may be severe problems with unmodeled dynamics, as was illustrated in Section 6.6., where the unmodeled dynamics drives the estimates to unreasonable values. Filtering the signals before they are introduced in the estimator is one possibility to alleviate the problem of unmodeled dynamics. The averaging theory introduced in Chapter 6 is a good tool to investigate the effects of filtering. This is illustrated in Example 11.3.

Example 11.3 Estimator filtering and unmodeled dynamics
Consider adaptation of a feedforward gain using the SPR rule. Let the process have the transfer function \( G(s) \) and assume that the adaptive controller is designed assuming that the transfer function is \( G_m(s) \). The adaptive system is then described by

\[
\frac{d\theta}{dt} + \gamma u_c \left( G(p)\theta u_c - \theta^0 G_m(p)u_c \right) = 0
\]

where \( u_c \) is the command signal. See Eq. (6.30). Using averaging analysis the stability condition is

\[
\gamma \text{avg}\{u_c(Gu_c)\} > 0
\]

If the command signal is

\[
u_c(t) = 1 + \sqrt{2}a \sin \omega t
\]
the stability condition becomes

$$G(0) + a^2 \text{Re} G(i\omega) > 0$$

This condition can be violated if $a$ is sufficiently large for frequencies where $\text{Re} G(i\omega) < 0$. If the command signal is filtered before introducing it to the parameter estimator the stability condition becomes

$$G(0)G_f^2(0) + a^2|G_f(i\omega)|^2 \text{Re} G(i\omega) > 0$$

where $G_f$ is the transfer function of the filter. Stability can clearly be established by choosing a proper low-pass filter. □

Filtering is one way to make the estimator less sensitive to unmodeled dynamics. The normalization method discussed in Section 6.6 is another method. When normalization is used the signals are also preprocessed before they are fed into the estimator. Normalization changes the amplitude of the signals that influences the adaption rate.

**Signal Conditioning**

To obtain robust control it is necessary that the model be accurate around the crossover frequency. To estimate an accurate reduced-order model with this property it is essential that the input signal have sufficient energy content around the crossover frequency and that it be so rich in frequency that it is persistently exciting. A relay experiment like the one done in the auto-tuner is a convenient way automatically to generate a signal whose energy is concentrated at the frequency at which the phase lag is $180^\circ$.

The necessary degree of persistent excitation is related to the complexity of the estimated model. This implies that the requirements on the input signal become more severe when the model order is increased. Since the input signal is generated by feedback, there is no guarantee that it will be persistently exciting. To guarantee a good model, it is thus necessary to monitor the excitation and the energy of the input signal in the relevant frequency bands.

When the input signal generated by the feedback loop is not persistently exciting or when the signal energy is too low, the estimated parameters will be poor. There are two ways to avoid this: either by injecting extra perturbation signals or by switching off the adaptation when the excitation is poor. Some results indicate that it is reasonable to estimate only when the absolute value of the useful input energy is above a certain level. Filtering or external perturbations are methods that can be used to shape the frequency content of the signal. One way to monitor the excitation is shown in Fig. 11.2. There are other safeguards of a similar nature to make sure that the estimation is done only when the data is reasonable.
The robustness problems in adaptive control discussed in Chapter 6 were due to high-frequency reference signals and measurement noise in combination with unmodeled dynamics. These difficulties can be eliminated or at least partly removed if the precautions above are taken.

**Example 11.4 Extra perturbation signals**

Consider the process and controller in Example 6.5. The process is described by a third-order system (Eq. 6.33), while the controller adjusts only two parameters. The problem with the controller shows up when the reference signal is not persistently exciting in the appropriate frequency range. If the only excitation is a high-frequency sinusoid, the adaptive system will attempt to match the model to the system at this frequency. This will result in a low-order model that will represent the dominating dynamics very poorly, and disastrous results can be expected. Figure 11.3(a) shows the parameter estimates obtained with the reference signal \( u(t) = \sin 16.09t \). The remedy is either to switch off the adaptation when there is no excitation in the appropriate frequency band or to inject a relevant perturbation signal. One way to improve the algorithm is thus to filter the output of the process and to introduce a low-frequency excitation signal.

Signal energy at low frequencies will make it possible to estimate better the dominating dynamics of the process. Figure 11.3(b) shows the parameter estimates when the output is filtered with a low-pass filter

\[
H_f(s) = \frac{\omega_f^2}{s^2 + 2\xi \omega_f s + \omega_f^2}
\]

with \( \omega_f = 5 \) and \( \xi = 0.707 \). The excitation signal is \( u(t) = \sin 2t + \sin 16.09t \). With the additional energy content in the lower-frequency range, the estimates will stay bounded and the closed-loop system will be stable. This behavior can also be verified by averaging-loop analysis of the type described in Section 6.5."
Parameter Tracking and Estimator Windup

The key property of an adaptive controller is its ability to track variations in process dynamics. To do so it is necessary to discount old data, which involves compromises. Excessively fast discounting will make the estimates uncertain if the parameters are constant; excessively slow discounting will make it impossible to track rapid parameter variations.

Exponential forgetting is one way to discard old data. The basic recursive equations are given by

\[ \dot{\theta}(t) = \hat{\theta}(t - 1) + K(t)(y(t) - \varphi^T(t)\hat{\theta}(t - 1)) \]  

(11.5)

\[ K(t) = P(t - 1)\varphi(t)(\lambda + \varphi^T(t)P(t - 1)\varphi(t))^{-1} \]  

(11.6)

\[ P(t) = (I - K(t)\varphi^T(t))P(t - 1) / \lambda \]  

(11.7)

With \( \lambda = 1 \), all data have the same weight, but with \( \lambda < 1 \), recent data are weighted more than old data. It is possible to generalize the method with exponential forgetting and have different forgetting factors for different parameters. Exponential forgetting works well only if the process is properly excited all the time. There are several problems with exponential forgetting when the excitation of the process changes. A typical
situation is when excitation results mainly from changes in the setpoint. There may then be long periods with no excitation, the estimator will forget the proper values of the parameters, and the uncertainties will grow. This may be called estimator windup (compare with integrator windup in conventional integral control). The problem can be understood from Eq. (11.7). The negative term in the right-hand side represents the reduction in uncertainty due to the last measurement. If there is no information in the last measurement, \( P(t)\phi(t) \) will not change direction and Eq. (11.7) reduces to

\[
P(t) = P(t - 1)/\lambda
\]

If \( \lambda < 1 \), then \( P(t) \) will grow exponentially until \( \phi \) changes direction. If there is no excitation for a long period of time, \( P(t) \) may be very large. Since \( P(t) \) also influences the gain in Eq. (11.6), there may be large changes in the estimated parameters when new information is coming into the system (for instance, when the reference value changes). The estimator windup may then cause a burst in the output of the process.

There are several ways to avoid estimator windup. The main idea is to ensure that \( P \) stays bounded. This can be done, for instance, by ensuring that the trace of \( P \) is constant in each iteration. Another possibility is to adjust the forgetting factor automatically, but this does not guarantee that \( P \) stays bounded. It has also been proposed to stop the updating of the parameters and the covariance matrix when \( P\phi \) or \( \varepsilon \) is sufficiently small (i.e., a dead-zone).

One version of the regularized constant-trace algorithm that has shown good properties in experiments is

\[
\theta(t) = \theta(t - 1) + a(t)K(t)(y(t) - \phi^T(t)\hat{\theta}(t - 1))
\]  

\[
K(t) = P(t - 1)\phi(t)(1 + \phi(t)^TP(t - 1)\phi(t) + \phi(t)^T\phi(t))^{-1}
\]  

\[
P(t) = P(t - 1) - a(t)\frac{P(t - 1)\phi(t)\phi^T(t)P(t - 1)}{1 + \phi(t)^TP(t - 1)\phi(t) + \phi(t)^T\phi(t)}
\]  

\[
P(t) = c_1\frac{\tilde{P}(t)}{tr(\tilde{P}(t))} + c_2I
\]  

\[
a(t) = \begin{cases} 
\bar{a} & |y(t) - \phi^T(t)\theta(t - 1)| > 2\delta \\
0 & \text{otherwise}
\end{cases}
\]  

where \( c_1 > 0, c_2 \geq 0, c \geq 0 \), and \( \delta \) is an estimate of the magnitude of the noise. The variable \( a(t) \) introduces a dead zone in the estimator. This can easily be introduced in Eqs. (11.5) (11.7) also.
Typical values for the parameters can be
\[
    a \in [0.1, 0.5] \\
    c_1/c_2 \approx 10^4 \\
    \varphi^T\varphi \cdot c_1 \gg 1 \\
    \varphi^T\varphi c \approx 1
\]
for typical values of $\varphi^T\varphi$. If $c_2$ is used in Eq. (11.11), it is not necessary to use $c$.

Another approach to the problem of estimator windup is to forget information only in the directions in which new information is gathered. The following estimator is then obtained:
\[
    \hat{\theta}(t) = \hat{\theta}(t - 1) + K(t)(y(t) - \varphi(t)^T\hat{\theta}(t - 1)) \\
    K(t) = \frac{P(t - 1)\varphi(t)}{v(t) + \varphi(t)^TP(t - 1)(1 - \alpha(t)v(t))} \\
    P(t) = P(t - 1) - \frac{P(t - 1)\varphi(t)\varphi(t)^TP(t - 1)}{(v(t)^{-1} - \alpha(t))^{-1} + \varphi(t)^TP(t - 1)\varphi(t)}
\]
\[
    \alpha(t) = \begin{cases} 
    0 & \alpha_d \leq 0 \\
    \alpha_d & 0 < \alpha_d \leq 1/(\varphi^TP\varphi) \\
    1/(\varphi^TP\varphi) & 1/(\varphi^TP\varphi) < \alpha_d \leq \frac{1}{v^{-1}} + 1/(\varphi^TP\varphi) \\
    0 & \alpha_d > \frac{1}{v^{-1}} + 1/(\varphi^TP\varphi)
\end{cases}
\]
\[
    \delta_d(t) = \frac{\varphi(t)^TP(t - 1)P(t - 1)\varphi(t)}{\varphi(t)^TP(t - 1)^2\varphi(t)} - a \\
    \alpha_d(t) = v(t)^{-1} + \frac{\delta_d(t)}{\delta_d(t)\varphi(t)^TP(t - 1)\varphi(t) - 1}
\]

The estimates will then converge to values such that $P(t) = a \cdot I$, where $a$ should be chosen as a small value. Typically, $a \in (0.0001, 0.001)$. Finally $v(t)$ should be an estimate of the variance of the residual $\varepsilon = y(t) - \varphi^T(t)\hat{\theta}(t - 1)$.

The covariance matrix $P$ in Eqs. (11.7), (11.11), or (11.15) should be a symmetric positive definite matrix. A straightforward implementation of the equations may lead to numerical problems in the same way as when implementing Kalman filters. The algorithm will become more numerically robust if U-D factorization or square root algorithms are used for the updating of the $P$ matrix.
The dead zone prevents the estimator from drifting when the errors are small and when the system is not persistently exciting. In Section 6.6 projections and leakage were discussed; these are precautions that can be taken to create an "attraction zone" for the parameters when they are far away from expected values. The leakage improves the robustness properties at the cost of an offset from the desired convergence point for the estimator. Dead zones, projections, and leakage will thus introduce application-dependent parameters. The theoretical results for the leakage-type modification are so far limited to bounded disturbances. In any case, the modifications are good from a practical point of view, since the robustness properties are increased.

**Example 11.5—Estimator windup**

The problem of estimator windup is illustrated by a simple example. Let the process to be controlled be described by

\[ y(t) - 0.9y(t - 1) = u(t - 1) \]

We want the pulse transfer operator from the reference signal to the output to have a pole in 0.7 and the gain to be unity. This is achieved with the controller

\[ u(t) = 0.3y(t) - 0.35y(t - 1) + u_c(t) - 0.5u_c(t - 1) \]

The process is controlled using a direct pole placement algorithm in which the parameters in the controller are estimated. Figure 11.4 shows the diagonal elements of the \( P \) matrix (Eq. 11.7) when different estimation schemes are used. The reference signal is a square wave with unit magnitude and period 100 up to time 300. After that the reference signal is constant. In Fig. 11.4(a) the estimation algorithm described by Eqs. (11.5) (11.7) is used, with \( \lambda = 0.99 \). When the reference signal is constant and the output has settled, there is no information in the measurements. The variance will then start to increase. Figure 11.4(b) shows the variances when Eqs. (11.13) (11.18) are used. The variances of the estimates now settle on constant values, and there is no estimator windup.

If the bias is estimated using the model of Eq. 11.4, it may be necessary to have different forgetting factors for the parameters in the \( A, B, \) and \( C \) polynomials and for the bias \( d \). This is the case, for instance, if the bias is influenced by load disturbances. The bias \( d \) must quickly be adjusted to a new value when the load changes, while the remaining process parameters usually do not change that rapidly. Finally, if the bias is estimated in this way, it is not possible to use the adaptive controller as a tuner since there will be no reset when the estimation is switched off.

Another way to avoid estimator windup, called *leakage*, was discussed in Section 6.6. In continuous time the estimator was modified as shown in
Figure 11.4 The diagonal elements of the $P$ matrix when controlling the process in Example 11.5. (a) Constant exponential forgetting factor $\lambda = 0.99$. (b) Estimation using directional forgetting as in Eqs. (11.13) (11.18).

Eq. (6.45) by adding the term $\alpha (\theta^0 - \theta)$. This means that the parameters will converge to $\theta^0$ when no useful information is obtained, i.e., when $e = 0$. A similar modification can also be made in discrete-time estimators. It is useful to observe that estimators with leakage are obtained by minimizing the loss function

$$V(\theta, t) = \frac{1}{2} \sum_{i=1}^{t} (y(i) - \varphi^T(i)\theta)^2 + \frac{1}{2} (\theta - \theta^0)^T R^{-1}(\theta - \theta^0) \tag{11.19}$$

Compare with the loss function used in ordinary least squares (Eq. 3.2). This loss function leads to the following recursive equations for updating the estimates

$$\theta(t) = (P^{-1}(t) + R^{-1}) \left( P^{-1}(t)(\hat{\theta}(t - 1) + K\varepsilon(t)) + R^{-1}\theta^0 \right) \tag{11.20}$$

where

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

Compare with Eq. (11.5). The estimate given by Eq. (11.20) also has the property that it converges to $\theta^0$ when there is no data, i.e., $\varepsilon(t) = 0.$
Notice that the interpretation that the estimate minimizes Eq. (11.19) is useful because it gives a guideline for choosing the symmetric matrix $R$. If the measurement noise has variance $\sigma^2$ Eq. (11.19) can be interpreted as the logarithm of the likelihood function when the estimates have a prior distribution that is Gaussian with mean $\theta^0$ and covariance $\sigma^2 R$.

### 11.3 Controller Implementation

Design and implementation of digital controllers are treated in books on digital control. Some aspects on this are discussed in this section under the motto:

> The design is automatic, so everything must be covered in the algorithm.”

#### Presampling or Anti-aliasing Filters

In all digital control applications it is important to have a proper filtering of the signals before they are sampled. Because of the aliasing problem connected with the sampling procedure it is necessary to eliminate all frequencies above the Nyquist frequency before sampling the signals. High frequencies may otherwise be interpreted as low frequencies and introduce disturbances in the controller. This implies that the signal conditioning is related to the choice of the sampling interval. Suitable choices of anti-aliasing filters are second- or fourth-order Butterworth, ITAE (Integral Time Absolute Error), or Bessel filters. They consist of one or several cascaded filters of the form

$$G_f(s) = \frac{\omega^2}{s^2 + 2\zeta \omega s + \omega^2}$$

Let $\omega_B$ be the desired bandwidth of the filter. The damping $\zeta$ and the frequency $\omega$ can be chosen according to Table 11.1. The anti-aliasing filter will then naturally low-pass filter the outputs of the process. The filter will also make it possible for the estimator to get good models in the correct frequency range. (Compare the discussion of $H_f$ in Section 11.2.) This feature of discrete-time systems may be one reason why the type of robustness difficulties discussed in Example 11.4 were first noticed in continuous-time adaptive controllers. The natural filtering obtained through the sampling helps the estimator to get better models.

How will the anti-aliasing filter influence the sampled data model? The Bessel filter is a good approximation of a time delay for frequencies up to around $\omega_B$. For instance, the fourth-order filter can be well approximated
Table 11.1 Damping and natural frequency of second-, fourth-, and sixth-order Butterworth ITAE, and Bessel filters. The filters have the bandwidth $\omega_B$.

<table>
<thead>
<tr>
<th>Order</th>
<th>Butterworth $\omega/\omega_B$</th>
<th>ITAE $\omega/\omega_B$</th>
<th>Bessel $\omega/\omega_B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1 0.71</td>
<td>1 0.71</td>
<td>1.27 0.87</td>
</tr>
<tr>
<td>4</td>
<td>1 0.38 1.48</td>
<td>0.32 1.59</td>
<td>0.62 0.96</td>
</tr>
<tr>
<td></td>
<td>1 0.92 0.83</td>
<td>0.83 1.42</td>
<td>0.96 0.49</td>
</tr>
<tr>
<td>6</td>
<td>1 0.26 1.30</td>
<td>0.32 5.14</td>
<td>0.49 0.82</td>
</tr>
<tr>
<td></td>
<td>1 0.71 0.98</td>
<td>0.60 4.57</td>
<td>0.82 0.98</td>
</tr>
<tr>
<td></td>
<td>1 0.97 0.79</td>
<td>0.93 4.34</td>
<td>0.98</td>
</tr>
</tbody>
</table>

by a time delay of

$$T_d = \frac{2\pi}{3\omega_B}$$

This implies that the sampled data model including the anti-aliasing filter can be assumed to contain an additional time delay compared to the process. Assume that the bandwidth of the filter is chosen as

$$|G_{aa}(i\omega_N)| = \beta$$

where $\omega_N = \pi/h$ is the Nyquist frequency and $G_{aa}(s)$ is the transfer function of the filter. The attenuation $\beta$ may be in the range 0.1 0.5. Table 11.2 gives some values of $T_d$ as a function of $\beta$. The table shows that, for a fixed sampling period or Nyquist frequency, the delay will increase if a higher attenuation is desired. This also means that the anti-aliasing filter must be taken into account when estimating the process model and in the design. For reasonable values of $\beta$, the filter can be approximated with a delay of one or two sampling periods. The price for having the anti-aliasing filter is that one or two extra parameters have to be estimated in the model. However, if the desired closed-loop bandwidth is less than 0.05 0.1 times the bandwidth of the anti-aliasing filter, then the filter will have very little influence on the process model around the desired crossover frequency.

In many process industry applications it can be sufficient to have long sampling periods. A corresponding analog anti-aliasing filter will then have quite large component values. This problem can be avoided by making a faster sampling, with an appropriate anti-aliasing filter, and then filtering the sampled signals digitally to remove frequencies above the Nyquist frequency for the control signal.
Table 11.2 The approximate time delay $T_d$ due to the anti-aliasing filter as a function of the desired attenuation $\beta$ at the Nyquist frequency for a fourth-order Bessel filter. $h$ is the sampling period.

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>$\omega_N/\omega_B$</th>
<th>$T_d/h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>3.1</td>
<td>2.1</td>
</tr>
<tr>
<td>0.1</td>
<td>2.5</td>
<td>1.7</td>
</tr>
<tr>
<td>0.2</td>
<td>2.0</td>
<td>1.3</td>
</tr>
<tr>
<td>0.5</td>
<td>1.4</td>
<td>0.9</td>
</tr>
<tr>
<td>0.7</td>
<td>1.0</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Example 11.6 The effect of the anti-aliasing filter

Let the process to be controlled be given by

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

which can be a model for a motor. The system is sampled with $h = 0.5$, and the desired closed-loop system is assumed to be described by a second-order system with $\omega_m = 1$ rad/s and $\zeta_m = 0.7$. A pole placement controller is determined. The output of the system is disturbed by a sinusoidal signal, i.e., the measured signal is

$$y_m(t) = y(t) + a_d \sin(\omega_d t)$$

This signal is filtered through a fourth-order Bessel filter with bandwidth $\omega_B$. Figure 11.5 shows the influence of the disturbance and the anti-aliasing filter. Figure 11.5(a) and (b) show the influence of the disturbance when $\omega_B = 25$ rad/s, which is far too high for the disturbance frequency. The Nyquist frequency is 6.28 rad/s. The disturbance frequency, which is $\omega_d = 11.3$ rad/s, is folded and gives rise to a low-frequency disturbance in the sampled measurement. The controller then tries to compensate this "imaginary" frequency. Figure 11.5(c) shows the effect of the anti-aliasing filter when $\omega_B - \omega_N$. The disturbance is smaller but still noticeable in the input and the output. The filter will, however, also influence the closed-loop performance. Figure 11.5(d) shows the same experiment as in Fig. 11.5(c), but when the controller is modified. The controller is now determined for the true process when the anti-aliasing filter is approximated by a delay of $0.7h$ (compare Table 11.2). The closed-loop response is improved compared to Fig. 11.5(c). Finally, Fig. 11.5(e) and (f) show the case when $\omega_B = \omega_N/2.5$ and when the filter is approximated by a delay of $1.7h$ in the design of the controller. 

\(\Box\)
Figure 11.5  Output, reference value, and control signal for the system in Example 11.4. The measurement disturbance has the frequency $\omega_d = 11.3$ rad s. (a) $a_d = 0, \omega_B = 25$ rad/s; (b) $a_d = 0.1, \omega_B = 25$ rad/s; (c) $a_d = 0.1, \omega_B = 6.28$ rad s; (d) $a_d = 0.1, \omega_B = 6.28$ rad s, and the regulator compensated for a delay of 0.7h; (e) $a_d = 0, \omega_B = 2.51$ rad/s, and the regulator compensated for a delay of 1.7h; (f) same as (e) but with $a_d = 0.1$.

The example shows that it is important to use an anti-aliasing filter and that the filter has to be considered in the design. It is, however, sufficient to approximate the filter by a time delay. This simplifies estimation design in the adaptive case.

**Postsampling Filters**

The output of a D-A converter is a piecewise constant signal. This means that the control signal fed to the actuator is composed of a series of steps where the smallest step change is given by the resolution of the converter. This is adequate for many processes. For some systems, like hydraulic servos for flight control and other systems with poorly damped oscillatory modes, the steps may, however, excite these modes. In such a case it is advantageous to use a filter that smooths the signal from the D-A converter. Such a filter is called a *postsampling filter*. The postsampling filter may be a simple continuous-time filter with a response time that is short compared with the sampling time. Special D-A converters that give a smooth signal have also been constructed.
Figure 11.6 Two ways to synchronize inputs and outputs. (a) The signals measured at time $t_k$ are used to compute the control signal applied at $t_{k+1}$. (b) The control signal is applied as soon as it is computed.

Computational Delay

Since the analog-to-digital (A-D) and digital-to-analog (D-A) conversions and the computations take time, there will always be a delay between the measurement and the time the control signal is applied to the process. This delay, which is called the computational delay, depends on how the control law is implemented in the computer. Two ways are illustrated in Fig. 11.6. In Case (a) the measured variable at time $t_k$ is used to compute the control signal applied at time $t_{k+1}$. In Case (b) the control signal is applied as soon as the computations are finished. The disadvantage of Case (a) is that the control action is delayed unnecessarily long. In Case (b) the disadvantage is that the time delay may change depending on the load on the computer or changes in the program. In both cases it can be necessary to include the computational delay in the design of the controller.

In Case (b) it is desirable to make the delay as small as possible. This can be done by making as few operations as possible between the A-D and D-A conversions. Assume that the regulator has the form

$$R^*(q^{-1})u(t) = T^*(q^{-1})u_c(t) - S^*(q^{-1})y(t)$$

(11.21)
One way to make the implementation is as follows:
1. Make A-D conversion of $y(t)$ and $u_c(t)$
2. Compute $u(t) = t_0 u_c(t) - s_0 y(t) + u_1(t)$
3. Make D-A conversion of $u(t)$
4. Compute

$$u_1(t + 1) = (1 - R^*) u(t + 1) + (T^* - t_0) u_c(t + 1) - (S^* - s_0) y(t + 1)$$

Notice that $u_1(t + 1)$ in Step 4 can be computed with the data available at time $t$. This makes it possible to calculate $u_1(t + 1)$ in advance, once the control signal has been applied to the process.

**Anti-reset Windup**

An integrator is an unstable system; sometimes the integrator can assume very large values if the control signal saturates when there is a control error. This is called reset windup or integrator windup. Special precautions must be taken in order to avoid this.

**Example 11.7 Anti-reset windup controller**

Consider a regulator described by Eq. (11.21). The regulator may contain unstable modes. One way to solve the reset windup problem is to rewrite Eq. (11.21) by adding $A_o^* (q^{-1}) y(t)$ on both sides. This gives

$$A_o^* u(t) = T^* u_c(t) - S^* y(t) + (A_o^* - R^*) u(t)$$

A regulator with anti-reset windup compensation is then given by

$$A_o^* v(t) = T^* u_c(t) - S^* y(t) + (A_o^* - R^*) u(t)$$
$$u(t) = \text{sat}(v(t)) \quad (11.22)$$

where sat(·) is the saturation function. This regulator is equivalent to Eq. (11.21) when the control signal does not saturate. $A_o^*$ is chosen as a stable polynomial and can be interpreted as the observer dynamics of the controller. A block diagram of Eq. (11.22) is shown in Fig. 11.7. A particularly simple case is when $A_o^* = 1$ which corresponds to a deadbeat observer. The controller is then

$$u(t) = \text{sat}(T^* u_c(t) - S^* y(t) + (1 - R^*) u(t)) \quad \square$$

**Choice of Sampling Interval**

The choice of sampling interval depends on the design algorithm as well as on the desired closed-loop performance. The sampling rate influences
many properties of a system like following of command signals, rejection of load disturbances and measurement noise, and sensitivity of unmodeled dynamics. In some cases the sampling interval also determines the computer power required. Selection of sampling rates is thus an essential design issue.

One rule of thumb that is useful for deterministic design methods is to let the sampling interval $h$ be chosen such that

$$\omega_o h \approx 0.1 - 0.5$$

where $\omega_o$ is the natural frequency of the dominating poles of the closed-loop system. If the dominating pole is real, one can use

$$h/T_o \approx 0.1 - 0.5$$

where $T_o$ is the time constant of the dominating pole. These rules imply that there will be about 5 to 20 samples in a step response of the closed-loop system. The choice of the sampling interval is also influenced by the disturbances. If the process is disturbed by occasional large disturbances, it can be necessary to shorten the sampling interval to ensure that the disturbance is not acting for too long a time on the process before it is detected and compensated for.

Stochastic design methods require other rules for the choice of the sampling interval. If the process zeros are canceled as, for instance, in minimum-variance control then it is necessary to choose a sampling interval such that the sampled data process model is minimum-phase. For linear quadratic design methods it is usually possible to use quite short sampling periods.

For stochastic design methods it is important how the performance of the closed-loop system is measured. One way is to consider the output and
input signal variances at the sampling instances. A more relevant measure is to consider the average variance of the underlying continuous-time process output. The reason is that the variance in “steady state” is periodic over the sampling interval. The periodicity is a fundamental property of sampled-data systems because of the periodicity of the changes in the control signal. The variance can be large at times between the sampling points. This periodical behavior can be considered as the stochastic correspondence to “hidden oscillations” in deterministic sampled-data control. To avoid large intersample variation of the output variance, it is advisable not to cancel process zeros in the left-hand side of the unit circle. Also, it can be better to choose a design method that has a continuous-time counterpart such as linear quadratic control.

The choice of the sampling interval also influences the determination whether the anti-aliasing filter should be taken into account in the design. First assume that the sampling frequency is high (20 100 times) compared to the desired closed-loop bandwidth. It is then not necessary to take the filter into account when making the design. The amplitude of the filter is about unity, and the phase lag is negligible at the crossover frequency. However, the influence of the filter cannot be neglected in the design if the Nyquist frequency is only a little bit larger than the desired crossover frequency. In general, this means that the estimated model must be of higher order than if a higher sampling rate were chosen. (Compare Example 11.4.) This is one reason for using a short sampling period.

The Automated Design Procedure

Since the design procedure is automatic, it is necessary to cover all possibilities and pitfalls in the logic of the design. For instance, it can be necessary to test whether the estimated process model is minimum-phase or whether there are common factors in the estimated polynomials. Direct adaptive controllers have the advantage that the design step is eliminated, since the parameters of the regulator are estimated directly. This can be made possible by a reparameterization of the process model. A typical example of a direct algorithm is the moving-average self-tuner discussed in Chapter 5. Direct methods are, however, limited to special design procedures.

The direct algorithms usually have more parameters to estimate than the indirect algorithms, particularly if there are long time delays in the process. Simulations and practical experiments indicate, however, that direct algorithms are more robust in many situations. Direct algorithms usually have the disadvantage that all process zeros are canceled. This implies that direct methods can only be used for processes with stable inverse or minimum-phase systems. Sampling of a continuous-time system often gives a discrete-time system with zeros on the negative real axis.
inside or outside the unit circle. (Compare Theorem 6.4.) It is not good practice to cancel these zeros, even if they are inside the unit circle, because this will give rise to "ringing" in the control signal. Direct methods with cancelation of process zeros cannot be used when the process is non-minimum-phase. It is thus advisable not to cancel any process zeros when using direct methods. There is no way to test whether the process is minimum-phase or not if only the controller polynomials are known. Indirect methods have the advantage that the $B$ polynomial in the process model is estimated. The stability of this polynomial can then easily be tested, and the design procedure can be changed if the estimated model is non-minimum-phase.

**Solution of the Diophantine Equation**

Several of the design methods discussed earlier involve the solution of a Diophantine equation of the form

$$AR + BS = A_m A_o$$

As discussed in Appendix A, this is the same as solving a set of linear equations. A condition for the existence of a solution is that there are no common factors in the $A$ and $B$ polynomials. In the automatic design in an adaptive controller it is thus necessary to check for and remove common factors. This can be done using Euclid's algorithm for $A$ and $B$. One algorithm is given in detail in Appendix A.

**11.4 Estimation and Control Interaction**

Estimation, design, and controller implementation have been discussed as separate subjects in the two previous sections but in an adaptive controller there is a strong interaction between these parts. For instance, the desired closed-loop performance will influence the frequency content of the input and output signals of the process. This will in turn influence the estimation of the parameters. One way to decrease the interaction is to use a low gain in the adaptation loop, but this makes it more difficult to follow fast variations in the parameters of the process.

**Computational Delay**

The updating of the estimated parameters and the design are done at each sampling instant. The timing of computations of the controller was discussed in Section 11.3. It was pointed out that it is important to have as short a computational delay as possible. The dual time scale of the adaptive control problem implies that the process parameters are assumed
to vary slowly. This means that the parameter estimates from the previous sampling instant can be used for calculating the control signal. There will thus be no extra time delay due to the adaptation, provided the parameter update and the controller design are done after the control signal is sent out to the process.

**Reset Action in Adaptive Regulators**

The introduction of reset action in general sampled data controllers was discussed in Section 11.3. An adaptive controller can also automatically introduce reset action. Since it estimates a model of the process and the environment, it can be expected that the self-tuner tries to model the offset and compensate for it. It is easy to check whether a particular self-tuner has this ability by investigating possible stationary solutions. A typical example is given below.

**Example 11.8—Automatic reset action**

Consider the simple direct moving-average self-tuning controller described in Chapter 5, which is based on least-squares estimation and minimum-variance control. The estimation is based on the model

\[ y(t + d) = R^*(q^{-1})u(t) + S^*(q^{-1})y(t) \]

and the regulator is

\[ u(t) = -\frac{S^*}{R^*} y(t) \]

The conditions for a stationary solution are that

\[ r_y(\tau) = \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} y(k + \tau)y(k) = 0 \quad \tau = d, \ldots, d + l \]

\[ r_{yu}(\tau) = \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} y(k + \tau)u(k) = 0 \quad \tau = d, \ldots, d + k \]

where \( k \) and \( l \) are the degrees of the \( R^* \) and \( S^* \) polynomials, respectively. These conditions are not satisfied unless the mean value of \( y \) is zero. When there is an offset, the parameter estimates will get values such that \( R^*(1) = 0 \), i.e., there is an integrator in the controller. However, the convergence to the integrator may be slow. It can be shown that other (both indirect and direct) self-tuning regulators also can give reset automatically. \( \square \)

A second way to introduce reset action is to estimate the bias in the process. A simple way to do this is to use the model of Eq. (11.4). With
this model it is easy to estimate $d$ and to compensate for it. One drawback is that an extra parameter has to be estimated. Further, it is necessary to have different forgetting factors on the bias estimate and the other estimates; otherwise the convergence to a new level will be very slow. Finally, if the bias is estimated in this way, it is not possible to use the self-tuner as a tuner, since there will be no reset when the estimation is switched off.

A third way to get reset action is to force an integrator into the regulator. That means that the controller has the form

$$R^* \Delta u(t) = -S^* y(t) + T^* u_c(t)$$

where

$$\Delta u(t) = (1 - q^{-1}) u(t) = u(t) - u(t - 1)$$

This form has several advantages. First, there will always be an integrator, even if the regulator parameters are not optimally tuned. Second, the high gain at low frequencies will increase the robustness of the system with respect to uncertainties in the process dynamics at low frequencies. This implies that the estimation can be concentrated at frequencies around the crossover frequency. One drawback of this method is that the self-tuner will try to eliminate the integral action when it is not needed and the regulator will then try to cancel a pole at the stability boundary.

**Start-up Procedures**

There are several ways to initialize a self-tuning algorithm, depending on the available *a priori* information about the process. In one case nothing is known about the process. The initial values of the parameters in the estimator can then be chosen to be zero or such that the initial controller is a proportional or integral controller with low gain. Auto-tuning, discussed in Chapter 8, is a convenient way to initialize the algorithm, because it generates a suitable input signal and safe initial values of the parameters. This will also give a good indication of how to choose the sampling interval.

The inputs and outputs of the process should be scaled so that they are of the same magnitude. This will improve the numerical conditions in the estimation and the control parts of the algorithm. The initial value of the covariance matrix can be $100$ times a unit matrix with proper scaling. These values are usually not crucial, since the estimator will get reasonable values in a very short period of time. Our experience is that $10$ to $50$ samples is sufficient to get a very good controller when the system is excited. During the initial phase it can be advantageous to add a perturbation signal to speed up the convergence of the estimator.

The situation is different if the process has been controlled before with a conventional or an adaptive controller. The initial values should then be such that they correspond to the controller used before.
Sometimes it is important to have as small disturbances as possible due to the start-up of the self-tuning algorithm. There are then two precautions that can be taken. First, the estimator can be used for some sampling periods before the self-tuning algorithm is allowed to put out any control actions. During that time a safe simple controller should be used. It is also possible and desirable to limit the control signal. The allowable magnitude can be very small during the first period of time and can then be increased when better parameter estimates are obtained. This kind of soft start-up is used in Asea’s Novatune. The drawback of having small input signals is that the excitation of the process will be poor and it will take longer to get good parameter estimates.

### Influence of the Design Variables

In the previous sections the influences of the different design variables have been isolated by looking at either the estimation part or the controller design. The complex interdependence is illustrated by an example.

#### Example 11.9—Influence of desired closed-loop bandwidth

The process is a fourth-order system

\[
G(s) = \frac{1}{(s + 1)^4}
\]

The process is identified as a second-order sampled data system using least-squares estimation without any prefiltering of the signals. The controller is of the form of Eq. (11.21) and has the same structure as a digital PID controller. The desired closed-loop response is given as the model

\[
G_m(s) = \frac{\omega_m^2}{s^2 + 2\zeta_m \omega_m s + \omega_m^2}
\]

with \( \zeta_m = 0.7 \). The system and the controller were simulated for different values of \( \omega_m \). Figure 11.8 shows the output and the control signal when the parameters have converged. For \( \omega_m \leq 0.3 \) the control is good. The behavior starts to deteriorate when \( \omega_m \) is further increased. The reason is the difficulty for the estimator to find a good second-order model of the process in the desired frequency band. The example shows that overly simple models can restrict the performance of the system due to modeling errors. The estimated model in Fig. 11.8 is actually

\[
\hat{H}(z) = \frac{0.0085z + 0.0637}{z^2 - 1.612z + 0.704}
\]

which has complex poles. The static gain of the model is 0.85, which is about 15% too low.
The consequence of mismodeling can be influenced in several ways. The adaptive regulator used in Fig. 11.8 was based on a design with a deadbeat observer. The system can be improved somewhat by introducing a design with another observer.

The effects of unmodeled dynamics can also be reduced by filtering the signals before they are introduced into the estimator. Fig. 11.9 shows the results obtained when inputs and outputs are filtered using

\[ H_f(z) = \frac{(1-a)^2}{(z-a)^2} \]

when \( a = 0.7 \) and 0.9. The estimated model when \( a = 0.9 \) becomes

\[ H(z) = \frac{0.00602 + 0.0606}{z^2 - 1.566z + 0.637} \]

This model has a DC-gain of 0.999 and the system also behaves much better.

**Summary**

The aim of the signal processing is to guarantee that the true process can be well approximated by the prejudice model (e.g., Eq. (11.3)) within the
desired bandwidth of the closed-loop system. To summarize, the following conclusions can be drawn from the discussion above.

- Analog anti-aliasing filter. Second- or fourth-order filter with bandwidth below or around the Nyquist frequency $\pi/h$, where $h$ is the sampling period.

- Bandpass filter with $H_f$ to get a weighting in the estimator in an appropriate frequency band. This will remove low-frequency disturbances such as levels and ramps. Known sinusoidals can also be removed using notch filters. The lower limit of the passband should be at least one decade below the desired crossover frequency.

- Estimate a low-order model using an algorithm with time variable exponential forgetting, regularized constant trace, or directional forgetting. The estimator should also contain a dead zone. Finally, the estimator may contain a “switch,” which detects whether the system is sufficiently excited. The “switch” can measure the power in different frequency bands and thus control whether the estimator should be active or not.

- The design method for the controller should be robust against unmodeled dynamics. Level and ramp disturbances are eliminated by introducing integrators in the controller. The control signal should be
limited, and the controller should include anti-reset windup.

A more detailed block diagram with added filters is given in Fig. 11.10. All these points contain choices of parameters. The discussion in the previous sections gives guidelines that can be used for the selection of the parameters.

11.5 Prototype Algorithms

Various general implementation aspects of adaptive controllers have been discussed in the previous sections. In this section a number of algorithms will be discussed in more detail. Guidelines for the coding of the algorithms will be given. Finally, a program skeleton will be developed, for implementing:

- Direct self-tuner
- Indirect self-tuner with zero cancelation
- Indirect self-tuner without zero cancelation

Before going into the details, it is important to structure the algorithm and modularize it into smaller parts.

Algorithm Skeleton

All adaptive control algorithms that are considered in this chapter have the following form:
Analog_Digital_conversion
Compute_control_signal
Digital_Analog_conversion
If estimate then
   begin Filter
      Organize_data_for_estimation
      Parameter_update
      If tune then
         begin th_design:=thEstimated
            Design_calculations
         end
   end
Organize_data
Compute_as_much_as_possible_of_control_signal

Row 1 implements the conversion of the measured output signal the reference signal, and possible feedforward signal. All the converted signals are supposed to be filtered through appropriate anti-aliasing filters, as discussed in the previous section. Row 3 sets the control signal to the process. Rows 14 and 2 contain the calculations of the control signal, which are independent of whether the parameters are estimated or not. Notice the division of the calculations of the control signal in order to avoid overly long computation times. All calculations that are possible to do in advance are done in Row 14. Only calculations that contain the last measurements are done in Row 2.

Rows 4–13 contain calculations that are specific for an adaptive algorithm. There are two logical variables estimate and tune which control whether the parameters are going to be estimated and whether the controller is going to be redesigned, respectively. The estimation is done in Rows 5–7, and the design calculations are done in Row 10. Row 13 organizes the data such that the algorithm is always ready to start estimation when the operator wishes.

The various adaptive algorithms discussed in this section differ only in the design calculations. The estimator part can be the same for all algorithms. One important part of the algorithms that will not be discussed here is the operator interface. This is usually a significant part of an adaptive control system, but it is very hardware-dependent, which makes it difficult to discuss in general terms. The calculations in Rows 4–13 will now be discussed in more detail.

Filter
The filter in Row 5 is the filter $H_f$ in Fig. 11.10. It should be a bandpass filter around the desired crossover frequency. Band-pass filters can be obtained from low-pass filters through a frequency transformation. For
instance, the normalized low-pass filters in Table 11.1 can be transformed into band-pass filters through the transformation

\[ s \rightarrow \frac{s^2 + \omega_l \omega_h}{s(\omega_h - \omega_l)} \]  

(11.23)

where \( \omega_l \) and \( \omega_h \) are the lower and upper cutoff frequencies, respectively. The transformed band-pass filter can then be digitized using, for instance, bilinear or Tustin's transformation. A digital low-pass filter \( H(z^{-1}) \) can be transformed directly using

\[ z^{-1} \rightarrow \frac{z^{-2} - (2\alpha k/(k + 1)) z^{-1} + (k - 1)/(k + 1)}{(k - 1)/(k + 1)z^{-2} - (2\alpha k/(k + 1)) z^{-1} + 1} \]

where

\[ \alpha = \cos(\omega_o h) = \frac{\cos((\omega_h + \omega_l)h/2)}{\cos((\omega_h - \omega_l)h/2)} \]

\[ k = \cot \left( \frac{\omega_h - \omega_l}{2} h \right) + \tan \left( \frac{\omega_B h}{2} \right) \]

\( \omega_o = \) center frequency of band-pass filter
\( \omega_B = \) bandwidth of low-pass filter

**Organize data for estimation and Organize data**

It will be assumed that the estimated model is

\[ y(t) = \varphi^T(t) \theta \]

where the components in \( \varphi \) are lagged filtered inputs and outputs. The ordering, the number of lags, and so on depend on the specific model; these details are easily sorted out for the chosen model structure. Rows 6 and 13 of the algorithm contain the bookkeeping of the \( \varphi \) vector (i.e., the usual shift of some parts of the vector and supplement of the latest measurements and outputs). This part of the algorithm should also include the filtering with \( A_o, A_m, \) etc., that is needed for the different types of algorithms discussed in Chapter 5.

**Parameter update**

For simplicity it is assumed that the estimation and the covariance update are done using Eqs. (11.5) (11.7). The calculations can be organized as in
the listing below, where eps is the residual, th_estimated is the parameter vector, P is the covariance matrix, phi is the data vector, and lambda is the forgetting factor.

\[
\begin{align*}
\text{eps} &= y - \phi^{'\text{th\_estimated}} \\
\text{w} &= P^*\phi \\
\text{den} &= \lambda^* + \phi^{'\text{w}} \\
\text{gain} &= \text{w/\text{den}} \\
\text{th\_estimated} &= \text{th\_estimated} + \text{gain*eps} \\
\text{P} &= (P - w*w'/\text{den})/\lambda
\end{align*}
\]

The prime is the transpose, and * is matrix multiplication. This skeleton can easily be transferred to any preferred programming language.

The estimator can be modified in many different ways, as discussed in Section 11.2. For instance, the update can be stopped if the residual is too small as in Eq. (11.12). Further it can be advisable to assure that the P matrix is symmetric. Finally, it is advisable to update the square root of the covariance matrix instead of the matrix itself. Compare the discussion in Chapter 3.

**Design calculations**

When a direct algorithm such as Algorithm 5.4 is used the controller parameters are the same as the estimated parameters, and there are no calculations that have to be done in the design block. In the indirect methods a polynomial equation has to be solved. The solution of the Diophantine equation is discussed in Appendix A. Some care must be taken because of difficulties with possible common factors in the estimated model polynomials.

**Compute control signal**

The computation of the control signal to minimize the computational delay was discussed in Section 11.3, along with the anti-reset windup. Here we will discuss how to introduce an integrator in the controller.

A regulator with integrator has the form

\[
R^*\Delta u(t) = -S^*y(t) + T^*u_c(t)
\]

or

\[
\Delta u(t) = -(R^* - 1)\Delta u(t) - S^*y(t) + T^*u_c(t) \\
u(t) = u(t - 1) + \Delta u(t)
\]

The regulator is implemented as discussed before, using \(\Delta u(t)\) instead of \(u(t)\). The control signal is finally calculated from the previous value and the computed correction \(\Delta u(t)\).
Notice that in the design calculations the Diophantine equation gets the form

\[ AR\Delta + BS = A_o A_m \]

i.e., the integrator is included in the polynomial equation.

**Summary**

The program skeleton in this section can now be supplemented with details to become a complete adaptive control algorithm. These details will depend on what algorithm is chosen to be implemented and on which programming language is chosen.

**A Direct Self-tuner**

There are many details to consider when developing good adaptive algorithms. This will be illustrated by deriving an algorithm for direct adaptive control. Consider a process model given by

\[ A(q)y(t) = B(q)u(t) \]  \hspace{1cm} (11.24)

Let the desired response to command signals be given by

\[ A_m(q)y(t) = A_m(1)u_c(t - d) \]  \hspace{1cm} (11.25)

where \( d = \deg A(q) - \deg B(q) \) and let the observer polynomial be \( A_o(q) \). The design equation is

\[ AR + BS = B^+ A_o A_m \]  \hspace{1cm} (11.26)

where \( B = b_o B^+ \). Requiring that the regulator has integral action we find that the polynomial \( R \) has the form

\[ R = R'_1 B^+(q - 1) = R'_1 B^+ \Delta \]  \hspace{1cm} (11.27)

Equation (11.26) then becomes

\[ A\Delta R'_1 + b_o S = A_o A_m \]  \hspace{1cm} (11.28)

Hence

\[ A_o A_m y = AR'_1 \Delta y + b_o S y = B R'_1 \Delta u + b_o S y = b_o (R' \Delta u + S y) \]  \hspace{1cm} (11.29)

where Eqs. (11.24) and (11.27) were used to obtain the second and third equalities. Rewriting Eq. (11.29) in the backwards operator we get

\[ A_o^*(q^{-1}) A_m^*(q^{-1}) y(t + d) = b_o (R'^*(q^{-1}) \Delta^*(q^{-1}) u(t) + S^*(q^{-1}) y(t)) \]  \hspace{1cm} (11.30)
This equation could be used as a basis for parameter estimation but there are several drawbacks by 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. (11.28) that

$$b_0S^*(1) = A_o^*(1)A_m^*(1) = A_o(1)A_m(1)$$  \hspace{1cm} (11.31)

All the parameters in the $S$ polynomial are thus not free. If all parameters are estimated there is, of course, no guarantee that Eq. (11.31) holds. It is, however, easy to find a remedy. A polynomial $S^*$ with the property given by Eq. (11.31) can be written as

$$b_0S^* = A_o(1)A_m(1) + (1 - q^{-1})S'^*(q^{-1})$$
$$= A_o(1)A_m(1) + S'^*(q^{-1}) \Delta^*$$

Equation (11.30) then becomes

$$A_o^*(q^{-1})A_m^*(q^{-1})y(t + d) - A_o(1)A_m(1)y(t)$$
$$= b_0(R'^*(q^{-1})\Delta^*u(t) + S'^*(q^{-1})\Delta^*y(t))$$  \hspace{1cm} (11.32)
$$= R^*(q^{-1})\Delta^*u(t) + S^*(q^{-1})\Delta^*y(t)$$

Division by $A_o^*A_m^*$ now gives

$$y(t + d) - \frac{A_o(1)A_m(1)}{A_o^*(q^{-1})A_m^*(q^{-1})} y(t) = R^*(q^{-1})u_f(t) + S^*(q^{-1})y_f(t)$$  \hspace{1cm} (11.33)

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. (11.25) it follows from Eq. (11.32) that

$$R^*(q^{-1})\Delta^*u(t) + S^*(q^{-1})\Delta^*y(t) + A_o(1)A_m(1)y(t)$$
$$= A_o^*(q^{-1})A_m(1)u_c(t)$$
The following modification of the algorithm is introduced in order to avoid windup

\[
A^*_o(q^{-1})(v(t) - A_m(1)u_c(t)) = -A_o(1)A_m(1)y(t) - S^*(q^{-1})\Delta^*y(t) - (R^*(q^{-1})\Delta^* - A^*_o(q^{-1}))u(t) \\
u(t) = \text{sat } v(t)
\]  

(11.34)

In summary, Algorithm 11.1 is obtained.

**Algorithm 11.1 A direct self-tuning algorithm**

*Step 1:* Estimate the parameters in Eq. (11.33) by recursive least squares.

*Step 2:* Compute the control signal from Eq. (11.34) using the estimates from Step 1.

This algorithm may be viewed as a practical version of Algorithm 5.4.

### 11.6 Conclusions

Various practical aspects and implementation issues have been discussed in this chapter. Based on these it should be possible to implement adaptive algorithms that contain many of the "safety nets" that are necessary when using a control algorithm in practice. The main thing is to safeguard the estimator, the control algorithm as well as the design calculations. The estimation must be done on good data only. The control algorithm must include limitation of the control signal and anti-reset windup. The design calculation must be safeguarded against common factors in the Diophantine equation and possible division by small parameter values.

### Problems

11.1 How should the disturbance annihilation filter \( H_f(q) \) be chosen if \( d(t) \) in Eq. (11.1) is a sinusoidal?

11.2 Consider Example 11.1. How is the elimination of a level influenced by the choice of the parameter \( \alpha \)?

11.3 Introduce a dead zone like that in Eq. (11.12) into the exponential forgetting algorithm of Eqs. (11.5) (11.7).

11.4 Simulate the system in Example 11.3 and investigate the estimation when using (a) constant exponential forgetting factor; (b) constant exponential forgetting factor with dead zone; (c) regularized constant-trace estimation.
Chapter 11  Practical Aspects and Implementation

11.5 Plot the Bode diagram for a fourth-order Bessel filter and compare it with a pure time delay. Consider the cases in Table 11.2.

11.6 Determine how the behavior of the anti-reset windup controller of Eq. (11.22) is influenced by the filter $A_o^*$.

11.7 Complete the algorithm skeleton for the cases (a) direct self-tuner (Algorithm 5.4); (b) indirect self-tuner without zero cancelation.

11.8 Make a transformation of a second- and fourth-order Bessel filter $\omega_B = 1$, into band-pass filters using Eq. (11.23). Use $\omega_l = 100$ and $\omega_h = 1000$ rad/s. Compare the band-pass characteristics by using Bode diagrams.

11.9 Consider the Diophantine equation

$$AR + BS = A_c$$

and let

$$A(q) = (q - 1)(q - 0.9)$$

Use the method in Appendix A and compute the resulting controller when (a) $B(q) = q - 0.6$; (b) $B(q) = q - 0.9$. Assume that the desired closed characteristic polynomial is

$$A_m(q) = q^2 - q + 0.7$$

11.10 One way to solve the Diophantine equation is to multiply it by a persistently exciting signal, such as white noise. Introduce the filtered signals

$$u_a(t) = \frac{A}{A_mA_o} v(t) \quad u_b(t) = \frac{B}{A_mA_o} v(t)$$

The Diophantine equation then becomes

$$Ru_a + Su_b = v$$

The coefficients of the $R$ and $S$ polynomials can now be determined using the method of least squares, and one iteration can be done at each sampling instance. Discuss the merits and drawbacks of this approach. (*Hint: What is the convergence rate?)

11.11 Compare experimentally the behavior of the estimators that are based on Eqs. (11.32) and (11.33), respectively. Make sure that measurement noise is present in the simulations.
11.12 Consider the simulation shown in Fig. 11.9. Experiment with different types of filters $H_f$. Try to derive some rules of thumb for selection of the filter.

11.13 The simulations in Fig. 11.8 were all made with deadbeat observers. Investigate experimentally by simulation the effects of different observers. Develop guidelines for choosing the dynamics of the observer.

References

Implementation issues for adaptive controllers are discussed in:


Different design methods and their properties are treated in:


Aspects on implementation of estimation routines are found in:


Different ways to introduce integrators in adaptive controllers are treated in:


Design of low-pass and band-pass filters can be studied in:

Chapter 12

APPLICATIONS

12.1 Introduction

There have been a number of applications of adaptive feedback control over the past 30 years. Experiments with adaptive flight-control systems were done early in 1960. Industrial experiments with self-tuners were performed in 1972. Full-scale experiments with adaptive autopilots for ship steering were done in 1977. Special adaptive systems have also been in continuous use for a long time. Some process control loops have been running continuously since 1974. There are also some special products that have long been operating. Self-oscillating adaptive systems have been used in operational missiles. Commercial systems for ship steering have been in continuous operation since 1980.

However, only in the 1980s have adaptive techniques started to have real impact on industry. Systems implemented using minicomputers ap-
peared in the early 1970s. The number of applications increased drastically with the advent of the microprocessor, which made the technology cost-effective. Because of this, adaptive regulators are also entering the marketplace even in single-loop controllers.

Several commercial products based on adaptive techniques were introduced in the early 1980s, and second-generation versions have been introduced in some cases. Adaptive techniques are used in a number of products. Gain scheduling is the standard method for design of flight control systems for high-performance aircraft, and it is also used in robotics and process control. The self-oscillating adaptive system is used in several missiles. There are several commercial autopilots for ship steering and motor drives and adaptive systems for industrial robots. For process control, adaptive techniques are used both in single-loop controllers and in general-purpose process control systems. A rough estimate indicates that in May of 1988 there are at least 70,000 loops in which adaptive techniques are used. Most industrial processes are controlled by PID regulators, and a large industrial plant may have hundreds of them. Many instrument engineers and plant personnel are used to select, install, and operate such regulators. In spite of this, many regulators are poorly tuned. One reason is that simple, robust methods for automatic tuning have not been available. Adaptive methods have recently been applied to provide automatic tuning of PID regulators. This is in fact one of the fastest-growing areas of application for adaptive control.

However, adaptive techniques are still not widely used; the technology is not mature. Because of commercial enterprises' involvement in adaptive control, it is not always possible to find out precisely what is being done. Various ideas are hidden in proprietary information that is carefully guarded.

This chapter is organized as follows. An overview of some applications is given in Section 12.2. A number of commercial products that use adaptation are presented in Section 12.3. Some specific applications are presented in more detail in the sections that follow.

12.2 Status of Applications

A large number of experiments with adaptive control have been performed since the mid-1950s. The experiments have had different purposes: to verify ideas, to find out how adaptive systems perform, to compare different approaches and to find out in which processes they are suitable. The early experiments which used analog implementations, were plagued by hardware problems. When digital process computers became available, they were natural tools for experimentation. Experiments with adaptive con-
trol required substantial programming since adaptation was not part of the standard software. Applications proliferated with the advent of the microprocessor, which is a convenient tool for implementing adaptive systems. Adaptive techniques now appear both in single-loop controllers and as standard elements of large process control systems. There are tailor-made controllers for special purposes that use adaptive techniques.

**Feasibility Studies**

A number of feasibility studies have been performed to evaluate the usefulness of adaptive control. They cover a wide range of control problems, such as autopilots for missiles, ships, and aircraft, engine control, motion control, machine tools, industrial robots, power systems, distillation columns, chemical reactors, pH control, furnaces, heating, and ventilation. There are also applications in the biomedical area. The feasibility studies have shown that there are cases in which adaptive control is very useful and others in which the benefits are marginal. Some industrial products also use adaptive techniques. There are both general-purpose controllers and regulators for special applications. Some industrial products are discussed in more detail in Section 12.3. Some generic uses of adaptive techniques are discussed below.

**Auto-tuning**

Simple regulators with two or three parameters can be tuned manually if there is not too much interaction between the adjustments of different parameters, but manual tuning is not possible for more complex regulators. Traditionally, tuning of such regulators has taken the route of modeling or identification and regulator design. This is often a time-consuming and costly procedure, which can only be applied to important loops or to systems that are to be manufactured in large quantities.

All adaptive techniques can be used to provide automatic tuning. In such applications the adaptation loop is simply switched on. Perturbation signals may be added to improve the parameter estimation. The adaptive regulator is run until the performance is satisfactory, then the adaptation loop is disconnected and the system left running with fixed regulator parameters. The particular methods for automatic tuning of PID regulators discussed in Chapter 8 have been found particularly attractive, because they require little prior information and they are closely related to standard industrial practice.

Auto-tuning can be considered as a convenient way to incorporate automatic modeling and design in a regulator. It simplifies the use of the regulator and it widens the class of problems in which systematic design methods can be used cost-effectively. This is particularly useful for design methods such as feedforward which depend critically on good models.
Automatic tuning can be applied to simple PID controllers as well as to more complicated regulators. It is very convenient to introduce tuning into a DDC package because the tuning algorithm can serve many loops. Auto-tuning can also be included in single-loop regulators. For example it is possible to obtain standard regulators in which the mode switch has three positions: manual, automatic, and tuning. A well-designed autotuner is very easy to use, even for unskilled personnel. Experience has shown it to be useful both for commissioning of new systems and for routine maintenance. Auto-tuners can also be used to enhance the skill of the instrument engineers.

Automatic tuning will probably also be a useful feature of more complicated regulators.

**Automatic Construction of Gain Schedules**

Gain scheduling is a very useful technique, but it has the drawback that it may be quite time- and cost-consuming to build a schedule. Auto-tuning can conveniently be used to build gain schedules. A scheduling variable is first determined. The parameters obtained when the system is running in one operating condition are then stored in a table together with the scheduling variable. The gain schedule is obtained when the process has operated at a range of operating conditions that covers the operating range.

**True Adaptive Control**

The adaptive techniques may, of course, also be used for genuine adaptive control of systems with time-varying parameters. There are many ways to do this. The operator interface is important, since adaptive regulators also have parameters that must be chosen. Regulators without any externally adjusted parameters can be designed for specific applications, in which the purpose of control can be stated a priori. The ship steering autopilot discussed in Section 12.4 is a typical example. In many cases, however it is not possible to specify the purpose of control a priori. It is at least necessary to tell the regulator what it is expected to do. This can be done by introducing dials that give the desired properties of the closed-loop system. Such dials are characterized as performance-related. New types of regulators can be designed using this concept. For example, it is possible to have a regulator with one dial, labeled with the desired closed-loop bandwidth. Another possibility would be to have a regulator with a dial that is labeled with the weighting between state deviation and control action in a LQG problem. Adaptation can also be combined with gain scheduling. A gain schedule can be used to get the parameters quickly into the correct region and adaptation can then be used for fine-tuning.
Feedforward control is very useful when there are measurable disturbances. However, feedforward control, being an open-loop compensation, requires good models of process dynamics. Adaptation therefore appears to be a prerequisite for effective use of feedforward compensation.

Abuses of Adaptive Control

An adaptive regulator is more complex than a fixed-gain regulator, since it is nonlinear. Before attempting to use an adaptive regulator, it may therefore be useful to investigate whether the problem can be solved with a robust constant-gain regulator, as discussed in Chapter 10. As pointed out in Chapter 2, it is not possible to judge the need for adaptation from the variations in the open-loop dynamics. The open-loop responses may vary much while the closed-loop responses are close and vice versa.

The complexity of the regulator has to be balanced against the engineering effort required to make the system operational. Experience has shown that only a modest effort is required to make a standard adaptive system work well.

12.3 Industrial Adaptive Controllers

A number of standard industrial products that incorporate adaptive control techniques will be presented in this section. The products range from auto-tuners to general purpose toolboxes for adaptive control.

Overview

Leeds and Northrup announced a PID controller with a self-tuning option in 1981. Asea Brown Boveri of Sweden presented a general-purpose adaptive regulator in 1982. SattControl in Sweden announced auto-tuning for PID regulators in a small DDC system in 1984 and a single-loop controller with auto-tuning in 1986. Foxboro introduced a self-tuning regulator in 1984, and Turnbull Control in the U.K. announced an adaptive PID controller in the same year. First Control Systems in Sweden introduced an adaptive regulator in 1986. In 1987 Yokogawa announced an adaptive PID regulator, which appears to have features similar to Foxboro's EXACT. Fisher Control presented its DPR 900 auto-tuning PID controller in 1987. These products will be discussed in more detail later in this chapter. We will start with the systems that are based on PID control, and this section ends with a few comments on developments.

A number of devices can be connected to a process to tune PID regulators. One is the Supertuner from Toyo Systems in Japan, another is the Protuner from Techmatlon in Arizona, and a third one is the Micon P-200 Controller from Powell-Process Instruments in the United States.
Electromax V  The Leeds and Northrup Adaptive Regulator

This regulator is an adaptive single-loop controller based on the PID structure. The regulator has an adaptive function as an option. The adaptation is a self-tuning regulator in which a second-order discrete-time model is estimated. The parameters of a PID regulator are then computed from the estimated model using a pole placement design. Different sampling rates are used in the parameter estimator and the control algorithm. The regulator is primarily intended for temperature control. The regulator can operate in three different modes: fixed, self-tune, and self-adaptive. In the fixed mode the regulator operates like an ordinary fixed-gain PID regulator. In the self-tune mode a perturbation signal is automatically introduced, a model of the process is estimated, and PID parameters are computed from the model. The parameters are displayed to an operator, who may accept or reject the new parameters. In the self-adaptive mode the parameters are updated continuously.

Parameter Estimation. Parameter estimation is performed in the self-tune and the self-adaptive modes, and estimation is performed in closed loop in both cases. Setpoint changes are generated automatically in the self-tune mode, to ensure that the estimation is based on good data. The changes are cycles of pulses. The pulse height (acceptable setpoint upset) is set by the operator. The cycle time is computed from the response time, which is also set by the operator. The goodness of fit is determined after each cycle; when the fit is good enough, a message is given. The operator may examine the regulator parameters obtained. If the fit is still poor after five cycles the procedure is aborted and a message is given. The estimation method is a gradient procedure for a discrete-time transfer function model. The sampling period is computed from the response time. The details have not been made public.

In the self-adaptive mode parameter estimation and computation of PID parameters are repeated at each sampling period. The estimated model is a second-order pulse transfer function. The parameters are estimated using the instrumental variable method, in which components of the regression vectors are formed from the model output. The signals are high-pass filtered before they are fed to the estimator. The parameter updating is discontinued when the control error is below a certain limit. The estimation technique used requires that a sampling interval be chosen. To do this it is necessary to know the time scales; this is one reason the pre-tuning mode is required.

Prior Information. To use the regulator it is necessary to specify five numbers: nominal values of the PID parameters, the process response time, and the admissible setpoint upset. The nominal values of the regulator parameters are needed because the estimation is done in closed loop. The
process response time is defined as the time it takes for the open-loop step response to reach 90% of the steady-state value. This number is used to determine the perturbation signal in the identification phase, the sampling period of the discrete-time model, and the desired response of the closed-loop system. Performance is quite sensitive to the choice of the response time. The admissible setpoint upset gives the amplitude of the pulses used in the identification phase. Tests are performed to make sure that the output does not saturate.

Pretune. If good estimates of the prior data are not available it is recommended that an open-loop step response be performed.

Industrial Experiences. The Electromax V was the first adaptive standard product, and there is considerable experience of its use. Part of this has also been published. The regulator can operate both as a PID tuner and as an adaptive PID regulator. The majority of applications are in temperature control. The experiences are generally quite favorable, although it is noted that adaptive control is not a panacea. Most of the benefits are derived from self-tuning, but in a number of cases the continuous adaptation has been profitable. In many of the cases in which adaptation has been found useful, the variations in process dynamics are related to operating conditions. Gain scheduling combined with auto-tuning would then be an alternative.

Difficulties in using the regulator have been observed with processes that have unsymmetric process response (typically, heating and cooling), rapid parameter variations, or strong nonlinearities. The regulator cannot be applied to processes such as silicon crystal growing that do not tolerate the process upsets required in the identification phase. Difficulties with regulators used in the self-adaptive mode have also been found under operating conditions in which the measured value is suddenly disconnected. The parameter estimation is then done on totally irrelevant data. The remedy is to stop the parameter updating when the output is disconnected.

EXACT The Foxboro Adaptive Regulator

This regulator is based on analysis of the transient response of the closed-loop system to setpoint changes or load disturbances and traditional tuning methods of the Ziegler-Nichols type.

Parameter Estimation. Assuming regulator parameters such that the closed-loop system is stable, a typical response of the control error to a step or impulse disturbance is shown in Fig. 12.1. Heuristic logic is used to detect that a proper disturbance has occurred and to detect the peaks $e_1$, $e_2$, and $e_3$ and period $T_p$. The estimation process is simple, but it is based on the assumption that the disturbances are steps or short pulses.
The algorithm can give wrong estimates if the disturbances are two short pulses, because $T_p$ will then be estimated as the distance between the pulses.

Control Design. The control design is based on specifications on damping, overshoot, and the ratios $T_i/T_p$ and $T_d/T_p$, where $T_i$ is the integration time, $T_d$ the derivative time, and $T_p$ the period of oscillation. The damping is defined as

$$d = \frac{e_3 - e_2}{e_1 - e_2}$$

and the overshoot as

$$o = -\frac{e_2}{e_1}$$

In typical cases it is required that both $d$ and $o$ are less than 0.3. Empirical rules are used to calculate the regulator parameters from $T_p$, $d$, and $o$. These rules are based on traditional tuning rules of the Ziegler-Nichols type, augmented by experiences from regulator tuning. The details are not available.

Prior Information. The tuning procedure requires prior information on the regulator parameters $K$, $T_i$, and $T_d$. It also requires information on the time scale of the process. This is used to determine the maximum time the heuristic logic waits for the second peak. Some measure of the process noise is also needed to set the tolerances in the heuristic logic. Some parameters may also be set optionally: damping $d$, overshoot $o$, maximum derivative gain, and bounds on the regulator parameters.

Pretune. The tuning procedure requires reasonable regulator parameters to be known so that a closed-loop system with a well-damped response is obtained. There is a pretune mode that can be used if the prior information needed is not available. A step test is done when the user specifies the step size. Initial estimates of the regulator parameters are determined from the
step, and the time scale and the noise level are also determined. The pretune mode can be invoked only when the process is in steady state.

*Industrial Experiences.* About 30,000 EXACT controllers were in operation as of December 1987. The system will also be available in Foxboro’s system for distributed process control. Users from a large number of installations have reported favorably, citing the ease with which regulators can be well tuned and the ability to shorten commissioning time. It is also mentioned that derivative action can often yield significant benefits.

**SattControl ECA40 and Fisher Control DPR 900**

This is the original auto-tuner based on relay oscillations, as described in Chapter 8. It was first introduced in a small (about 45 loops) DDC system for industrial process control SDM20. In this application the tuner can be connected to tune any loop in the system. Relay auto-tuning is also available in single-loop PID controllers (SattControl ECA40 and Fisher Control DPR900). In these regulators tuning is done on demand by pushing a button on the front panel, so called *one-button tuning.* The regulators are also provided with facilities for gain scheduling. There is a table with three regulator settings.

*Parameter Estimation.* The ultimate period and the ultimate gain are determined by an experiment with relay feedback. The fluctuations in the output signal are measured, and the hysteresis of the relay is set slightly wider than the noise band. The initial relay amplitude is fixed. The amplitude and period are measured for each half-period. A feedback adjusts the relay amplitude so that the limit cycle oscillation has a given amplitude. When two successive half-periods are sufficiently close, PID parameters are computed and PID control is initiated automatically.

*Control Design.* When the ultimate gain and the ultimate period are known, the parameters of a PID regulator can be determined by a modified Ziegler-Nichols rule. There is also a limited amount of logic to determine whether derivative action is needed.

*Prior Information.* A major advantage of the auto-tuner is that no parameters have to be set *a priori.* To use the tuner the process is simply brought to an equilibrium by setting a constant control signal in manual mode. The tuning is then activated by pushing the tuning button. The regulator is automatically switched to automatic mode when the tuning is complete. Different control objectives may be obtained by modifying the parameters in the Ziegler-Nichols rule. One mode is chosen by default but the user can request a slower or an extra-fast response.

*Industrial Experiences.* About 2,000 ECA40s are in operation as of December 1987. Both the auto-tuning and the gain-scheduling features have
been found to be very useful. The system has been considered very easy to use, even by inexperienced personnel. In many applications it has contributed significantly to improved tuning. It has also been demonstrated that commissioning time can be shortened significantly by using automatic tuning, and that the standard regulator can be applied to processes having a wide range of time scales. Simplicity is the major advantage of the auto-tuner. This has proven particularly useful for plants that do not have qualified instrument engineers and for operation during the night shift, when instrument engineers are not available. It is also easy to explain the auto-tuner to the instrument engineers. The properties of the auto-tuner are illustrated by two examples.

Example 12.1  Level control
Figure 12.2 shows the behavior of the regulator when used to control the level of a vessel in a pulp mill. A regulator with pure proportional action was used originally, which resulted in the steady-state error shown in the figure. The tuning took about two minutes and resulted in a PI regulator. This example illustrates the usefulness of the logic for selecting control action. The figure also indicates the control quality.

Although the auto-tuner has been used mostly for conventional loops for control of flow, pressure, and level, it has also been applied to more difficult problems. One example is given below.

Example 12.2—Temperature control in a distillation column
This is a conventional control loop in which the temperature in a tray is measured and the boil-up is manipulated. This control loop was part of a process system with many loops. There had been severe problems with the
temperature control for a long time, and several attempts had been made to tune the loop. Figure 12.3 shows a recording of the temperature. The figure shows that the loop is oscillatory with the regulator tuning that was used ($K_c = 8$, $T_i = 2000$ and $T_d = 0$). Also notice the long period of the oscillation. The regulator was switched to manual at time 11:30, and the temperature then started to drift. Auto-tuning was initiated at time 14:00. The tuning phase was completed at time 20:00, when the regulator was automatically set to automatic control. The regulator parameters obtained were $K_c = 1.3$, $T_i = 4300$, and $T_d = 1100$. Notice that the whole tuning procedure is fully automatic. The only action taken by the operator was to initiate tuning at time 14:00. The temperature variations during tuning are not larger than those obtained with the conventional regulator settings. The example shows that the auto-tuner could cope with a process having drastically different time scales than those normally used.

**Eurotherm Temperature Controller**

Temperature control is traditionally done with simple PID regulators, which are cheaper than conventional industrial controllers. Auto-tuning is
Figure 12.4 Tuning procedure used in the Eurotherm temperature controller.

now starting to be used in such simple systems. One example is regulators produced by Eurotherm in the U.K. A modified relay tuning is used in those regulators (see Fig. 12.4). Full control power is used until an artificial setpoint is reached. Two half-periods of a relay tuning are then used, and the regulator parameters are calculated from the transient. The regulator also has facilities for automatic on-line tuning based on transient response analysis. There are over 25,000 regulators of this type.

Novatune—The Asea Brown Boveri Adaptive Regulator

The Asea Novatune is an adaptive regulator that is incorporated as a part of a distributed system for process control. The system is block-oriented which means that the process engineer creates a system by selecting and interconnecting blocks of different types. The system has blocks for conventional PID control, logic, and computation. Three different blocks called STAR1, STAR2, and STAR3 are adaptive regulators. The adaptive regulators are self-tuning regulators based on least-squares estimation and minimum-variance control. The regulators all use the same algorithm; they differ in the regulator complexity and the prior information that must be supplied when using them.

The Novatune differs from the other regulators discussed previously in that it is not based on the PID structure. Instead, its algorithm is based on a general pulse transfer function. It also admits dead-time compensation
and feedforward control. The Novatune system may be viewed as a toolbox for solving control problems. It is now also incorporated in Asea Master, which is a distributed system for process control. The product is called Asea Master Piece.

Principle. The Novatune is a direct self-tuning regulator similar to Algorithm 5.2 in Section 5.3. The parameters of a discrete-time model are estimated using recursive least squares. The control design is a minimum-variance regulator, which is extended to admit positioning of one pole and a penalty on the control signal. The block diagrams in Fig. 12.5 shows two of the adaptive modules. The Novatune system has three adaptive modules, STAR1, STAR2, and STAR3. The latter is the most complicated. The simpler ones have fewer inputs and have default values on some of the parameters in STAR3. In the block diagram the input signals are shown on the left and top sides of the box, the output signals on the right, and the parameters on the bottom. The parameters can be changed at configuration time. The parameters $PL$, $T$, and $PN$ can also be changed on-line.

The simplest module, STAR1, has three input signals: the manual input $UEXT$, the measured value $FB$, and the setpoint $REF$. It has three parameters. The variable $PY$ is the smallest relevant change in the feedback signal; the adaptation is inhibited for changes less than $PY$. The parameters $MAX$ and $MIN$ denote the bounds on the control variable,
and $T$ is the sampling period.

The module STAR2 has more input signals. It admits a feedforward signal $FF$. There are also four signals, $HI$, $LO$, $DH$, and $DL$, that admit dynamic changes on the bounds of the control variable and its rate of change. There are also additional parameters: $PN$, for a penalty on the control variable, and $KD$, which specifies the prediction horizon. The module also has two additional mode switches: $REGAD$, which turns off adaptation when false and $SOFT$ which allows a soft start.

The module STAR3 has an additional function $LOAD$, which admits parameters stored in an EEPROM to be loaded. It also has several additional parameters, which admit positioning of one pole $PL$ and specification of controller structure $NA$, $NB$, $NC$, and $INT$.

**Parameter Estimation.** The parameter estimation is based on the model

$$(1 - PLq^{-1})y(t + KD) - (1 - PL)y(t) = A(q^{-1})\Delta y(t) + B(q^{-1})\Delta u(t) + C(q^{-1})\Delta v(t)$$

where $A$, $B$, and $C$ are polynomials in the delay operator $q^{-1}$ $y$ is the measured variable, $u$ the control signal, $v$ a feedforward signal, and $\Delta$ the difference operator $1 - q^{-1}$. Compare with Algorithm 11.1. The integers $NA$, $NB$, and $NC$ give the number of coefficients in the polynomials $A$, $B$, and $C$. The number $PL$ is the desired pole location for the optional pole. When parameter $INT$ is zero, a similar model without differences is used. The parameters are estimated using recursive least squares with a forgetting factor $\lambda = 0.98$. Parameter estimation is suspended automatically when the changes in the control signal and the process output are less than $PU$ and $PY$. The parameter updating may also be suspended on demand through the switch $REGAD$. In combination with other modules in the Novatune system, this constitutes a convenient way to obtain robust estimation.

**Control Design.** The control law is given by

$$(\rho + B(q^{-1}))\Delta u(t) = (1 - PL)(u_c(t) - y(t)) - A(q^{-1})\Delta y(t) - C(q^{-1})\Delta v(t)$$

where $\rho$ is a penalty factor related to $PN$. Since the algorithm is a direct self-tuner, the regulator parameters are obtained directly from the estimated parameters.

**Industrial Experiences.** The Novatune has been applied to a wide range of process control problems in the steel, pulp, paper, and petrochemical industries waste water treatment, and climate control. Some applications
have given spectacular improvement of performance compared to PID control. This is particularly the case for processes with time delay, in which adaptive feedforward can be used. It has also been used to make special-purpose systems for special application areas like paper winding and climate control. Some Novatune applications are described in more detail in Section 12.4. As of 1987 there were about 2500 loops adaptively controlled by Asea Master Piece/Novatune. The essential drawback of the Novatune is that it is based on a direct self-tuner. This means that the sampling period and the parameter $KD$ have to be chosen with care. It may, for example, be difficult to use very short sampling periods.

**Firstloop**  The **First Control Adaptive Regulator**

The adaptive system Firstloop was developed by First Control Systems, a small company founded by members of the Novatune team. Firstloop is a small regulator module with up to eight self-tuning regulators. The system is a toolbox with modules for adaptive control, logic, filtering square root functions, and operator communication. An interesting feature is that the adaptive regulator is the only regulator available in the system. However, by choosing the number of parameters of the estimated model it is possible to get different controller structures for instance a PID controller. The software admits easy configuration of a control system. The Firstloop controller is shown in Fig. 12.6. The adaptive control module will be described in detail. Firstline is a distributed process control system with a block-oriented language for control design. The adaptive controller is incorporated as a standard function module.
### Figure 12.7 The expert module STREGX in Firstloop.

**Principle.** The adaptive control unit used in Firstloop and Firstline is based on recursive estimation of a transfer function model and a control law based on indirect pole placement. The controller also admits feedforward. The main advantage of using an indirect pole placement algorithm is that the system can be applied to non-minimum-phase systems and systems with time-varying time delays. This also implies that short sampling periods can be used. (Compare the discussion in Section 6.8.) The adaptive module comes in two versions, a standard module and an expert module. The standard module is intended to be used by ordinary instrument engineers who are not specialists in adaptive control. The expert module shown in Fig. 12.7 is intended for specialists in adaptive control. Many parameters are given default values in the standard module. The variables that must be specified are shown in Fig. 12.7. The signal connections are measured value \( MV \) setpoint \( SP \), external control signal \( UE \), feedforward \( FF1 \) \( FF2 \), and regulator output \( U \). The mode switches \( ON \), \( AUTO \), and \( ADAPT \) are for on/off, auto/normal, and adaptation on/off, respectively. Parameters \( UMAX \) and \( UMIN \) define the actuator range. Variables \( HI \), \( LO \) \( DUP \), and \( DUM \) specify the limits on the control signal that are used internally in the regulator. The performance related parameters are \( POLE \), which gives the desired closed-loop pole, and \( BMPLVL \), which gives the admissible initial change of the control variable at mode switches.
The desired closed-loop pole is the major variable to be selected. The choice of this variable clearly requires knowledge of the time scales of the process. The recommended rule of thumb is to start with a large value and gradually decrease it.

Parameter Estimation. The parameters of a high-order transfer function model are estimated. Systems with variable time delay can be captured provided that a large number of \( b \) parameters are used. Up to 15 parameters can be estimated in the model. The number of parameters in the model is specified by \( NA \), \( NB \), and \( NC \). Common factors in the pulse transfer function are canceled automatically.

Control Design. The control design is based on pole placement. The desired response is characterized as a first-order system with delay. The remaining poles are positioned at the origin. The design of the algorithm is based on solving the Diophantine equation, by a method that cancels common factors in the polynomials \( A \) and \( B \). (Compare Appendix A.) The details of the control design are proprietary.

Safety Network. The algorithm is provided with extensive safety logic. Adaptation is interrupted when variations in measured signals and control signals are too small. The limits are given with the parameters \( RESU \) and \( RESY \). Adaptation is also interrupted when the control error is below a certain limit, and there are safeguards to ensure that the influence of a single measurement error or sudden large disturbance is limited. (Compare Section 6.6.) Measured values that result in large model errors are also given a low weight automatically. The details of the safety logic are not available. Different models can be stored for use in different situations. The regulator is initialized by a model number equal to \( MODEL \) when \( LOAD \) changes from false to true.

Industrial Experiences. Firstloop and Firstline are used in a number of high-performance process control systems. About 200 loops were installed in 1987. They include control of pulp mills, paper machines, rolling mills, and pilot plants for chemical process control.

Discussion

The products described give an idea of how adaptive techniques are used in commercial standard products. Some insight can be derived by analyzing the existing products and trends. Experience from the applications clearly indicates the need for tuning and adaptation; there are undoubtedly many control loops that are poorly tuned. This results in loss of energy, quality, and effective production time. It is also of interest that many different techniques are used and there are also promising adaptive algorithms that have not yet reached the marketplace. A few specific issues will be discussed in more detail.
Computing Power. Industrial use of adaptive methods has been possible thanks to the availability of microprocessors, and the technique has also benefited from advances in microelectronics. Most of the commercial systems are based on 8-bit processors, with their inherent limitations in addressing capability. This applies to all the PID auto-tuners and the first version of Novatune that used less than 64 kbyte of memory. With 16-bit processors and larger address spaces it is possible to use more sophisticated algorithms and better man-machine communication. The PID auto-tuners typically run with sampling rates up to 50 Hz. Systems like Novatune and Firstloop can use 100 Hz or more as sampling rates.

Intentional Perturbation Signals. To estimate parameters it is necessary to have data in which there are variations in the control signal. Such variations can be generated naturally or introduced intentionally. Natural perturbations can occur because of disturbances or poorly tuned regulators. Intentional perturbations can be introduced when natural perturbations are not present, as suggested by dual control theory. This method is used in several of the auto-tuning schemes. If prior information about the system dynamics is available, it is possible to find signals that are optimal for the purpose of estimating parameters. Relay feedback automatically generates an input signal having a lot of energy at the frequency at which the process has a phase lag of 180°. Although intentional perturbation signals both are useful and justified by theory they are often controversial. It should be remembered, however, that poorly tuned regulators may also be considered as perturbations.

Regulator Structures. Different regulator structures are used in the commercial systems. There are both PID controllers and general transfer function systems that admit feedforward and compensation for dead time. The main advantage of the PID structure is that it is close to current industrial practice. Within the PID family there are cases in which derivative action is of little benefit. Systems like the SattControl ECA40 can determine this and choose PI action automatically. However, there is no system that can choose regulator structure generally, although it seems possible to design such systems. The benefits of feedforward control from measurable disturbances have been known for a long time. Experience with Novatune and Firstloop clearly shows the benefit of adaptive feedforward control. Since feedforward control critically depends on a good model adaptation is almost a prerequisite for feedforward control. Adaptive controllers like the Novatune and Firstloop use a regulator structure that is a general transfer function model like

\[ R(q)u(t) = T_1(q)u_c(t) + T_2(q)v(t) - S(q)y(t) \] (12.1)

where \( u \) is the control variable, \( u_c \) the command signal, \( v \) a measured disturbance, and \( y \) the controlled output. The polynomials \( R, S, T_1, \) and
$T_2$ can be chosen so that the regulator corresponds to a PID regulator. The regulator Eq. (12.1) can, however, also be much more general than a PID controller. It can incorporate many classical features such as filtering, disturbance models, Smith predictors, and notch filters. For more demanding control problems the general transfer function regulator thus has significant advantages over the PID regulator. However, more expertise in control engineering is needed to understand and interpret the parameters of a regulator like Eq. (12.1). Since the PID regulator is so common, we can expect it to coexist with more general regulators for a long time.

To a limited extent, multivariable control problems can be handled using the feedforward feature in Novatune and Firstloop. None of the commercial systems admits truly multivariable adaptive control.

**Pre-tuning.** It is interesting to note that many schemes have been provided with a pre-tuning feature. In some cases it appears that this was added afterwards. The reason is undoubtedly that too much user expertise is required for the standard algorithms. The selection of sampling periods or the equivalent time scales is a typical example. It appears that the relay method for automatic tuning would be an ideal method for pre-tuning.

**Tuning Automatically or on Demand.** The existing products include systems in which tuning is initialized on demand from the operator or automatically. Users of both schemes have documented their experiences. It appears that there are a number of processes for which regulators should be retuned for different operating conditions. See, for instance, the remark on industrial experience in the discussion of the Leeds and Northrup regulator. In many cases there are measurable signals that correlate well with the operating conditions. In these cases it seems that the combination of on-demand automatic tuning with gain scheduling is a good solution. This will give systems that change parameters more quickly than systems with adaptation. Of course, it is convenient to have tuning initiated automatically, but it is difficult to give general guidelines for when tuning should be initiated. The simple schemes currently in use are often based on simple level detection. Further research is required to find conditions for retuning; this is discussed further in Section 13.5.

An analysis of the division of labor between man and machine gives another viewpoint on the question of on-demand or automatic tuning. When tuning is done on demand of the operator, the ultimate responsibility for tuning clearly remains with the operator or the instrument engineer. This responsibility is carried even further in some systems, in which the instrument engineer has to acknowledge the tuned values before they are used. A good solution would be a system in which the responsibility and the tuning techniques could be moved from the man to the computer system. Ideally, the system should also allow the operator to learn more about control in general and the particular process in ques-
tion. Experimental architectures that allow this are available but not in commercial systems.

Requirements for the User

The requirements for the user are very different for the various commercial systems. The PID regulators in which tuning is initiated automatically require very little. Regulators with on-demand tuning require somewhat more knowledge on the part of the user. Systems like Novatune and Firstloop can be regarded as toolboxes for solving control problems that are more demanding. They also allow complex control systems to be configured. This is clearly illustrated by the experiences from Novatune installations. The system was designed by a very qualified team that included several first-rate Ph.D.s. The design team was also responsible for many of the initial installations, which were extremely successful.

More recent versions of the toolbox systems are much easier to use. Moderate-sized systems have been successfully implemented by instrument engineers with little knowledge of advanced control. There are several reasons for the increased user-friendliness of the systems. The safety logic has been improved significantly; modules in which many parameters are given default values have been designed; and computer-based configuration tools, with a lot of knowledge built-in, have been developed. The toolboxes thus allow a user to get started quickly with a modest knowledge of adaptive control, and they also make it possible for a user to make more advanced systems when more knowledge is acquired.

12.4 Some Novatune Applications

It was mentioned in Section 12.3 that the Novatune is a toolbox for solving control problems with tools for adaptive control. A good insight into applications of adaptive control can be derived from Novatune applications some of which will be described in this section.

Chemical Reactor Control

Chemical reactors are typically nonlinear. Characteristics such as catalyst activity change with time, as does the raw material. There are often inherent time delays, which may vary with production level. Poor control can result in lower product quality, damage to the catalyst, or even explosions in exothermic reactors. Chemical reactors are therefore potential candidates for adaptive control. The process in this application consists of two parallel chemical reactors in which ethylene oxide is produced by catalytic oxidation of ethylene. The process is exothermic and time-variable because
of changes in catalyst activity. It is essential to keep the temperature accurately controlled: a reduction of temperature variations improves the yield and prolongs the life of the catalyst. Stable steady-state operation is also a first step towards plant optimization.

The plant was equipped with a conventional control system that used PID regulators to control flow and temperature. The plant personnel were dissatisfied with the system because it was necessary to switch the regulators to manual control in case of many major disturbances, which could happen several times per day.

A schematic diagram of the process is shown in Fig. 12.8. The reactor is cooled by circulating oil to a cooler. The temperature of the coolant at the inlet to the reactor is the primary controlled variable, and the reactor outlet temperature and the coolant flow are also measured. The control signal is the flow to the cooler. The dynamics relating temperatures and flow to valve openings have variable delays and gains.

Disturbances in the process are caused by variations in the incoming gas and load changes. Large disturbances occur with changes in production level or with "shutdowns" caused by failure in surrounding process equipment. During shutdowns it is most important to maintain the process temperature as long as possible so that the production can be restarted easily. With the conventional control system, temperature fluctuations were around ±0.5°C during normal operation and up to ±2°C during larger disturbances. With adaptive control the variations were reduced to ±0.1°C during normal operation and ±0.5°C during large upsets.

The adaptive control system was implemented using the STAR3 mod-
Figure 12.9  Schematic diagram of pulp drying and the control system.

ule with feedforward from the reactor outlet temperature. Using the other modules in the system, it was also straightforward to handle the dual valves and to reset to manual mode for startup and shutdown.

The system has been in continuous operation since 1982 on a reactor at Berol Kemi AB, which produces 30,000 tons per year. The operational experiences with the system have been very good. With adaptive control it was possible to reduce the temperature fluctuations significantly. The regulators are now kept in automatic mode most of the time, even during production changes. This has made it possible to revise operational procedures, since operators do not have to spend their time supervising the reactor temperature.

Pulp Dryer Control

Drying processes are common in process industries. The mechanisms involved in drying are complex and poorly understood, and their dynamics depend on many changing factors. There are often significant benefits in improved regulation, since an even moisture content is an important quality factor. There are also significant potential energy savings. Drying processes are thus good candidates for adaptive control.

In pulp drying, a wet pulp sheet passes a steam-heated drying section and cooling section. A typical system is shown schematically in Fig. 12.9. The moisture content of the sheet entering the dryer is about 55%. At the exit it is typically 10–20%. It takes about nine minutes to pass the dryer and about half a minute to pass the cooler. The dryer dynamics are complicated. It is influenced by many factors, such as the pH of the sheet. The measurements of the moisture content are obtained by a traversing microwave sensor that moves back and forth across the pulp sheet, describing a diagonal pattern on the sheet. When one traverse movement
is complete, the mean value of the diagonal is stored in the computer, the mean value algorithm is reset, the sensor moves back, and the procedure repeats itself. It takes a little less than one minute for the sensor to move across the sheet. With manual control the fluctuations in moisture content often exceed ±1%. The control system configuration is shown in Fig. 12.9. The moisture control is carried out by an adaptive software regulator STAR. The moisture content measured by the traversing system is low-pass filtered and connected to the \( FB \) input of the STAR. The desired moisture content is chosen by the operator outside the Novatune and software connected to the \( REF \) input of the STAR. The pH value, measured in an earlier process section, is used as feedforward signal. This signal is connected to the \( FF \) input of the STAR. The control signal of the STAR defines the desired steam pressure, which is measured and controlled to the desired value by a conventional hardware PI regulator. The control signal of this regulator acts continuously upon the steam flow valve.

The sampling period used in the adaptive regulator was 3.5 minutes. A fourth-order Butterworth filter was used as an anti-aliasing filter. This was implemented using the Novatune tools. When the production rate was changed, large upsets were noticed, lasting for about 30 minutes, because it took 5 15 samples for the adaptive regulator to settle. It was highly desirable to reduce these upsets, and this was done by introducing a special production rate compensation in the form of a pulse transfer function of the type

\[
H(z) = \frac{b(z - 1)}{z - a}
\]

This gives a rapid change of the steam pressure when pulp speed changes. It was not necessary to make this filter adaptive. The system has been in operation since 1983 at a pulp mill at Mörrum's Bruk that produces 330,000 tons of paper pulp per year. The operational experiences have been very good. Fluctuations in moisture content have been reduced from 1% to 0.2% which improves quality. It also allows the setpoint to be moved closer to the target value, which results in significant energy savings.

**Control of a Rolling Mill**

The process control applications are typical steady-state regulation problems. The rolling mill control problem is much more batch-oriented. It illustrates the use of adaptive techniques in machine control. There are many types of rolling mills, each with its specific control problem. This particular application deals with a skin pass mill located at the end of the production line. The material processed by the mill may vary significantly in dimension and hardness.

The purpose of the mill is to influence quality variables such as hardness and yield limit. A schematic diagram of the process is shown in
Fig. 12.10. Let $v_1$ be the speed of the strip entering the mill and $v_2$ the speed of the strip at the exit. Due to the thickness reduction, the exit speed is larger than the entrance speed. The elongation is defined as

$$\varepsilon = \frac{v_2 - v_1}{v_1}$$

The key control problem is to keep a constant elongation. There is a difficult measurement problem, since the velocity difference is so small. The process operates over a wide range of conditions; the following operating modes can be distinguished.

- Slow rolling at low speed during start-up
- Acceleration to fast rolling
- Fast rolling at production speed
- Intermediate decelerations to slow rolling or even to standstill
- Deceleration to slow rolling at the end of the strip
- System at rest waiting for the next strip

Transition from one mode to another is performed automatically on demand from the operator. It is essential that the control system handles these transitions well. The process dynamics relating elongation to roll force can be described as a high-order dynamical system with an open-loop response time of less than 0.05 s. Changes in production rate from 0 to 2000 m/min in less than 10 s are typical. The dynamics change drastically
during the operation; the dynamics of rolling change due to variations in the speed, hardness, and dimension of the strip. There are also significant changes of the inertia of the coilers. All material starts on one coiler and ends up on the other. There are variations in the oil film on the roller bearings due to variations in speed and pressure. The dynamics of the hydraulic system vary with the operating point.

The changes in dynamics due to changing speed are predictable and can (in principle) be taken care of by gain scheduling. Variations in dimension can be handled similarly. The hardness cannot be measured directly on-line, so it must be handled by feedback and adaptation.

A block diagram of the control system is shown in Fig. 12.10. The speed variations are taken care of in an elegant way. In the Novatune, sampling can be triggered by an arbitrary signal. In this case it is triggered by the pulse counters that measure strip speed. This means that sampling is related to the length of the strip, not to time. This is a simple way of making the control system invariant to strip speed (the same idea was used in the ship steering example in Section 9.5). The measurement of the velocity difference is implemented using pulse generators and counters.

For each strip a saved model is loaded into the regulator, and the adaptation is switched on with some delayed action (15 sampling intervals) to avoid adaptation during the first few steps, in which the measurement is irregular. The initial model is taken from a soft strip so that there will be no excessive control action at start-up. Soon enough, the regulator will adapt to the conditions of the new strip.

Figure 12.11 illustrates a typical run of a strip. Notice in particular how well the system copes with the velocity variations and with the mode changes. The installation of the system took about a week, mostly devoted to function and signal checking and tests. The regulator functioned almost immediately when connected to the process. After that, approximately two days were devoted to checking and tuning performance. This involved experiments with different sampling rates. A significant part of the installation time also involved other parts of the system, particularly the logic. Operational experiences with the adaptive control system have been very favorable. The variation in elongation was better than with a conventional system, and the adaptive system also settled faster during mode switches. The system has been in continuous operation since 1983.

12.5 A Ship Steering Autopilot

A conventional autopilot for ship steering is based on the PID algorithm. Such a regulator has manual adjustment of the parameters of the PID regulator and often also a dead zone called weather adjust a simple
version of a performance-related knob. The reason manual adjustments are necessary is that the dynamics of a ship vary with speed, trim, and loading. It is also useful to change the autopilot settings when disturbances in terms of wind, waves, currents, and water depth are changed. Adjustment of an autopilot is a burden on the crew. A poor adjustment results in unnecessarily high fuel consumption. It is therefore of interest to have adaptive autopilots. A ship steering autopilot, Steermaster 2000 from Kockum Sonics AB in Sweden will be described in this section.

Ship Steering Dynamics

Simple ship steering dynamics were presented in connection with the discussion of gain scheduling in Section 9.5. That section detailed how the dynamics vary with the velocity of the ship and showed how the variations could be reduced by gain scheduling. It has been shown by hydrodynamic theory that the average increase in drag due to yawing and rudder motions can be approximately described by

$$\frac{\Delta R}{R} = k (\bar{\psi}^2 + \lambda \delta^2)$$  \hspace{1cm} (12.2)

where $R$ is the drag and $\psi^2$ and $\delta^2$ denote the mean square of heading error and rudder angle amplitude, respectively. The parameters $k$ and
\( \lambda \) will depend on the ship and its operating conditions. The following numerical values are typical for a tanker:

\[
k = 0.014 \text{ deg}^2 \quad \lambda = 1.12
\]

It is thus natural to use the criterion

\[
V = \frac{1}{T} \int_0^T \left( (\psi(t) - \psi_{\text{ref}})^2 + \lambda \delta^2(t) \right) dt \tag{12.3}
\]

as a basis for the design and evaluation of autopilots for steady-state course keeping. The disturbances acting on the system are due to wind, waves, and currents. A detailed characterization of the disturbances and their effect on the ship’s motion is difficult. In a linearized model, disturbances appear as additive terms. It is common practice to describe them as random signals; the waves have a narrow band spectrum, as illustrated in Example 2.6. The center frequency and the amplitude may vary significantly.

**Autopilot Design**

An autopilot has two main tasks: steady-state course keeping and turning. Minimization of drag induced by the steering is the important factor in course keeping, and steering precision is the important factor when turning. It is therefore natural to have a dual mode operation. These two modes will be described below, together with the basic autopilot functions.

The influence of variations in the ship’s speed is handled by gain scheduling. The other disturbances are taken care of by feedback and adaptation. Implementation of the gain scheduling is discussed in Section 9.5. It requires a measurement of the forward velocity of the ship. A block diagram of the autopilot is shown in Fig. 12.12. If disturbances are regarded as stochastic processes, steady-state course keeping can be
described as a linear quadratic gaussian problem. It is then natural to estimate an ARMAX model (Eq. 3.35). The particular process model used is

\[
\Delta \psi(t) - a \Delta \psi(t - h) = b_1 \delta(t - h) + b_2 \delta(t - 2h) + b_3 \delta(t - 3h) \\
+ e(t) + c_1 e(t - h) + c_2 e(t - 2h)
\] (12.4)

This model is built on Nomoto’s approximation (compare Section 9.5). The additional \( b \) term was introduced to allow additional dynamics to be captured as an increased time delay. The difference occurs because there is a pure integration in the model from rate of turn to heading angle. A control law that minimizes the criterion of Eq. (12.3) is then computed using the certainty equivalence principle. This approach requires the solution of a Riccati equation, which can be done analytically in the particular case. A straightforward minimum-variance control law was used in some early experiments. This was replaced by the LQG control law described above, because there were significant advantages at short sampling intervals, which could not be used with the minimum-variance control law. The sampling interval in the model is set during commissioning.

**Turning Regulator**

The major concern when turning is to keep tight control of the motion of the ship even at the expense of rudder motions. For high turning rates the dynamics of many ships are nonlinear. This is illustrated in Fig. 12.13, which shows the steady-state relation between turning rate and rudder angle. The normal course-keeping regulator can handle small
Figure 12.14  The operator’s panel for the Steermaster autopilot.

changes in heading, but it cannot handle large maneuvers because of the nonlinearity discussed above. A special turning regulator was therefore designed. The regulator is a high-gain regulator in which the feedback is of PID type. (Compare Fig. 1.2.) Appropriate PID parameters are determined during commissioning. The model used is nonlinear. It is designed so that the command signal is turning radius. The turning rate is thus \( r = u/R \), where \( u \) is the ship’s speed and \( R \) the turning radius.

**Man-machine Interface**

The fact that turning radius is used as a command signal instead of turning rate simplifies maneuvering considerably, because it is easy to determine an appropriate turning radius from the chart. It also improves path following, since the ship’s speed may change during a turn. This is then compensated for automatically. The man-machine interface is very simple (see Fig. 12.14). There is one joystick to increase and decrease the heading. An optional joystick provides override control; whenever this is moved, it gives direct control of the rudder angle. Control can be transferred to the autopilot by a reset button. When making a turn, the desired turning radius is set by increase-decrease buttons. The turn is initiated when the joystick is moved to the new desired course. The turn is then executed, and the ship turns until the desired course is reached. The fixed-gain regulator is used during the turn, and the adaptive course-keeping regulator is initiated when the turn is complete.

There are no adjustments on the course-keeping regulator; everything is handled adaptively. Some default values are set during commissioning,
but the fixed-gain regulator can be activated when operator pushes a switch labeled fixed control. This is typically used when there are heavy waves coming from behind, so-called quartering sea. This condition makes steering difficult, because the effective rudder forces are small and the disturbing wave forces large.

Operational Experiences

Early versions of the autopilot were field-tested in 1974, and the product was announced in 1979. There were 64 systems installed as of December 1987. The product is used in various kinds of ships. One installation, in a ferry that navigates between Stockholm and Helsinki has been in continuous operation since 1980. It uses adaptive control all the time. The ability to cope with large variations in speed has been found very useful, and the turning radius feature is particularly useful for navigation in archipelagos, where a lot of maneuvering is necessary. Figure 1.10 indicates the improvements in course-keeping that can be obtained through adaptation. The decreased drag with the data shown in the figure corresponds to a reduction in fuel consumption of 2.7%.

12.6 Ultrafiltration

Patients with little or no renal function need some form of artificial blood purification to stay alive. In dialysis the blood is cleansed of waste products and excess water and the electrolytes in the blood are also normalized. There are more than 350,000 patients all over the world taking this treatment a couple of times a week. In its most common form, hemodialysis, the blood flows past a semipermeable membrane with a suitably composed dialysis fluid on the other side. Because of the large number of different dialyzers available on the market, the control algorithm in the dialysis machine must be able to handle a wide span in process gain and other process characteristics.

An adaptive pole placement controller has been used in the Fluid Control Monitor (FCM), recently introduced by Gambro AB in Lund, Sweden. It has performed very well and is probably one of the most widely used adaptive controllers in the world today. More than 4,000 have been delivered since the introduction of the product in late 1986. This section will describe the system.

Process Description

A schematic view of the Gambro AK-10 dialysis system is shown in Fig. 12.15. Only the parts relevant to flow and pressure control are shown
in detail. Clean water is heated to around 37 C, and salt is added to physiological concentration. A pressure drop in the restrictor is created by the first pump, to degas the solution. The restrictor and the first pump \((P_1)\) determine the flow into the dialyzer. Due to the compressibility of the air in the bubble chamber, flow changes to the dialyzer will be slowed down by a time constant.

After passing a few measuring devices and a valve, the fluid leaves the Dialysis Fluid Monitor (DFM) and passes the first measuring channel of the FCM before entering the dialyzer. Before returning to the DFM, the second measuring channel of the FCM is passed. In the DFM a few more measuring devices and valves are passed before the second pump \((P_2)\). A restrictor is placed on the outlet in order to allow positive pressures in the dialyzer.

To maintain a specified transmembrane pressure the DFM has a control system, that is based on a conventional fixed-gain digital PI regulator. This regulator has a sampling period of 0.16 s and an integration time of about 30 s. See the block diagram in Fig. 12.16. The purpose of the fluid control module is to control the weight loss during the treatment. This is done by the external control loop shown in Fig. 12.16 which has the flow difference \(Q_f\) as the measured variable and the setpoint to the pressure controller as the control variable.

**Process Dynamics**

The dialyzer dynamics can be approximately described by the model

\[
C \frac{dp}{dt} = Q_f - Bp
\]
where $p$ is the transmembrane pressure and $Q_f$ the net fluid flow from the dialyzer. The constant $C$ is the compliance. Parameter $B$, which represents the static gain, may for example vary from $1.6 \cdot 10^{-12}$ to $120 \cdot 10^{-12}$ (m$^3$s$^{-1}$Pa$^{-1}$), i.e. a gain variation by a factor of 75.

The complete dynamics of the pressure loop can be approximately described as a second-order transfer function. It has one pole associated with the dynamics of the ultrafiltration and another associated with the pressure control system. The PI regulator is tuned conservatively so that both poles are real. The dominating time constant is 30–50 s. The transfer function from the pressure setpoint to the flow $Q_f$ then also has the same poles, but it also has a zero corresponding to the pole $s = -B/C$ of the ultrafiltration (see Fig. 12.16). This zero can change significantly with the type of dialysis filter used. A consequence is that there is a drastic difference in the dynamics obtained for different filters. This is illustrated in Fig. 12.17, which shows responses in differential flow to step changes in the pressure setpoint. Figure 12.17(a) shows responses for a fiber membrane and Fig. 12.17(b) for a plate membrane. Notice the drastic difference in the static gains and the noise levels.

The main function of the system is to control the total water removal $V$ during the treatment. The water removal is given by

$$\frac{dV}{dt} = Q_f$$

(12.5)

An Earlier Control System

An earlier system used a PI regulator in the outer loop. Because of the large gain variations it was necessary to use a conservative setting with
low gain. This resulted in very sluggish control of the weight loss. Experiments with various simple forms of gain adjustment did not solve the problem.

**Adaptive Control**

The adaptive regulator was designed as an indirect adaptive pole placement algorithm.

**Parameter Estimation.** The dynamics can be expected to be of third order, representing the dynamics of the pressure loop and the dynamics of the filter introduced to filter the flow signal. This filter has a time constant of about 30 s. Experiments with system identification indicated however, that data could be fitted adequately by

\[
Q_f(t) = aQ_f(t - h) + b_1p_c(t - h) + b_2p_c(t - 2h) \quad (12.6)
\]

where \(Q_f\) is the filtration flow and \(p_c\) the setpoint of the pressure loop. This model represents first-order dynamics with a time delay. A sampling interval of 5 s was found suitable. The parameter estimation was made on differences to avoid problems with a constant level in the signals.

The estimated steady-state gain is an important parameter. With a low estimated gain, the gain in the controller will be large. It is therefore advantageous to have the sum of the \(b\) parameters as one of the
estimated parameters, so that it is easy to set a lower limit to the estimated gain. This has been done in the FCM by using the regression vector \([Q_f(t) \ p_c(t) \ p_c(t - p_c(t - h))]\) instead of \([Q_f(t) \ p_c(t) \ p_c(t - h)]\). If the estimated gain becomes too small, the estimate is stopped at the limit.

A constant forgetting factor of 0.999 is used to track slowly time-varying parameters. To improve numerics, only the diagonal elements of the covariance matrix \(P\) are divided by this factor. It is well known that the equation for \(P(t)\) may be sensitive to numerical precision when a forgetting factor is used. This is due to the fact that the eigenvalues of the \(P\) matrix may be widely separated. Several methods to handle this problem were described in Chapter 11 and in the discussion of Novatune and Firstloop in this chapter. In this case the problem was avoided by careful scaling, and an ordinary recursive least-squares method could be used.

*Control Design.* A conventional pole placement algorithm and a design that guarantees integral action were used. (See Section 11.4.) Several factors influence the choice of desired closed-loop poles. If a smooth control is desired in the steady state, the speed of setpoint changes should not be set too high. Secondly, the first step response at start-up must not be quicker than the time required to get a reasonable model. A reasonable response time in accumulated flow is one hour. The other closed-loop poles, which correspond to flow changes, were specified by time constants 25 s, 15 s, and 15 s.

The regulator can be reparameterized to correspond to a PID controller with a filtered derivative part. The structure was chosen so that the regulator corresponds to a discrete-time PID regulator in which the reference signal only enters the \(P\) and \(I\) parts. This corresponds to \(b = 1\) in Eq. (8.3) in Chapter 8. A possible common factor in the estimated model was canceled before entering the design calculations.

**Special Design Considerations**

Control of fluid removal during dialysis has a direct influence on the wellbeing of the patient. This imposes heavy demands on the control system. Several safety features have been included. Smooth performance from the first moment of control is essential. This can be achieved by a careful choice of certain parameters, as discussed below.

*Filtering.* The measured flow signal is corrupted by measurement noise. Since a new value is available every second, it is possible to filter the signal. With a sampling period for control of 5 s, it was found suitable to use a first-order filter with a time constant 30 s to filter flow and accumulated flow before using the values in the control algorithm. This
smooths the control signal considerably without preventing fairly quick setpoint changes.

*Limits on Setpoint Changes.* Both the absolute level and the rates of setpoint changes were limited based on physical constraints. The PID regulator was provided with conventional anti-windup protection to avoid problems with saturation. Parameter updating is also interrupted when the pressure setpoint is kept constant at a limit. At start-up, when the model parameters may be far from their best values, it is also wise to prevent the control algorithm from changing the control signal (i.e., the pressure setpoint) too rapidly. The rate limit on the pressure setpoint prevents this; experience has shown that this limit is only hit rarely.

*Start-up.* A critical moment for an adaptive controller is the start, before the model parameters have been accurately estimated. It was required that its step response should be almost perfect from the beginning. For this reason most of the development time was spent in adjusting the parameters to assure a smooth start. The following parameters were then found to be important:

- Initial values of the parameter estimates
- Initial values of the covariance matrix $P$
- The desired closed-loop poles
- The time allowed for signals to settle before estimation and control starts
- Limits on the estimated parameters (especially the static gain)
- The limit on control changes (and control)

The initial values of the parameter estimates are important, since they determine the initial controller parameters. They were chosen to model a high-gain dialyzer, with an extra time delay, in order to give a cautious low-gain controller. This is perfect for a highly permeable (i.e., high-gain) membrane, but for normal membranes the pressure changes will be too small, which is soon detected by the parameter estimator.

It is important to choose the $P$ matrix carefully. This determines the speed of parameter estimation. Too large $P$'s will make the estimates noisy, and there is a risk that the estimates may temporarily give bad controllers. Also, a too large $P$ can quickly eliminate the carefully chosen initial parameters in the estimator. With too small $P$'s the time needed to find a good model can be very long, which is not at all acceptable.

It was found advantageous to introduce a lower bound on the estimated gain in the model. With low-gain dialyzers there would otherwise be a tendency for the estimator to decrease the gain estimate too much, so that the controller gain would be too high for a while. A suitable limit for the model gain could be determined from the known data of existing
dialyzers. To facilitate the checking of the estimated gain a special form of the process model was used. The estimated pole was also bounded away from a pure integrator, since this pole enters the expression for the gain limit.

The limit on the setpoint changes also helps to assure a smooth start-up. The desired closed-loop poles are important parameters. The equivalent time constants should be chosen sufficiently long to give the estimator time to find a good model before the setpoint is approached for the first time. They should also be as short as possible, in order to give a rapid response to setpoint changes. They were chosen to 714, 5, 3, and 3 sample intervals, which correspond to 1 hour, 25 s, and 15 s. Without the requirement of a smooth start-up it would have been possible to speed up the desired dynamics considerably. However, setpoint changes are not very frequent, and smooth start-up is much more important than rapid setpoint changes.

If by chance the desired pressure is already set at start-up, there would be no pressure change that would help to improve the estimates of model parameters. Therefore, there is a period of forced small pressure changes for the first eight minutes after a reset. This is accomplished by periodic changes of the setpoint every 45 s.

With an adaptive controller it is very important to assure that the estimated model is never destroyed. The estimator should therefore always be given true values for control and measured signals. If for some reason such as an alarm situation causing the DFM to bypass the dialysis fluid the control signal is not allowed to do its job, the estimator must be turned off. The controller will then use the old estimates for a while.

After all such breaks, and also at start-up, a settling period is allowed during which correct signals are entered into all the vectors but no estimation is done. This settling period is very important, especially at start-up, when the estimates are most sensitive to changes in the signals. Errors in the signals also force the $P$ matrix to decrease rapidly, so that future learning is slowed down considerably.

Alarms. Appropriate alarms are an important part of any useful control system. An alarm indicates if the volume control error is too large and also if something is wrong in the dialysis fluid monitor or with the pipes. If there is a stop in the blood pipe from the dialyzer to the drip chamber, the blood pressure within the dialyzer will rise thus causing a large ultrafiltration rate and minimized pressure. The alarm in the FCM will then cause the DFM to enter a patient-safe condition.

Operational Experience

It has been possible to use the algorithm to handle ultrafiltration control for all kinds of dialyzers available today. Treatment modes such as single-
needle or double-needle treatment or sequential dialysis with periods of isolated ultrafiltration have been tested. Dialyzers with variations in values of $B$ from 1 to 70 have been tested in the laboratory without any problems occurring. After a period of approximately five months of clinical trials at several clinics, full-scale production started in the autumn of 1986. Over 4,000 units have been delivered as of February 1988. Since every machine may be used in several hundred treatments each year there is now extensive practical experience with this algorithm which seems to work well under all kinds of conditions.

12.7 Conclusions

In this chapter we have tried to give an idea of how adaptive techniques are used in real control systems. A few general observations can be made. Although there were more than 70,000 adaptive loops in operation in 1988, it is clear that adaptive control is not a mature technology. The techniques were introduced in products in the beginning of the 1980s. Those in use today are mostly first-generation products; there are second-generation products only in a few cases.

The description of the products and the real applications show clearly that although the key principles are straightforward, many “fixes” must be done in order to make the system work well under all possible operating conditions. The need for safety nets, safety jackets, or supervision logic is not specific to adaptive control. Similar precautions must be taken in all real control systems, but since adaptive control systems are complex to start with the safety nets required can be quite elaborate.

The examples clearly show that adaptive systems are not black box solutions that are a panacea. Rather, adaptive methods are useful in combination with other control design methods. Both in the rolling mill example and in the ship steering autopilot, adaptation was combined with gain scheduling. Another example is the use of a feedforward signal in the pulp dryer to improve the adaptation transient.

A third observation is that the man-machine interface is very important. A fourth observation is that some operating conditions are not conveniently handled by adaptive control. One example is the behavior of ship steering autopilots in quartering sea.

There are unquestionably many different adaptive techniques but so far only a few of them have been used in industrial products. In many cases the choices have not been made by comparing several alternatives; one method has been chosen quite arbitrarily. This means that many alternatives have not been tried.

The computing power available has a significant influence on the type
of control algorithms that can conveniently be implemented. The simple auto-tuners use simple 8-bit microprocessors, while some of the more advanced systems use full 32-bit architecture. In most process control applications there are no problems with computing time. The rolling mill applications on the other hand, are quite demanding. The computing power available also has a significant impact on what man-machine interface can be implemented.

The applications also indicate the importance of the safety network. It is of interest to see the facilities provided in the toolbox and the specific solutions used in the dedicated systems. It is clearly much simpler to design a safety network for a dedicated system, where good parameter bounds can be established.

References

A number of applications are described in the books:


and in the survey papers:


Proceedings of the IFAC, CDC, and ACC are also good reference sources. More details about the products are available in manuals, brochures, and application notes from the manufacturers. The Leeds & Northrup Electromax V is described in:


*Preprints AIChe Anaheim Symposium*, (Session 31 Software for Advanced Control).

The Foxboro EXACT is described in:


The relay auto-tuning is described in:


The Asea Novatune/Master Piece is described in:


The description of the rolling mill example is based on:


The ship steering example is based on:


The particular control algorithm used in the product is described in:


The description of the control system for ultrafiltration is based on:

Chapter 13

PERSPECTIVES ON
ADAPTIVE CONTROL

13.1 Introduction

In this final chapter we attempt to give some perspective on the field of adaptive control. This is important but difficult, because the field is in rapid development. The starting point is a short discussion of some closely related areas that are not covered in the book. This includes adaptive signal processing in Section 13.2 and extremum control in Section 13.3. Particular attention is given to the field of adaptive signal processing, in which a cross-fertilization with adaptive control appears particularly natural. Adaptive regulators and auto-tuning have complimentary properties. Auto-tuners require very little prior information and give a robust ballpark estimate of gross system properties. Adaptive regulators require more prior knowledge, but they can give systems with much improved performance. It thus seems natural to combine auto-tuning with
adaptive control, in systems that combine several algorithms. Apart from algorithms for control, estimation, and design, it may also be useful to include supervision. It seems logical to use an expert system to monitor and control the operation of such a system. Systems of this type have been called expert control systems and are briefly discussed in Section 13.4. The use of expert systems also provides a natural way to separate algorithms from logic that occurs in all control systems. Adaptation is related to learning; in Section 13.5 we discuss some early uses of learning in control systems and how it is related to adaptive control as we now understand it. Finally, in Section 13.6 we attempt to speculate on future directions in the theory and practice of adaptive control.

13.2 Adaptive Signal Processing

Automatic control and signal processing have strong similarities; similar mathematical models and techniques are used in both fields. However, there are also some significant differences. The time scales can be different. Signal processing often deals with rapidly varying signals, as in acoustics, where sampling rates of tens of kHz are needed. In control applications it is often (but not always) possible to work with much slower sampling rates.

A more significant difference is that time delays play a minor role in signal processing. It is often permissible to delay a signal without any noticeable difficulty. Since control systems deal with feedback, even small time delays can result in drastic deterioration in performance. A third difference is in the industrial markets for the technologies. In signal processing there are some standard problems that have a mass market, such as in the field of telecommunications. The control market is more diversified and fragmented. Adaptive control is used to design control systems that work well in an unknown or changing environment. The environment is represented by process dynamics and disturbance signals. Adaptive signal processing is used to process signals whose characteristics are unknown or changing. The development of adaptive control and adaptive signal processing have strong similarities, although there are some differences. More emphasis is given in signal processing to fast algorithms. Although there have been attempts to bring the fields closer together, much more effort is needed in this direction. To illustrate this we will describe a few typical adaptive signal processing problems.

Prediction, Filtering, and Smoothing

Prediction, filtering, and smoothing are typical signal processing problems, which can all be described as follows. Given two signals $x$ and $y$ and a
filter $F$, determine the filter such that the signals $y$ and $\hat{y} = Fx$ are as close as possible. The problem can be illustrated by the block diagram in Fig. 13.1. In a typical case we have

$$x(t) = s(t) + n(t) \quad \text{and} \quad y(t) = s(t + \tau)$$

where $s$ is the signal of interest and $n$ is some undesirable disturbance. The problem is called smoothing if $\tau < 0$, filtering if $\tau = 0$, and prediction if $\tau > 0$. Solutions to such problems are well known for signals with known spectra and quadratic criteria. The corresponding adaptive problems are obtained when the signal properties are not known. All recursive parameter estimation methods can be applied to the adaptive signal processing problems. This is illustrated in Fig. 13.2, which gives a typical adaptive solution. The adjustment mechanism can be any recursive parameter estimator. The details depend on the structure of the filter and the particular estimation method chosen. An example illustrates the idea.

Example 13.1—Output error parameter estimation
Assume that the filter is represented as an ordinary pulse transfer function

$$F(z) = \frac{b_0 z^{n-1} + b_1 z^{n-2} + \cdots + b_{n-1}}{z^n + a_1 z^{n-1} + \cdots + a_n}$$

To obtain a recursive estimator the parameter vector

$$\theta = [a_1 \ldots a_n \ b_0 \ldots b_{n-1}]$$

Figure 13.2 a) An adaptive system for filtering, prediction or smoothing and (b) its simplified representation.
and the regression vector
\[ \varphi(t) = \begin{bmatrix} -\hat{y}(t-1) & \ldots & -\hat{y}(t-n) & x(t-1) & \ldots & x(t-n) \end{bmatrix} \]
are introduced. The error is then given by
\[ \varepsilon(t) = y(t) - \hat{y}(t) = y(t) - \varphi^T(t-1)\hat{\theta}(t-1) \]
and the equation for updating the estimate is
\[ \theta(t) = \theta(t-1) + P(t)\varphi(t-1)\varepsilon(t) \]

The special case of Example 13.1, obtained when the filter is an FIR filter and a gradient parameter estimation scheme is used, is particularly simple. This is the LMS algorithm.

**Block Diagram Representation**

The block diagram in Fig. 13.2 represents a solution to a generic signal processing problem. To make it easy to build large systems, it is convenient to consider this module as a building block that can be used for many different purposes. This is simpler if a proper representation is used. For that purpose it is convenient to represent the module as a block, which receives signals \(x\) and \(y\) and delivers estimates \(\hat{y}\) and \(\hat{\theta}\). Such a representation, shown in Fig. 13.2(b), makes it possible to describe several adaptive signal processing problems.

**Adaptive Noise Cancellation**

Consider the situation with a mobile telephone in a car where there is a considerable ambient noise. Assume that two microphones are used. One is directional and picks up the driver’s voice corrupted by noise; the other is directed away from the driver and picks up mostly the ambient noise. By connecting the microphones to an adaptive filter as shown in Fig. 13.3 it is possible to obtain a signal that is considerably improved. Removal of
power frequency hum from measurement signals is another application at adaptive noise cancellation.

Adaptive Differential Pulse Code Modulation (ADPCM)

Digital signal transmission is becoming important because of the rapid development of new hardware. Its use in ordinary telephone communication is increasing. Pulse code modulation (PCM) is the standard method for converting analog signals to digital form. The analog signal is filtered and digitized using an analog to digital (A-D) converter. The digitized signal is then transmitted in serial form. If the A-D converter has $B$ bits and the sampling is $f$ Hz, the transmission rate required is $fB$ bits/s. For standard voice signals a sampling rate of 8 kHz is typically used. A resolution of 12 bits in the A-D converter is required to get good-quality transmission. The bit rate required is thus 96 kbit/s. By having an A-D converter with a nonlinear characteristic it is possible to reduce the bit rate to 64 kbit/s, which is the standard for digital voice transmission.

It is highly desirable to reduce the transmission rate, because more communication channels are then obtained with the same transmission equipment. The bit rate can be reduced significantly by using differential pulse code modulation (DPCM). In this technique, the signal to be transmitted is first filtered with a predictive filter. The innovations of the signal are then computed as $\varepsilon = y - \hat{y}$, and only the innovations are transmitted (see Fig. 13.4). The receiver has a prediction filter with the same characteristics as the filter in the sender. The signal $\hat{y}$ can then be reconstructed in the receiver. The bit rate required is reduced significantly because fewer bits are required to represent the residual. For voice signals it has been shown that a resolution at 4 bits is sufficient. This means that the bit rate required for the transmission can be reduced from 64 kbit to 32 kbit.

The prediction filter depends on the character of the transmitted signal. Substantial research into the characterization of speech has shown that it can be well predicted by linear filters. However, the properties of the filter will change with the particular sound that is spoken. To predict speech well it is thus necessary to make the filters adaptive. The
transmission scheme obtained is then called adaptive differential pulse code modulation (ADPCM). Such a scheme which uses an adaptive filter based on the output error method is shown in Fig. 13.5. Notice that the adaptive filters at the transmitter and the receiver are driven by the residual only. If the filters in the receiver and the transmitter are identical, the filter parameters will automatically be the same. The adaptive filters have therefore been standardized by CCITT (Comité Consultatif Internationale de Télégraphique et Téléphonique). The filter used has the transfer function

\[ H(z) = \frac{b_0 z^5 + b_1 z^4 + \cdots + b_5}{z^4 (z^2 + a_1 z + a_2)} \]

The regression vector associated with the output error estimation is

\[ \varphi(t) = [-x(t) \quad -x(t-1) \quad e(t) \cdots e(t-5)] \]

and the associated parameter vector is

\[ \theta = [a_1 \quad a_2 \quad b_0 \ldots b_5] \]

The standard least-squares estimator is of the form

\[ \theta(t+1) = \theta(t) + P(t+1) \varphi(t) e(t+1) \]

Several drastic modifications are made to simplify the calculations. A constant value of the gain is used. The multiplication is avoided by just using the signs of the signals. Leakage is also added to make sure that the estimator is stable. The updating of the parameters \( b_i \) is then given by the sign-sign algorithm

\[ \hat{b}_i(t) = (1 - 2^{-8}) \hat{b}_i(t) + 2^{-7} \text{sgn} (e(t-i)) \text{sgn} (e(t)) \]  

\[(13.1)\]
Similar approximations are made in the other equations. The computations in Eq. (13.1) are very simple. They can be done by shifts and addition of a few bits, which can be accomplished with a small VLSI circuit. The CCITT ADCPM standard was achieved after significant experimentation. It is a good example of how drastic simplifications can be made with good engineering.

13.3 Extremum Control

The control strategies that have been discussed in the book have mainly been such that the reference value is assumed to be given. The reference value is often easily determined. It can be the desired altitude of an airplane, the desired concentration of a product, or the thickness at the output of a rolling mill. On other occasions it can be more difficult to find the suitable reference value or the best operating point of a process. For instance, the fuel consumption of a car depends, among other things on the ignition angle. The mileage of the car can be improved by a proper adjustment, but the efficiency will depend on such conditions as the condition of the road and the load of the car. To maintain the optimal efficiency it is necessary to change the ignition angle.

To track a varying maximum or minimum is called extremum control. The static response curve relating the inputs and the outputs in an extremum control system is nonlinear. The task of the controller is to find the optimum operating point and to track it if it is varying. Several processes have this kind of behavior. Control of the air-fuel ratio of combustion is one example. The optimum will change, for instance, with temperature and fuel quality. Another example is water turbines of the Kaplan type in which the blade angle of the turbine is changed to give maximum output power. The same problem is encountered in windmills, in which pitch angle is changed depending on wind speed.

Extremum control is related to optimization techniques; many of the ideas have been transferred from numerical optimization. There was a great interest in extremum control in the 1950s and 1960s, and some commercial products were put on the market. For instance, the first computer control systems installed in the process industry were motivated by the possibility of making optimization of the setpoints of the controllers. The interest then declined, partly due to the difficulty of implementing the optimizing controllers. Furthermore, there is great difficulty in finding appropriate process models. The development of the computers has lead to a renewed interest in extremum control and its combination with adaptive control. Improved efficiency of the process can result in large savings in the cost of energy and raw material.
Figure 13.6 A simplified block diagram of an extremum control system.

Figure 13.6 shows a simplified block diagram of an extremum control system. The process can work in open loop or in closed loop, as in the figure. The most important feature is that the process is assumed to be nonlinear, in the sense that at least the performance is a nonlinear function of the reference signal. The goal of the search algorithm is to keep the output as close as possible to the extremum despite changes in the process or the influence of disturbances. The output used in the search algorithm is some measurement of the performance of the system, for instance, efficiency. The conventional regulator can also use this signal, but it is more common that the regulator uses some other output of the process.

Models

Extremum control systems are by necessity nonlinear. How the processes are modeled is therefore all-important. In many investigations of extremum control systems it is assumed that the systems are static. This assumption can be justified if the time between the changes in the reference value is sufficiently long. For static systems it is possible to use many of the methods from numerical optimization. A typical description of the process is

$$y(t) = f(u(t), \theta, t)$$

(13.2)

where $f$ is a nonlinear function and $\theta$ is a vector of unknown parameters that may change with time.

If there are dynamics in the process, the performance may not have settled at a new steady-state value before the next measurement is taken. This will give an interaction in the control system that can be difficult to handle. The dynamic influence will increase the complexity considerably.

In many applications it is not easy to find the appropriate models and to determine the exact nature of the nonlinearities involved. It can therefore be appropriate to combine adaptivity and extremum control. One way to simplify the identification of an unknown nonlinear model is
to assume that the process can be divided into one nonlinear static part and one linear dynamic part. Models with different properties are obtained if the nonlinearity proceeds or follows the linear part. The complexity of the problem will also depend on which of the variables in the process can be measured. One special type of model that has been used in extremum control systems is the \textit{Hammerstein models}. A typical discrete-time model of this type is

\begin{equation}
A(q)y(t) = B(q)f(u(t))
\end{equation}

where \( f \) is a nonlinear function, typically a polynomial.

The main effect of an input nonlinearity is that it restricts the possible input values for the linear part. The nonlinear control problem can then be treated as a linear control problem with input constraints. The case with the output nonlinearity perhaps leads to more realistic problems but is also more difficult to solve.

\textbf{Extremum Control of Static Systems}

The first extremum control systems were based on analog implementation. One way to make the optimization is the so-called perturbation method. The basic idea is to add a known time-varying signal to the input of the nonlinearity, then observe the effect on the output, and make a correlation between these two signals. Depending on the phase between the two signals, it can be determined in which direction the extremum is. The perturbation method has been used for extremum control of chemical reactors, combustion engines, and gas furnaces, for instance.

Extremum control of static systems as in Eq. (13.2) is in essence a problem of numerical optimization. With the analog implementations, the possible methods were severely restricted. When a digital computer is available, standard algorithms for function minimization can be used. Usually it is only possible to measure the function values, not its derivative. The function minimization then has to be done using numerically computed derivatives. Some methods only use function comparisons. These methods can be used even for minimization of nonsmooth functions.

Performance measurements are typically corrupted by noise. It is then necessary to average out the influence of the noise. This implies that the gain in the optimization algorithm should go to zero. However if the extremum is changing with time, the gain should not go to zero. This is the same compromise as discussed in connection with tuning and adaptive control.

Most schemes for extremum control of static systems do not build up any information about the nonlinearity. The "states" of the algorithms are essentially the current estimate of the optimum point and some previous measurements. By using a model and system identification, it is possible to
utilize the measurements of the system better and to follow time variations in the process.

**Extremum Control of Dynamic Systems**

If there are dynamics in the process, it is necessary to take this into consideration when doing the optimization. The correlation and interaction between different measurements of the performance will otherwise confuse the optimization routine. One possibility discussed above is to wait until the transients have vanished before the next change is made. Of course, this will increase the convergence time, especially if the process has long time constants. One way around the problem is to base the optimization on nonlinear dynamic models. An example is the Hammerstein model,

\[ A(q)y(t) = b_0 + B_1(q)u(t) + B_2(q)u^2(t) + C(q)e(t) \quad (13.4) \]

which corresponds to an input nonlinearity of second order. The main reason for the popularity of this model is not that it is a good picture of the reality, but rather that it is linear in the parameters. The parameters can be estimated, for example, using recursive least squares. The static response between the input and the output is given by

\[ A(1)y_0 = b_0 + B_1(1)u_0 + B_2(1)u_0^2 \]

The methods for static optimization discussed above can now be used. Also note that the gradients and the Hessian are easily computed, which will speed up the convergence.

**Conclusions**

The field of extremum control is far from mature. One crucial point is the modeling of the processes and the nonlinearities. It is generally very difficult to analyze nonlinear control problems and to derive optimal controllers, especially if there are stochastic disturbances acting on the system. The extremum control problem also has connections with the dual control problem discussed in Chapter 7. Extremum-seeking methods combined with adaptive control are of great practical interest, since even small improvements in the performance can lead to large savings in raw material and energy consumption. There are commercial extremum controllers.

**13.4 Expert Control**

All practical control systems contain heuristics. This appears as logic around the basic control algorithm. Adaptive systems have a lot of heuristics in the safety logic. Expert systems offer an interesting possibility of
structuring the logic in a control system. If a good way to handle heuristic logic is available, it is also possible to introduce more complex control systems that contain several different algorithms. For example, it is possible to combine auto-tuners and adaptive algorithms that have complementary properties. The auto-tuner requires little prior information; it is very robust and can generate good parameters for a simple control law. Adaptive regulators can be more complex, with potentially better performance. Since they are based on local gradient procedures, they can adjust the regulator parameters to give a closed-loop system with very good performance, provided that reasonably good a priori guesses of system order, sampling period, and parameters are given. The algorithms will not work if the prior guesses are too far off. With poor prior data they may even give unstable closed-loop systems. This has led to the development of the safety logic discussed in Chapters 11 and 12.

Expert Systems

One objective of expert systems is to develop computer-based models for problem solving that are different from physical modeling and parameter estimation. An expert system attempts to model the knowledge and procedures used by a human expert in solving problems within a well-defined domain. Knowledge representation is a key issue in expert systems. Many different approaches have been attempted, such as first-order predicate calculus (logic), procedural representations, semantic networks, production systems or rules, and frames. A knowledge-based expert system consists of a knowledge base, an inference engine and a user interface.

The Knowledge Base. The knowledge base consists of data and rules. The data can be separated into facts and goals. Examples of facts are statements such as “the system appears to be stable,” “PI control is adequate,” and “deviations are normal.” Typical examples of goals are “minimize the variations of the output,” “find out if gain scheduling is necessary,” and “find a scheduling table.” Data is introduced into the database by the user or via the real-time knowledge acquisition system. New facts can also be created by the rules. The rule base contains production rules of the type: “if premise then conclusion do action.” The premise represents facts or goals from the database. The conclusion can result in a new fact being added to the database or a modification of an existing fact. The action can be to activate an algorithm for diagnosis, control, or estimation. These actions are different from those found in conventional expert systems. The rule base is often structured in groups or knowledge sources that contain rules about the same subject. This simplifies the search. In the control application the rules represent knowledge about the control and estimation problem that are built into the system. This includes the appropriate characterization of the algorithms, judgemental knowledge on when to apply
them, and supervision and diagnosis of the system. The rules are introduced by the knowledge engineer via the knowledge acquisition system, which assists in writing and testing rules.

*Inference Engine.* The inference engine processes the rules to arrive at conclusions or to satisfy goals. It scans the rules according to a strategy, which decides from the context (current database of facts and goals) which production rules to select next. This can be done according to different strategies. In *forward chaining* it is attempted to find all conclusions from a given set of premises. This is typical for a data-driven operation. In *backward chaining* the rules are traced backward from a given goal to see if it can be supported by the current premises. This is typical for a diagnosis problem. The search can be organized in many different ways, depth-first or breadth-first. There are also strategies that use the complexity of the rules to decide the order in which they are searched. To devise efficient search procedures it is convenient to decompose the rule base into pieces that deal with related chunks of knowledge. If the rules are organized in that way, it is also possible for a system to focus its attention on a collection of rules in certain situations. This can make the search more efficient.

*User Interface.* The user interface can be divided into two parts. The first part is the development support that the system gives. This contains tools such as rule editor and rule browser for development of the system knowledge base. The other part is the run-time user interface. This contains explanation facilities that make it possible to question how a certain fact was concluded, why a certain estimation algorithm is executing, etc. It is also possible to trace the execution of the rules. The user interface can also contain facilities to deal with natural language.

**Expert Control**

The idea of expert control is to have a collection of algorithms for control, supervision, and adaptation that are orchestrated by an expert system. A block diagram of such a system is shown in Fig. 13.7. A comparison with Fig. 1.8 shows that the system is a natural extension of a self-tuning regulator. Instead of having one control algorithm and one estimation algorithm, the system has several algorithms. It also has algorithms for excitation and for diagnosis, as well as tables for storing data. Apart from this, the system also has an expert system, which decides when a particular algorithm should be used. The expert system contains knowledge about particular algorithms and the conditions in which they can be used.

In the special case in which there is only one algorithm of each category, Fig. 13.7 can be viewed as a well-structured way of implementing safety logic for an ordinary adaptive regulator. In that case the approach
Figure 13.7 A knowledge-based expert control system

has the advantage that it separates the safety logic from the control algorithms. Another advantage is that the knowledge is explicit and can be investigated via the user interface.

13.5 Learning Systems

The notion of learning systems has been developed in the fields of artificial intelligence, cybernetics, and biology. In its most ambitious form, learning systems attempt to describe or mimic human learning ability. This goal is still far away. The learning systems that have actually been implemented are simple systems that have strong relations to adaptive control. The systems have many names: neural nets, connectionist models, parallel distributed processing models etc.

Michie’s Boxes

This system grew out of early work on artificial intelligence (see Michie and Chambers, 1968) and attempts to balance an inverted pendulum (see Fig. 13.8). The system has four state variables, $\varphi$, $\dot{\varphi}$, $x$, and $\dot{x}$ which are quantized in a crude way, with five levels for the position variables $x$ and $\varphi$ and three levels for the velocity variables $\dot{x}$ and $\dot{\varphi}$. The state space can thus be described by 225 discrete states. The control variable is quantized into two levels: force left ($L$) or force right ($R$). The control law can be represented by a binary table with 225 entries. In the experiment the table was initialized with randomly chosen $L$’s and $R$’s in the table. A simple scoring method was used to update the table entries as a result of
13.5 Learning Systems 491

![Inverted Pendulum Diagram](image)

Figure 13.8 An inverted pendulum.

Experimental runs. Scoring was based on how long the pendulum stayed upright and the number of times the pendulum was in a discrete state. The system was able to balance the pendulum for about 25 minutes after a 60-hour training period. The table that defines the control action can be expressed in logic as:

*If cart is far left and cart is hardly moving and pendulum is hardly leaning and pendulum is swinging to right then apply force right.*

For this reason, the control law is also called *linguistic control*. When the logic is replaced by fuzzy logic, it is also called *fuzzy control*.

The training algorithm used in Michie's Boxes is similar to that used in programs for playing checkers and chess, but the pendulum problem is simpler than game playing. Training can be shortened by using a teacher; that is, to apply a scoring algorithm to an experiment in which the pendulum is balanced by an expert. A learning system of this type is obviously closely related to a model-reference adaptive system. The reference model can be viewed as a teacher.

**The Perceptron**

In a system like Michie's Boxes the control law is a logic function that gives the control action as a function of sensor patterns. The function is adaptive in the sense that it will adjust itself automatically. The *perceptron* proposed by Rosenblatt (1959) is one way to obtain a learning function. To describe the perceptron let $u_i, \ i = 1, 2 \ldots, n$ be inputs and $y_i, \ i = 1, 2, \ldots n$ be outputs. In the perceptron the output is formed as

$$y_i = f \left( \sum_{j=1}^{n} w_{ij} u_j - b \right) \quad i = 1, 2 \ldots m$$

(13.5)
where $w_{ij}$ are weights, $b$ a bias and $f$ a threshold function e.g.

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

To update the weights, the perceptron uses a very simple idea, which is called Hebb’s principle: Apply a given pattern to the inputs and clamp the outputs to the desired response, then increase the weights between nodes that are simultaneously excited.

This principle was formulated in Hebb (1949), in an attempt to model neuron networks. Mathematically it can be expressed as follows:

$$w_{ij} = w_{ij}(t) + \gamma u_i(t) (y^0_j - y_j)$$

(13.6)

where $y^0_j$ is the desired response and $y_j$ the response predicted by the model Eq. (13.5). By regarding the weights as parameters it becomes clear that the updating formula of Eq. (13.6) is identical to a gradient method for parameter estimation.

Widrow (1962) developed special-purpose hardware called the Adaline, to implement perceptron-like devices. The learning algorithm used by Widrow was based on a simple gradient algorithm like Eq. (13.6). In devices like the perceptron and the Adaline, learning is interpreted as adjusting the coefficients in a network. From this point of view it can equally well be claimed that an adaptive system like the MRAS or the STR is learning. The mechanisms for determining the parameters are also similar.

A drawback with the perceptron is that it can only recognize patterns that can be separated linearly. It fell in disfavor because of exaggerated claims, which could not be justified. It was heavily criticized in a book by Minsky and Papert (1969). The idea of designing learning networks did however, persist.

**The Boltzmann Machine**

The Boltzmann Machine may be viewed as a generalization of a perceptron. It was designed as a highly simplified model of a neural network. The machine consists of a collection of elements whose outputs are zero or one. The elements are linked by connections having different weights. The output of an element is determined by the outputs of the connecting elements and the weights of the interconnections. The firing is randomized in such a way that the probability of firing increases with the weighted sum of the inputs to an element. Some elements are connected to inputs, others to outputs, and there are also internal nodes. The connections in a Boltzmann Machine are assumed to be symmetric, which is a significant restriction.
In the perceptron there is a direct coupling between the inputs and the output. The Boltzmann Machine is much more complicated, because it can also have internal nodes. This implies that Hebb’s principle cannot be applied directly. An extension called back-propagation has been suggested in Rumelhart et al. (1986).

There are many variations of neural networks. Dynamics can be introduced in the nodes. Hopfield observed that the weights could be chosen so that the network would solve specific optimization problems (Hopfield and Tank, 1986).

**Hardware**

An interesting feature of the neural networks is that they operate in parallel and that they can be implemented in silicon. Using such circuits may be a new way to implement adaptive control systems. A particularly interesting feature is that it is easy to integrate the networks with sensors.

### 13.6 Future Trends

In this section we speculate on open research issues and the future of adaptive control. One interesting aspect of adaptive control is that it may be viewed as an automation of the modeling and design of control systems. To apply a technique automatically it is necessary to have a very clear understanding of the conditions under which it can be applied. Ideally this understanding should be formalized. Research on adaptive control will thus sharpen the understanding of control and parameter estimation.

**Industrial Impact**

Adaptive control technology is now starting to have a real impact on industry. There are several products on the market that use adaptive techniques. A few systems have been in continuous operation since 1976, and several hundred systems have been in continuous operation since 1983. Over 10,000 systems have been in operation since 1985. A large variety of techniques are used in the commercial systems.

A large number of PID regulators are already provided with automatic tuning or adaptation. A crude estimate is that about 70,000 high-quality regulators of this type are manufactured per year. About 20% of these are currently equipped with some form of adaptation. Almost all new systems announced have some method for adaptation or automatic tuning. Adaptive techniques are also introduced even in simple regulators for temperature control.

Adaptation is also being introduced in the large distributed system for process control. At least four manufacturers, Asea Brown Boveri, Bailey
Control  First Control Systems, and Foxboro, have adaptive control as a standard feature of their systems. Adaptive methods are also being used in special-purpose control systems.

The long-range viability of the PID algorithm is an interesting question. The algorithm is widely used and adequate for many purposes, and there are a large number of users who are very familiar with it. On the other hand, there are cases where other algorithms give superior performance. It is interesting to note that one manufacturer has decided to base its system on a general adaptive control algorithm that is automatically reduced to a PID if possible. We have no good guess what will happen in the future but prefer to let time tell.

In summary, many adaptive algorithms are well understood. Our insight into how adaptive methods can be used to engineer better control system is growing. Insight, understanding, and appropriate computing hardware are available. It seems likely that a large proportion of the control systems made in the future will have automatic tuning or adaptation. When adaptive control becomes more widely used, interesting phenomena that demand theoretical understanding will undoubtedly also be observed. For instance, what happens when many adaptive controllers are connected to one process? Will they interact? How should the system be initialized? We can thus look forward to interesting developments.

Algorithm Development

There are several important issues that relate to algorithm development. Current toolboxes for adaptive control use only a few of the algorithms that have been developed. It seems safe to guess that the toolboxes will be expanded, and it would also seem useful to include auto-tuners in the toolboxes to simplify initialization. Significant improvements can thus be achieved with tools that are already known but there is also a need for improved techniques. Better methods for control system design are needed. Techniques that can explicitly handle actuator constraints and model uncertainties would be valuable contributions. It would be very useful to have methods for estimating the unstructured uncertainties.

Diagnostic routines that will tell if a control algorithm is behaving as expected are needed. Such algorithms are well known for minimum-variance control, in which monitoring can be done simply by calculating covariances. It is straightforward to develop similar techniques for other design methods.

There is both theoretical and experimental evidence that probing signals are useful. It is also clear that it is not practical to introduce probing via stochastic control theory, because of the excessive computational requirements. A significant challenge is therefore to find other ways to introduce probing. There are many who intuitively object to introducing
probing signals intentionally. It must be remembered that a poorly tuned regulator will give larger than necessary deviations in controlled variables.

A systematic approach to design and implementation of safety networks is an issue of great practical relevance. Expert systems may be useful in this context.

**Multivariable Adaptive Control**

In this book we have focused on single input, single output systems, mainly to keep the presentation simple, but there has also been much research on multivariable adaptive control. Many of the results can be extended, but there is one difficulty. For single input, single output systems it is possible to find a good canonical form to represent the systems where the only parameter is the order of the system. For multivariable systems it is necessary also to know the Kronecker indices to obtain a canonical form. This is difficult both in theory and in practice. For special systems like those found in robotics a suitable structure can often be found using prior knowledge of the system.

Most adaptive control systems used so far are single-loop control. Coupled systems can be obtained by interconnection via the feedforward connection. Interesting phenomena can occur when such regulators are used on multivariable systems; analysis of the behavior of such systems is a fascinating problem.

**Theoretical Issues**

There are many unresolved theoretical problems in adaptive control. For example, we have no good results on the stability of schemes with gain scheduling. Much work is also needed on analysis of convergence rates. On a very fundamental level, there is a need for better averaging theorems. Many results apply only to periodic signals. This is natural, since the theory was originally developed for nonlinear oscillations. It would be highly desirable to have results for more general signal classes.

Several important problems have arisen in applications. The most important one is the design of proper safety logic; this is currently done in an ad hoc fashion. The development is also hampered by the fact that much of the information is proprietary, for competitive reasons.

**13.7 Conclusions**

In this book we have attempted to give our view of the complex field of adaptive control. There are many unresolved research issues and many white spots on the map of adaptive control. The field is developing rapidly, and new ideas are continually popping up. Our opinion is that adaptive
control is a good tool that a control engineer can use on many occasions. It is our hope that this book will help spread the use of adaptive control and that it may inspire some of you to do research that will enhance our understanding of adaptive systems.

References

There is an extensive literature on adaptive signal processing. A good treatment is given in:


There are strong international efforts by the IFAC and the IEEE to bridge the fields of adaptive control and adaptive signal processing. A student would be well advised to pay attention to this effort. Adaptive filters are discussed in:


The CCITT standard on adaptive differential pulse code modulation is described in:


Optimalizing control was introduced in the paper:


A good overview of extremum control problems is given in the survey paper:


An old but still relevant survey is also:

The notion of expert control was introduced in:


A detailed description of a system based on this idea is given in:


Good sources for knowledge about expert systems are:


The program Boxes is described in:


Fuzzy logic was introduced in:


Early examples of learning systems are given in:


The perceptron is described in:


A critique of it is given in:


Hebb’s principle for adjusting the weights in a neural network is described in:

Widrow’s Adaline is described in:


This system was applied to many problems, such as how to stabilize an inverted pendulum. Examples of neural networks and some of their uses are found in:


These books contain many references and a detailed treatment of the Boltzmann machine. A spectacular application of the Boltzmann machine is given in:


Hopfield’s network is described in:


How to implement neural networks in silicon and how to integrate them with sensors is found in the books:


Appendix A

REGULATOR DESIGN

Adaptive control may be viewed as a combination of automated modeling and design. It is essential to have precise statements of the problems of modeling and design. Many design techniques might be considered. To focus on the adaptive problem, we have concentrated on one simple class, namely pole placement and quadratic optimization. This appendix gives a brief summary of these methods. Such material is usually covered in introductory control courses and in courses on design and digital control. For additional material we refer to Åström and Wittenmark (1984).

Pole Placement Design

Pole placement is one of the simpler direct design procedures. The key idea is to find a feedback law such that the closed-loop poles have the
desired locations. Let the process to be controlled be described by

\[ Ay = Bu + v \]  \hspace{1cm} (A.1)

where \( u \) is the control variable, \( y \) the measured output, and \( v \) a disturbance. The symbols \( A \) and \( B \) denote polynomials in the differential operator \( p = \frac{d}{dt} \) for continuous-time systems or the forward shift operator \( q \) for discrete-time systems. It is assumed that \( A \) and \( B \) are relatively prime i.e., that they do not have any common factors. Further, it is assumed that \( A \) is monic, i.e., that the coefficient of the highest power in \( A \) is unity. The pole-excess \( d = \deg A - \deg B \) is for discrete-time systems the time delay of the process. Let the desired response from the reference signal \( u \) to the output be described by the dynamics

\[ A_m y_m = B_m u_c \]  \hspace{1cm} (A.2)

Furthermore, let \( A_o \) be the specified observer polynomial. The dynamics of the observer are not controllable from the reference input \( u_c \). To get a differentiation-free (continuous-time) or causal (discrete-time) controller, the model Eq. (A.2) must have the same or higher pole excess as the process of Eq. (A.1). This gives the condition

\[ \deg A_m - \deg B_m \geq \deg A - \deg B \]  \hspace{1cm} (A.3)

A general linear regulator can be described by

\[ Ru = T u_c - S y \]  \hspace{1cm} (A.4)

Elimination of \( u \) between Eqs. (A.1) and (A.4) gives

\[ y = \frac{BT}{AR + BS} u_c + \frac{R}{AR + BS} v \]

\[ u = \frac{AT}{AR + BS} u_c - \frac{S}{AR + BS} v \]

To achieve the desired input-output response, the following condition must hold

\[ \frac{BT}{AR + BS} = \frac{B_m}{A_m} \]  \hspace{1cm} (A.5)

The denominator \( AR + BS \) is the closed-loop characteristic polynomial. To carry out the design, the polynomial \( B \) is factored as

\[ B = B^+ B \]
where $B$ is a monic polynomial whose zeros are stable and so well damped that they can be canceled by the regulator. When $B^+ = 1$, there is no cancellation of any zeros. Since $B$ is canceled, it also factors the closed-loop characteristic polynomial. The other factors of this are $A_m$ and $A_o$. This gives the Diophantine equation

$$AR + BS = A_o A_m B^+$$

It follows from this equation that $B^+$ divides $R$. Hence

$$R = R_1 B^+$$

$$AR_1 + B^- S = A_o A_m$$

(A.6)

The solution of the Diophantine equation (Eq. A.6) is essentially the same as solving a set of linear equations. This is further discussed below. Equation (A.6) has a unique solution if $A$ and $B^-$ are relatively prime. Furthermore, it follows from Eq. (A.5) that $B^-$ must divide $B_m$ and that

$$T = A_o B_m / B^-$$

(A.7)

The pole placement design procedure can now be summarized in the following algorithm.

**Algorithm A.1—Pole placement design**

*Data:* Polynomials $A$, $B$.

*Specifications:* Polynomials $A_m$, $B_m$, and $A_o$.

*Compatibility Conditions:*

$B^-$ divides $B_m$

$$\deg A_m - \deg B_m \geq \deg A - \deg B$$

(A.8)

$$\deg A_o \geq 2 \deg A - \deg A_m - \deg B^+ - 1$$

(A.9)

*Step 1:* Factor $B$ as $B = B^+ B^-$.  

*Step 2:* Solve $R_1$ and $S$ from the equation $AR_1 + B^- S = A_o A_m$.  

*Step 3:* Form $R = R_1 B^+$ and $T = A_o B_m / B^-$.  

The control law is then given by

$$Ru = Tu_c - Sy$$

There are many variations of the pole placement procedure. A particularly simple case is when the whole $B$ polynomial is canceled. We then obtain the following algorithm.
Algorithm A.2—Pole placement design with all zeros canceled

Data: Polynomials $A$, $B$.

Specifications: Polynomials $A_m$, $B_m$, and $A_o$.

Compatibility Conditions:

$$\deg A_m - \deg B_m \geq \deg A - \deg B$$

$$\deg A_o \geq 2 \deg A - \deg A_m - \deg B - 1$$

Step 1: Form $B = b_0 B$.

Step 2: Solve $AR_1 + b_0 S = A_o A_m$.

Step 3: Form $R = R_1 B$ and $T = A_o B_m / b_0$.

The control law is

$$Ru = Tu_c - Sy$$

Notice that Step 2 is very simple in this case. $R_1$ is simply the quotient and $b_0 S$ the remainder when dividing $A_o A_m$ by $A$. This implies that the coefficients in the $R$ and $S$ polynomials can be obtained from a triangular linear system of equations. Another useful special case is when there are no cancellations at all. The design procedure then becomes that of the following algorithm.

Algorithm A.3—Pole placement design with no cancellations

Data: Polynomials $A$ and $B$.

Specifications: Polynomials $A_m$, $B_m$, and $A_o$.

Compatibility conditions:

$$B \text{ divides } B_m$$

$$\deg A_m - \deg B_m \geq \deg A - \deg B$$

$$\deg A_o \geq 2 \deg A - \deg A_m - 1$$

Step 1: Solve $AR + BS = A_o A_m$.

Step 2: Form $T = A_o B_m / B$.

The control law is

$$Ru = Tu_c - Sy$$

The Observer Polynomial

State feedback design combined with an observer gives a closed-loop system in which the observer dynamics are not controllable from the reference signal. This implies that in input-output form there is a pole-zero
cancellation. This is the reason for the notation observer dynamics for the extra dynamics $A_o$. The observer polynomial should be stable and fulfill the compatibility conditions. As a rule of thumb the observer dynamics should be faster than the desired closed-loop response determined by $A_m$.

When the properties of the disturbances are known, optimal filtering theory can be used to determine the optimal observer. For instance, if the system is described by

$$Ay - Bu + Ce$$

where $e$ is white noise, then the optimal choice is $A_o = C$.

**Model-following or Pole Placement**

Model-following generally means the case in which the closed-loop output is made to follow a model as specified by Eq. (A.2), i.e., both poles and zeros of the model are specified in the formulation. Pole placement, on the other hand, specifies only the closed-loop poles. To make a good control system it is necessary to consider the zeros of the open-loop system when specifying the closed-loop poles. Compare Eq. (A.6). In some papers the authors make a clear distinction between model-following and pole placement. Our view is that these two notations are essentially the same. The relation between model-following and pole placement establishes the connections between self-tuning regulators and model-reference systems. It follows from the equations that the control law of Eq. (A.4) can be written as

$$u = G_m G^{-1} u_c - \frac{S}{R} (y - y_m)$$

where $G_m = B_m / A_m$ and $G = B / A$. See Fig. A.1. The block diagram in Fig. A.1 is useful for the purpose of explaining model-following, but the system cannot be implemented as it is shown in the figure, because
the inverse process model $A/B$ is generally not realizable. However, the cascade combination of the reference model and the inverse process model is realizable if the model-following problem is well posed, i.e., if Eq. (A.3) is satisfied.

In practice it is common to make several approximations. The advantage of the feedforward path is that it makes it possible to obtain a fast response. This can often be achieved with a crude approximation of the reference model and the inverse response model. A lead-lag filter is often used as an approximation. Also notice that the reference model and the inverse process model can be nonlinear without causing any stability problems, because they only appear as part of a feedforward compensator. The low-frequency behavior is assured by the feedback.

**Sensitivity of the Design**

Pole placement or perfect model-following is a straightforward design method that is easy to automate. The sensitivity of the closed-loop system depends, however, on the true process model, the observer polynomial and the desired response chosen.

The following result from Åström and Wittenmark (1984) describes the influence of modeling errors.

**Theorem A.1—Robustness**

Consider a pole placement design based on an approximate model

$$G = \frac{B}{A}$$

Let $G^o$ be the transfer function of the system to be controlled. Assume that $G$ and $G^o$ have the same number of poles outside the stability boundary and that $G_m = B_m/A_m$ is stable. Then the closed-loop system related to $G^o$ is stable if

$$G(z) - G^o(z) < \left| \frac{G(z)T(z)}{G_m(z)S(z)} \right| = \left| \frac{G(z)}{G_m(z)} \right| \cdot \left| \frac{G_{ff}(z)}{G_{fb}(z)} \right|$$  \hspace{1cm} (A.10)

for $z = i\omega$ in continuous time or $z = e^{i\omega h}$ in discrete time, where

$$G_{ff}(z) = \frac{T(z)}{R(z)} \quad G_{fb} = \frac{S(z)}{R(z)}$$

The theorem is easy to apply when a design has been done. The right-hand side of Eq. (A.10) can be easily calculated and does not depend on the true transfer function. The conditions on the model precision can thus be expressed in terms of frequency domain conditions.
It is also important that the choice of the desired model $G_m$ be related to the process $G$. The regulator obtained may be extremely sensitive to parameter variations if $A_o$ and $A_m$ are not chosen carefully. The poles of $G$ can be changed through feedback, but the zeros of $G$ can be changed only through cancellation and addition of the desired zeros. In the design it is therefore good practice not to change the open-loop zeros but to keep them in the model $G_m$.

**Disturbances with Known Dynamics**

The pole placement design can be modified to account for disturbances with known dynamics. Assume that the disturbance $v$ is generated from the dynamical system

$$A_d v = e$$

where $e$ is a pulse, a set of widely spread pulses, white noise, or the equivalent continuous-time concepts. For example, in discrete-time systems a step disturbance is generated by

$$A_d(q) = q - 1$$

and for continuous-time systems by

$$A_d(p) = p$$

A sinusoid is similarly generated in the discrete-time case by

$$A_d(q) = q^2 - 2q \cos \omega h + 1$$

and in the continuous-time case by

$$A_d(p) = p^2 + \omega^2$$

The system model then becomes

$$AA_d y = BA_d u + e$$

With the regulator of Eq. (A.4) we find

$$y = \frac{BT}{AR + BS} u_c + \frac{R}{A_d(AR + BS)} e$$

$$u = \frac{AT}{AR + BS} u_c - \frac{S}{A_d(AR + BS)} e$$

(A.11)
The closed-loop characteristic polynomial thus contains the disturbance dynamics as a factor. This polynomial is typically unstable. It follows from Eq. (A.11) that in order to maintain a finite output in case of these disturbances $A_d$ must divide $R$. This would make $y$ finite, but the controlled input $u$ may be infinite. This is of course necessary to compensate for an infinite disturbance. The polynomial $R$ must thus be of the form

$$R = R_1 A_d \tag{A.12}$$

This implies that a model for the disturbance dynamics is built into the regulator. The idea is called the internal model principle. Notice that $R$ will contain an integrator if the disturbance has step character.

Using Eq. (A.12), the design equation becomes

$$AA_d R_1 + BS = A_o A_m$$

This equation has a solution with $\deg S < \deg A + \deg A_d$. The compatibility conditions that ensure a causal control law are then

$$\deg A_o > 2 \deg A + \deg A_d - \deg A_m - 1$$

If the disturbances are generated by the model

$$A_d v = Ce$$

where $C$ is also known, the observer polynomial $A_o$ should be equal to $C$.

**Notch Filter Design**

When designing controls for systems with mechanical resonances, it is common to neglect the resonances in the design but to include a notch filter to make sure that the signals close to the resonances will not be amplified by the control system. Such ideas can also be incorporated to the pole placement design. Let $A_n$ be a polynomial representing the zeros of the notch filter. To make sure that the regulator has the notch property we must thus require that $A_n$ divides $S$. Hence

$$S = S_1 A_n$$

The design equation then becomes

$$AR + S_1 A_n B = A_o A_m$$

This equation has a solution with $\deg S_1 < \deg A$. To obtain a realizable control law we must thus require that $R$ have a degree not less than $\deg B + \deg A_n - 1$. The compatibility conditions thus become

$$\deg A_o \geq 2 \deg A + \deg A_n - \deg A_m - 1$$
Response to Command Signals

With the design chosen, the response to reference signals and load disturbances is generated by the characteristic equation $A_p A_m$. We sometimes wish to have a different response to command signals. This can be achieved by introducing

$$u_c = \frac{A_m}{A_c} u_r$$

The response to command signals is then

$$y = \frac{B_m}{A_c} u_r$$

Quadratic Optimization

Instead of using pole placement, some design methods are based on minimization of quadratic loss functions. A common criterion in continuous time is

$$J = \int_0^\infty (y^2(t) + \rho u^2(t)) \, dt$$

and in discrete time

$$J = \sum_{t=1}^{\infty} (y^2(t) + \rho u^2(t))$$

The control law that minimizes this criterion is obtained by first finding a stable polynomial such that

$$\rho A(s)A(-s) + B(s)B(-s) = \rho P(s)P(s)$$

and in discrete time

$$\rho A(z)A^*(z) + z^d B(z)B^*(z) = \rho P(z)P^*(z)$$

where $A^*(z)$ is the reciprocal polynomial of $A(z)$. The control law is then obtained as the solution to the pole placement problem with $A_m = P$. The optimization method can be regarded as a convenient way of choosing the closed-loop polynomial. This is also convenient from the point of view of operator interface, since it is only necessary to choose one parameter $\rho$ instead of $n$ zeros of a polynomial.

Actuator Saturation and Anti-windup

The design method above is based on purely linear arguments. This is a good approximation, but there is one nonlinear effect that must always be considered, namely actuator saturation. If the linear regulators are
used without any precautions, performance may be bad when the actuator saturates. The control law is therefore implemented as follows

\[ A_o u = f(Tu_c - Sy + (A_o - R_o)u) \]

where the nonlinear function \( f \) is a model of the actuator nonlinearity. For a simple actuator that saturates at \( u_{\text{low}} \) and \( u_{\text{high}} \), the function becomes

\[ f(x) = \begin{cases} 
  u_{\text{low}} & x \leq u_{\text{low}} \\
  x & u_{\text{low}} < x < u_{\text{high}} \\
  u_{\text{high}} & u \geq u_{\text{high}} 
\end{cases} \]

This device will avoid integrator windup. It is discussed in depth in Åström and Wittenmark (1984).

Summary

The pole placement procedure can be modified to cover many situations that are important in practice. However, to avoid cluttering up the presentation of the adaptive algorithms we will primarily use the procedures when all or none of the zeros are canceled. In the design of practical adaptive algorithms the modifications discussed are very important. Once the basic ideas are understood, these modifications are easy to carry out.

The Diophantine Equation

The equation

\[ AR + BS = A_c \]  \hspace{1cm} (A.13)

plays a crucial role in the design procedure. This is a linear equation in the polynomials \( R \) and \( S \). There always exists a solution if \( A \) and \( B \) are relatively prime; the equation has many solutions. This is easily seen: if \( R_0 \) and \( S_0 \) are solutions, then

\[ R = R_0 + BQ \]
\[ S = S_0 - AQ \]

are also solutions where \( Q \) is an arbitrary polynomial. A particular solution can be specified in several different ways. The particular solution is constrained by the fact that the control law must be causal, i.e. \( \deg S \leq \deg R \). The compatibility conditions ensure that a causal solution exists.
Among the many choices, we can mention a few alternatives. The case \( \deg A_m = \deg A = \deg A_o + \deg B^+ + 1 \) gives \( \deg R = \deg S = \deg T = \deg A - 1 \). This is the standard case of a state feedback and a Luenberger observer. It corresponds to equality in all compatibility conditions.

**Continuous-time Systems**

For continuous-time systems it may be advantageous to find solutions in which the degree of \( R \) is larger than the degree of \( S \). This gives a rapid roll off of the loop gain at high frequencies and an associated robustness of the closed-loop system. This can be obtained from the given design procedures by selecting an observer polynomial of high degree and picking the appropriate solutions of the Diophantine equation.

**Discrete-time Systems**

For discrete-time systems it is advantageous to keep \( \deg R = \deg S \), to avoid an unnecessary delay in the regulator.

**Solving the Diophantine Equation**

By equating coefficients of equal order, the Diophantine equation given by Eq. (A.13) can be written as a set of linear equations:

\[
\begin{pmatrix}
1 & 0 & \ldots & 0 & b_0 & 0 & \ldots & 0 \\
a_1 & 1 & \ddots & \vdots & b_1 & b_0 & \ddots & \vdots \\
a_2 & a_1 & \ddots & 0 & b_2 & b_1 & \ddots & 0 \\
\vdots & \vdots & \ddots & 1 & \vdots & \vdots & \ddots & b_0 \\
a_n & \vdots & a_1 & b_n & \vdots & b_1 \\
0 & a_n & \vdots & 0 & b_n & \vdots \\
\vdots & \vdots & \ddots & \vdots & \ddots & \ddots \\
0 & \ldots & 0 & a_n & 0 & \ldots & 0 & b_n
\end{pmatrix}
\begin{pmatrix}
\tau_1 \\
\vdots \\
\tau_k \\
s_0 \\
\vdots \\
s_l
\end{pmatrix}
= 
\begin{pmatrix}
a_{c1} - a_1 \\
\vdots \\
a_{cn} - a_n \\
0 \\
\vdots \\
a_{ck+l+1}
\end{pmatrix}
\]

(A.14)

The matrix on the left-hand side is called the *Sylvester matrix* and occurs frequently in applied mathematics. It has the property that it is nonsingular if and only if the polynomials \( A \) and \( B \) do not have any common factors. If there are no common factors, there exists a unique solution to Eq. (A.14). Notice, however, the nonuniqueness with respect to the orders of \( R \) and \( S \). Different choices of \( k \) and \( l \) will give different \( R \) and \( S \), as discussed above.
The solution to Eq. (A.14) can be obtained by Gaussian elimination. This method does not use the special structure of the Sylvester matrix. There are also special polynomial methods for solving Eq. (A.14) directly. The calculations may be too time-consuming to be used in adaptive controllers, in which the design calculations are redone at each sampling point. The pole placement procedure with all process zeros or all poles canceled has the advantage that the Sylvester matrix is triangular. This means that the coefficients in the $R$ and $S$ polynomials can be obtained one at a time.

It is also possible to solve the Diophantine equation using polynomial calculations. This has the advantage that possible common factors in the polynomials $A$ and $B$ can be canceled. The method is based on a classical algorithm due to Euclid.

**Euclid’s Algorithm**

This algorithm finds the greatest common divisor $G$ of two polynomials $A$ and $B$. If one of the polynomials, say $B$, is zero then $G$ is equal to $A$. If this is not the case the algorithm is as follows. Put $A_0 = A$ and $B_0 = B$ and iterate the equations

$$A_{n+1} = B_n$$
$$B_{n+1} = A_n \mod B_n$$

until $B_{n+1} = 0$. The greatest common divisor is then $G = B_n$. Backtracking we find that $G$ can be expressed as

$$AX + BY = G$$

where the polynomials $X$ and $Y$ can be found by keeping track of $A_n \div B_n$ in Euclid’s algorithm. This establishes the link between Euclid’s algorithm and the Diophantine equation. The extended Euclidean algorithm gives a convenient way to determine $X$ and $Y$ as well as the minimum degree solutions $U$ and $V$ to

$$AU + BV = 0$$

Equations (A.16) and (A.17) can be written as

$$F \begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} X & Y \\ U & V \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} G \\ 0 \end{pmatrix}$$

The matrix $F$ can thus be viewed as the matrix, which performs row operations on $[A \ B]^T$ to give $[G \ 0]^T$. A convenient way to find $F$ is to observe that

$$\begin{pmatrix} X & Y \\ U & V \end{pmatrix} \begin{pmatrix} A & 1 & 0 \\ B & 0 & 1 \end{pmatrix} = \begin{pmatrix} G & X & Y \\ 0 & U & V \end{pmatrix}$$
The extended Euclidean algorithm can be expressed as follows: Start with the matrix

\[
\begin{pmatrix}
A & 1 & 0 \\
B & 0 & 1 \\
\end{pmatrix}
\]

Perform elementary polynomial row operations until a matrix with zero in the (2,1) position is obtained, i.e.,

\[
\begin{pmatrix}
G & X & Y \\
0 & U & V \\
\end{pmatrix}
\]

which gives the solution.

**Solving the Diophantine Equation**

Using the extended Euclidean algorithm it is now straight-forward to solve the Diophantine equation

\[
AR + BS = C \quad \text{(A.19)}
\]

This is done as follows: Determine the greatest common divisor \( G \) and the associated polynomials \( X, Y, U, \) and \( V \) using the extended Euclidean algorithm. To have a solution to Eq. (A.19) \( G \) must divide \( C \). A particular solution is given by

\[
\begin{align*}
R_0 &= XC \div G \\
S_0 &= YC \div G
\end{align*} \quad \text{(A.20)}
\]

and the general solution is

\[
\begin{align*}
R &= R_0 + QV \\
S &= S_0 + QU
\end{align*} \quad \text{(A.21)}
\]

where \( Q \) is an arbitrary polynomial.

**References**

The pole placement design is extensively discussed in:


The choice of the closed-loop system dynamics and regulator design are discussed in:

More about the Sylvester matrix can be found in:


Solution of the Diophantine equation is discussed in:


The notion of positive realness is important in analysis and design of an MRAS. It is therefore useful to have criteria to determine if a given transfer function is positive real or strictly positive real. It is also useful to have methods for constructing a positive real transfer function.

The analysis problem is discussed first. “Positive real” (PR) and strictly positive real (SPR) were defined in Chapter 4 as follows.

**Definition B.1**
A complex valued function $G(s)$ is *positive real* (PR) if
1. $G(s)$ is real for $s$ real
2. $\text{Re } G(s) \geq 0$ for $\text{Re } s \geq 0$.

A function $G(s)$ is *strictly positive real* (SPR) if $G(s - \varepsilon)$ is positive real for some real $\varepsilon > 0$. □
An SPR function is not required to be a rational function, but that is the case that is of interest in this context. The definition implies that every point on the imaginary axis or in the right half of the s plane is mapped into the right half of the G plane or onto its imaginary axis.

It is easily shown that if G(s) is PR so is its reciprocal 1 G. It is difficult to test Condition 1 in the definition for the entire right half of the s plane. The following theorem for rational transfer functions gives an equivalent set of necessary and sufficient conditions that are more easily tested.

**Theorem B.1** Equivalent conditions for positive realness
A rational function G is PR if and only if it satisfies all of the following conditions.
1. G(s) is real for s real.
2. The denominator polynomial of G is stable or may have zeros on the iω axis. That is, G must be analytic in the right-half plane.
3. If G has poles on the iω axis, these poles must be simple and have real and positive residues.
4. The real part of G is nonnegative along the iω axis, i.e.,

\[
\text{Re}(G(i\omega)) \geq 0
\]

This theorem is useful because it can be tested analytically. The second condition can be tested by an ordinary Routh-Hurwitz stability test. The fourth condition can be tested similarly. To develop such a test, notice that if

\[
G(s) = \frac{B(s)}{A(s)}
\]

then

\[
\text{Re} G(i\omega) = \text{Re} \frac{B(i\omega)}{A(i\omega)} = \frac{B(i\omega)A(-i\omega)}{A(i\omega)A(-i\omega)}
\]

Since the denominator is nonnegative and G(i\omega) is symmetric with the real axis, it suffices to investigate whether the function

\[
f(\omega) = \text{Re} \{B(i\omega)A(-i\omega)\}
\]

is nonnegative for \( \omega \geq 0 \). Notice that f is an even function of \( \omega \). 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

\[
g(x) = f(\sqrt{x})
\]
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 \( g_1(x) = g(x) \), \( g_2(x) = g'(x) \). Then 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 \( a \to b \) is then \( V(a) - V(b) \).

The procedure is illustrated by an example.

**Example B.1—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. Further,

\[
f(\omega) = \text{Re} \left( (-\omega^2 + 6i \omega + 8)(-\omega^2 - 4i \omega + 3) \right) = \omega^4 + 13\omega^2 + 24
\]

Hence

\[
g(x) = x^2 + 13x + 24
\]

We get

\[
g_1(x) = x^2 + 13x + 24
\]

\[
g_2(x) = 2x + 13
\]

\[
g_3(x) = \frac{73}{4}
\]

Hence \( V(0) = 0 \), \( V(\infty) = 0 \), and the transfer function is SPR.

In the design of an MRAS the following problem also appears. Consider a rational transfer function of the form

\[
G(s) = \frac{B(s)}{A(s)}
\]

with \( A(s) \) known and stable. Find a \( B \) such that \( G \) is SPR. This problem can be solved by using the Kalman-Yakobovich Lemma. Introduce the following realization of \( 1 / A(s) \):

\[
\frac{dx}{dt} = \begin{pmatrix}
-a_1 & -a_2 & \cdots & -a_{n-1} & -a_n \\
1 & 0 & 0 & 0 & 0 \\
\vdots & & & & & \\
0 & 0 & 1 & 0 & 0
\end{pmatrix} x + \begin{pmatrix}
1 \\
0 \\
\vdots \\
0
\end{pmatrix} u
\]
Appendix B  Positive Real Transfer Functions

Choose a symmetric positive definite matrix $Q$. Solve the equation

$$A^T P + P A = -Q$$

The coefficients of a $B$ polynomial such that $B(s)/A(s)$ is SPR are then the first row of the $P$ matrix.

References

More about the properties and tests of PR and SPR can be found in:

INDEX

A
Ackermann, J., 39, 56
actuator saturation, 507
Adaline, 492, 498
adaptation problem, 16
adaptive autopilot for ship,
  22, 355, 463
adaptive control, abuses of,
  25, 443
adaptive control, applications
  of, 21, 439
adaptive control, history of 2
adaptive control, stochastic, 302
adaptive control theory, 14, 219

adaptive feedforward, 190, 452
adaptive filtering, 73, 479, 480
adaptive industrial controllers,
  443
adaptive manipulator control,
  151
adaptive noise cancellation, 481
adaptive prediction, 479
adaptive predictive control, 198
adaptive signal processing, 479
adaptive smoothing, 479
ADPCM, 482, 483
airplane control, 38, 368
airplane, dynamics, 38
Alster J., 327
Amsle B. E., 400
Anderson, B. D. O., 27, 217, 299
Annaswamy, A. M., 160, 298 299
anti-aliasing filter, 415, 429
anti-aliasing filter, delay due to 415
anti-reset windup, 420, 507
Anton, J. J., 497
applications, 439
a priori information 147
151, 208, 328
Arnold, V. I., 298
Årzen, K.-E. 497
Asea Brown Boveri, 443, 450
Asea Novatune, 450, 458
Ash, R. H., 341
associated differential equation, 268
342, 371, 398, 400, 437, 476
477, 497, 499, 508 511
Athans, M., 57, 298
Atherton, D. P., 400
augmented error, 137 143
144, 183
auto-regression 75
auto-tuning 7 23, 328, 441, 447
averaging, 18 235, 237
averaging stochastic 267
averaging theory, 3, 298
B
Bai E. W., 218
band-pass filters, 430
Barnett, S., 512
Barr, A., 497
Bar-Shalom, Y., 325, 327
Belanger, P. R., 327
Bellman equation, 19, 303, 311
Bellman, R. 2 28, 298, 325
Bengtsson, G., 477
Bessel filter, 415
BIBO stability, 132
Bierman, G. J., 104, 438
Billings, S. A., 28, 160, 437 476
Bitmead, R. R., 27, 299
Blackman, P. F., 496
Blankinship, W. A., 512
Bode, H. W., 42, 57, 398
Bogoliubov, N. N., 298
Bohlin, T., 326
Böhm, J., 217
Boltzmann machine, 492
Borisson, U., 215
Boxes, 490
Braun, L., 26
Bristol, E. H., 477
Brockett, R. W., 370
Butchart, R. L., 159
Butterworth filter, 415
C
Cadorin, D., 327
Caines, P. E., 297
Canudas de Wit, C., 218
CARIMA model, 204
cautious control, 21, 308, 314
certainty equivalence, 21, 302, 308
certainty equivalence principle 14, 164, 312
Chambers, R., 490, 497
chattering, 390
chemical reactor control, 458
Clark, D., 300
Clarke, D. W., 165, 188, 192, 214, 215, 216, 217 437
Clary, J. B., 217
closed-loop estimation, 82
Cohen, G. H., 341
combustion control 364
computational delay, 419
computing power, 456
concentration control 31, 349
connectionist model, 490
consistency, 63
constant-gain feedback, 30
constant trace algorithm, 411
control design, 446, 447, 452
        455, 472, 499
corvergence, 63, 219, 229
Coon, G. A., 341
covariance resetting, 230
Craig, J. J. 162
Cutler, C. R. 216

D
Dasgupta, S., 217
dead zone, 264, 411, 413
Decaulne, P. 400
De Keyser, R. M. C., 216
delta operator, 210, 217, 278
describing function method, 333
Desoer, C. A., 161
Desphande, P. B., 341
dialysis, 468
Diophantine equation, 109,
        501, 508
direct adaptive controller,
        14, 165
directional forgetting, 412
direct self-tuner, 433
direct self-tuning regulator,
        173, 177
discounting factor, 68
distillation column, 448
disturbance annihilation
        filter, 405
dither signal, 387
Doyle, J. C., 57, 399
DPCM, 482
Draper, C. S., 496
Draper, N. R., 104
dual control, 20, 304, 306
dual controller, 312
dual input describing function
        381, 382
Dugard, L., 215
dynamic positioning, 41

E
Edgar, T. F., 28, 476
Edmunds, J. M. 214
EEAS, 389
Egardt, B., 160, 215, 227,
        229, 297, 477
Electromax V, 444
electronic amplifier, 42
Elevitch, C., 321 327
Elliot, H., 216
ELS, 78
Emelyanov, S. V., 400
equation error, 75
Eriksson, J., 371, 477
error augmentation, 143
error model, 8, 135, 141
estimation, 58
estimation, continuous time
        model, 71, 76
estimator filtering, 405, 407
estimator implementation,
        89, 403
estimator windup, 410, 411
Euclid’s algorithm, 510
Eurotherm, 449
EXACT, 445
excitation condition, 73
experimental condition, 79
expert control, 479, 487, 489
expert systems, 488
explicit adaptive controller, 14
explicit criterion minimization,
       117
explicit self-tuning control, 165
exponential discounting, 68
exponential forgetting, 68, 410
extended least squares, 78
externally excited adaptive
        system, 388
extremum control, 484

F
feedforward, 190, 452, 454
Feigenbaum, E. A., 497
Feldbaum, A. A., 304, 325
Feuer, A., 297
filtering, 404, 407, 415, 472
finite impulse response models
   FIR, 72
FIR model, 72
First Control Systems, 443, 453
Firstline, 453
Firstloop, 453
Fisher Control, 443, 447
flight control systems, 38, 368
Flügge-Lotz, I., 379
forgetting factor, 68, 410
Fortescue, T. R., 218
Foxboro, 443, 445
Francis, B. A., 57
Franklin, G. F., 217
fuel-air control, 366
Fu, K. S., 497
fuzzy control, 491

G
gain changers in SOAS, 386
gain limitation, 121, 188, 265
gain schedules, 442
gain scheduling, 6, 343
gain scheduling, applications
   of, 355, 447, 463, 465
gain scheduling, design of,
   346, 442
Gambro AB, 468
Gauss, K. F., 59
Gawthrop, P. J., 27, 165, 188,
   192, 214, 215, 217, 297
Gelb, A., 399
genralized minimum-variance
   controller, 188
genralized predictive control,
   204
Gille, J. C., 400
global stability, 220
Gold B., 438
Goodwin, G. C., 27, 104, 160,
   161, 214, 215, 217, 297, 299,
   327, 437, 438
Gorozdos, R. E., 400
Gough, N. E., 400
gradient method, 110
Gregory, P. C., 26, 399
Grimble M. J., 217
Grossberg, S., 498
Guckenheimer, J., 298
Gupta, M. M., 26
Gutman, P. O., 398

H
Hagander, P., 301
Hägglund, T., 218, 341, 342,
   400, 438, 477
Hahn, W., 125, 161
Hale, J. K., 298
Halousková, A., 217
Hammerstein model, 486, 487
Hang, C. C., 161
Hanson, R. J., 104
harmonic balance, 333
Harris, C. J., 28, 160, 437, 476
Hawk, W. M., 476
Hayes-Roth, F., 497
Hebb, D. O., 492, 497
Hebb's principle, 492
Hecth-Nielsen, R., 498
Helmersson, A., 318, 326
high frequency gain, 141,
   144, 282
high gain feedback, 42, 373, 380
Hill, D. J., 161, 299, 437
Holmes, P., 298
Holst, J., 300
Hoopes, H. S., 476
Hopfield, J. J., 493, 498
Hopfield network, 493
Horowitz, I. M., 27, 56, 373,
   398, 399, 400
Householder transformation, 92
Hsu, P., 162
Hughes, D. J., 327
Hunt, L. R., 370, 371
hyperstate, 19, 303, 307

I
IEEE, 56
implementation, 402
implicit adaptive controller, 14
implicit self-tuning control, 165
indirect adaptive controller,
14, 165
indirect self-tuning regulator, 166
indirect self-tuning regulator,
algorithm of, 167
industrial adaptive controllers,
443
industrial experiences, 21,
445, 447, 455
initialization, 425
input-output stability, 131
instability mechanisms, 285
instantaneous gain, 141, 144, 282
integrator windup, 420
internal model principle, 506
Ioannou, P. A., 299
Irving, E., 218
Isermann, R., 437
Ismail, Z. M., 400
ITAE filter, 415
Itkis, U., 400

J
Jacobs, O. L. R., 56, 326, 327
Jayant, N. S., 496
Jezek, J., 512
Johansson, R., 215, 216, 218, 300
Johnson Jr, C. R., 27, 299, 496

K
Kaczmarz's algorithm, 69, 70
Källström, C. G., 371, 477
Kalman, R. E., 27, 165, 214
Kalman-Yakubovich lemma,
129, 161
Kanjilal, P. P., 217
Kárnya, M., 217, 218
Kershenbaum, L. S., 218
Kezer, A., 27, 159
King, R. E., 400
Kockum Sonics AB, 464
Kohonen, T., 498
Koivo, H., 215
Kokotovic, P. V., 27, 299
Kosut, R. L., 27, 299
Kraus, T. W., 477
Kreisselmeier, G., 299
Krener, A. J., 370
Krylov, A. N., 298
Kučera, V., 512
Kulhavý, R., 217, 218, 438
Kumar, P. R., 299, 300
Kushner, H., 300

L
Lachmann, K., 437
lambda sond, 367
Landau, Y. D., 160
Larimore, M. G., 496
Larminat, P. de, 297
Lawson, C. L., 104
leakage, 264, 413
Leal, R. L., 217
learning, 490
least-squares estimation, 10, 59
least-squares estimation,
recursive, 66
least-squares method, normal
equation, 61
least-squares method, recursive
computations, 64
least-squares method, statistical
interpretation, 63
Leeds and Northrup, 443, 444
Lefschetz, S., 161
Lenat, D., 497
Lennartson, B., 437, 511
Leufel, H., 477
level control, 448
Lewis, R. C., 476
limit cycle oscillation, 336, 381
linear quadratic self-tuning regulator, 194
linguistic control, 491
Lin, S.-F., 299
Lin, Y.-H., 160, 297
Li, W., 162
Li, Y. T., 496
Ljung, L., 28, 103, 104, 161, 268, 299, 300, 438
LMS algorithm, 481
Long, R. S., 215
loop transfer recovery, 373
Lozier, J. C., 399
Lyapunov’s second method, 124
Lyapunov, stability theorem, 125

M
Manfredi, C., 216
man-machine interface, 24, 442, 467
Mareels, I. M. Y., 27, 299
Märtensson, B., 292, 301
matrix inversion lemma, 65
Mayne, D. Q., 160, 217, 299, 437
McAvoy, T., 399
McClelland, J. L., 498
McMillan, G. K., 341
Mead, C. A., 498
Metivier, M., 300
Meyer, G., 370, 371
Michie, D., 490, 497
Micon P-200 Controller, 443
Middleton, R. H., 217, 299, 437
Milito, R., 327
Millman, R. S., 371
minimum-variance controller, 175
Minorsky, N., 298
Minsky, M., 492, 497
Mishkin, E., 26
MIT rule, 8, 110, 113, 131, 247
model-following, 107, 380, 503
model-reference adaptive systems, 7, 105, 124, 131, 247
model-reference adaptive systems, modified adjustment laws, 121
Mohtadi, C., 216, 217
Molusis, J. A., 327
monic, 108, 500
monitoring of excitation, 265, 409
Monopoli, R. V., 28, 160, 218, 370, 476
Mookerjee, P., 327
Morari, M., 399
Morse, A. S., 160, 291, 297, 301
Mosca, E., 216, 327
moving-average controller, 175
Moylan, P. J., 161
MRAS, 7, 105
MRAS based on stability theory, 124
MRAS, design using Lyapunov theory, 126
MRAS, general linear systems, 139
MRAS parallel, 106
MRAS, partially known systems, 151
MRAS problem, 107
myopic controller, 21, 305
Myron, T. J., 477

N
Narendra, K. S., 28, 160, 218, 297, 298, 299, 370, 476
Neumann, L., 398
neural net, 490
Nichols, N. B., 341
Niemi, A. J., 371
Noll, P., 496
Nomoto model, 357
nonlinear actuators, 30, 346, 507
nonlinear transformations, 346, 351
nonstationary disturbances, 40
normal equation, 61, 71
normalization, 122, 266
normalized relative sensitivity, 44
Norton, J. P., 104
Novatune, 23, 450
Novatune applications, 458
Nussbaum, R. D., 291, 301
Nyquist, H., 42

O
ODE approach, 268
Oldenburger, R., 496
one-button-tuning, 329, 447
one-step control, 21, 305
operational experiences, 468, 474
Orava, P. J., 371
orthogonal transformation, 92
Osburn, P. V., 27, 159
output error, 75, 480
overparameterization, 239, 280

P
Padilla, C. S., 327
Padilla, R. A., 327
Papert, S., 492, 497
parameter convergence, 17, 117, 229
parameter equilibrium, 239, 285
parameter estimation, 3, 58, 444, 445, 447, 452, 455, 471
parameterization, 276
parametric excitation, 285
Parks, P. C., 27, 159, 160, 161
partially known systems, 151, 208, 279
passive, 133
Patchell, J. W., 326
Payne, R. L., 327
PE, 74
Pelegrin, M. J., 400
perceptron, 491
persistent excitation, 17, 73, 74, 80, 81, 118, 233
persistent excitation of order n, 74
perturbation signals, 304, 315, 456
Peterka, V., 104, 165, 214, 216, 217
Peterson, B. B., 298
pH control, 359
pole-excess, 176, 500
pole placement, 110, 503
pole placement design, 499
Popov, V. M., 161
positive real, 129, 513
postsampling filters, 418
Pott, D. C., 370
Powell-Process Instruments, 443
PR, 129, 513
practical aspects, 402, 472
Prager, D., 214
Praly, L, 27, 297, 299
presampling filter, 415
pretune, 445, 446
pre-tuning, 328, 332, 335, 457
prior information, 444, 446, 447
Priouret, P., 300
probing, 321
projection algorithm, 69, 70, 72
projections, 264
prototype algorithms, 429
Protuner, 443
pulp dryer control, 460

Q
QR transformation, 92
quadratic optimization, 194, 204, 507

R
Rabiner, L. R., 438
Ramadge, P. J., 161, 297
Ramaker, B. C., 216
real-time estimation, 10, 58, 66, 71
receding-horizon controller, 198
reciprocal polynomial, 165
recursive maximum likelihood, 78
regression model, 60
regression variable, 60
regressor, 60
regularized constant trace algorithm, 411
regulator design, 499
regulator structures, 456
relatively prime, 500
relay feedback, 332, 380
relay oscillation, 333, 336, 383
relay systems, 379
reset action, automatic, 424
resetting, 68, 230
reset windup, 420
residual, 60
Riedle, B. D., 27, 299
Rippe, A., 477
RLS, 66
RML, 78
robot, 34, 151, 374
robot control, 151, 374
robotics, 34, 151, 374
robust controllers, 4, 30, 42, 373
robust design methods, 373
robust estimation, 91
robust high-gain control, 4, 42, 373
robustness, 219, 247
Rocchi, S., 327
Rohrs, C., 298
rolling mill, control of, 461
Rosenberg, C. R., 498
Rosenblatt, F., 491, 497
rotary dryer, 39
Rudolph, W., 477
Rumelhart, D. E., 498
Schwartz, A., 300
Seborg, D. E., 28, 476
Sejnowski, T., 498
self-organizing control (SOC), 1
self-oscillating adaptive systems, 379
self-oscillating adaptive systems, design procedure, 385
self-oscillating adaptive systems, main result, 383
self-oscillating adaptive systems, principle of, 380
self-oscillating adaptive systems (SOAS), 5, 372
self-tuning feedforward, 190, 451, 454, 460
self-tuning regulator, 11, 163
sensitivity derivative, 111
separation, 312
set-point limits, 473
Shackcloth, B., 159
Shah, S. L., 28, 476
Shaked, U., 398
Shapiro, A., 399, 400
Shinskey, F. G., 341
ship steering, 355
ship steering autopilot, 463
Sidi, M., 398
sign-sign algorithm, 111, 483
Singh, R. P., 160, 298
singular perturbations, 3
Sin, K. S., 27, 104, 214, 297, 438
sliding mode, 390
Slotine, J. J., 162
slowly varying parameters, 68, 237
slow perturbations, 238
small gain theorem, 134
Smay, J. W., 399, 400
Smith, C. L., 341
Smith, H., 104
SOAS, 379
Söderström, T., 28, 103, 104, 300, 437, 438
S
SPR, 16, 129, 130, 134, 136, 513
SPR rule, 130, 131, 247
square root method, 92
stability, 16, 118, 219
startup, 473
Stearns, S. D., 496
Steermaster, 464
Stein, G., 56, 57, 298, 370, 399
Sten, L., 371, 477
Sternby, J., 301, 326, 327, 477
stochastic adaptive control,
18, 302
stochastic approximation, 70, 72
stochastic averaging, 267
stochastic control, 3, 303
Stoica, P., 103
STR, 11, 163
strictly passive, 133
strictly positive real, 16, 129, 513
structured model uncertainty, 45
Supertuner, 443
supervision logic, 475
Su, R., 370, 371
Sussman, H. J., 371
Sylvester matrix, 509
system identification, 3, 72

T
Tank, D. W., 493, 498
Techmatlon, 443
temperature control, 448
Thompson, M. O., 399
Thorell, N. E., 371, 477
time scaling, 346, 349, 463
Toyo Systems, 443
transversal filter, 72
Treichler, J. R., 496
Trulsson, E., 161
Tse, E., 325
Tsypkin, Ya. Z., 2, 26, 161,
342, 399
Tuffs, P. S., 216
tuning automatic, 23, 328,
445, 447, 457
tuning on demand, 457
tuning problem, 16
Turnbull Control, 443
turning regulator, 466
turn-off phenomenon, 314
Tustin's operator, 278
two-degree-of-freedom system,
5, 373

U
ultimate gain, 334
ultimate period, 334
ultrafiltration, 468
Unbehauen, H., 28, 476
underlying design problem,
11, 164
universal stabilizers, 290
unmodeled dynamics, 247, 250,
251, 263, 404, 407
unstructured model uncertainty,
45
Utkin, V. I., 400, 401

V
Valavani, L. S., 160, 297, 298
Van Cauwenberghe, A. R., 216
Vander Velde, W. E., 399
Varaiya, P., 300
variable structure system,
373, 389
Vidyasagar, M., 125, 161
VSS, 389

W
Watermann, D., 497
Weinberg, L., 516
Wellstead, P. E., 214, 215, 438
Welsh, J. R., 399
Wenk, C. J., 327
Whatley, M. J., 370
Whitaker, H. P., 27, 106, 159
Widrow, B., 492, 496, 498
Wieslander, J., 165, 214, 326
Wittenhausen, H. S., 326
Wittenmark, B., 27, 104, 165,