PID Controllers, 2nd Edition

by
Karl J. Åström

and

Tore Hägglund

Printed in the United States of America.
10 9 8 7 6 5 4

ISBN 1-55617-516-7

Library of Congress Cataloging-in-Publication Data

Aström, Karl J (Karl Johan), 1934 -
PID controllers; theory, design, and tuning/Karl Johan Aström and Tore Hägglund.--2nd ed.
p. cm.
Rev. ed. of: Automatic tuning of PID controllers. C1988.
Includes bibliographical references and index.
ISBN 1-55617-516-7
1. PID controllers. I. Hägglund, Tore. II. Aström, Karl J. (Karl, Johan), 1934-
Automatic tuning of PID controllers. III. Title
TJ223.P55A87 1994
629.8--dc20 94-10795
CIP

Preface

In 1988 we published the book *Automatic Tuning of PID Controllers*, which summarized experiences gained in the development of an automatic tuner for a PID controller. The present book may be regarded as a continuation of that book, although it has been significantly expanded. Since 1988 we have learned much more about PID control as a result of our involvement in research and industrial development of PID controllers. Because of this we strongly believe that the practice of PID can be improved considerably, and that this will contribute significantly to improved quality of manufacturing. This belief has been strongly reinforced by recent publications of the industrial state of the art, which are referenced in Chapter 1.

The main reason for writing this book is to contribute to a better understanding of PID control. Another reason is that information about PID control is scattered in the control literature. The PID controller has not attracted much attention from the research community during the past decades, and it is often covered inadequately in standard textbooks in control. We believe that this book will be useful to users and manufacturers of PID controllers as well as educators. It is important to teach PID control in introductory courses on feedback control at universities, and we hope that this book can give useful background for such courses.

It is assumed that the reader has a control background. A reader should be familiar with concepts such as transfer functions, poles, and zeros. Even so, the explanations are elementary. Occasionally, we have stated facts without supporting detailed arguments, when they have seemed unnecessary, in an effort to focus on the practical aspects rather than the theory. A reader who finds that he needs som specific background in process control is strongly advised to consult a text in process control such as Seborg *et al.* (1989).

Compared to the earlier book we have expanded the material substantially. The chapters on modeling, PID control, and design of PID controllers have been more than doubled. The chapter on automatic tuning has been completely rewritten to account for the dynamic product development that has taken place in the last years. There are two new chapters. One describes new tuning methods. This

material has not been published before. There is also a new chapter on control paradigms that describes how complex systems can be obtained by combining PID controllers with other components.

We would like to express our gratitude to several persons who have provided support and inspiration. Our original interest in PID control was stimulated by Axel Westrenius and Mike Sommerville of Eurotherm who shared their experience of design and of PID controllers with us. We have also benefited from discussions with Manfred Morari of Caltech, Edgar Bristol of Foxboro, Ken Goff formerly of Leeds and Northrup, Terry Blevins of Fisher-Rosemount Control, Gregory McMillan of Monsanto. Particular thanks are due to Sune Larsson who initiated our first autotuner experiments and Lars Bååth with whom we shared the pleasures and perils of developing our first industrial auto-tuner. We are also grateful to many instrument engineers who participated in experiments and who generously shared their experiences with us. Among our research colleagues we have learned much from Professor C. C. Hang of Singapore National University with whom we have done joint research in the field over a long period of time. We are also grateful to Per Persson, who developed the dominant pole design method.

Several persons have read the manuscript of the book. Willy Wojsznis of Fisher-Rosemount gave many valuable suggestions for improvements. Many present and former colleagues at our department have provided much help. Special thanks are due to Eva Dagnegård and Leif Andersson who made the layout for the final version and Britt-Marie Mårtensson who drew many of the figures. Ulf Holmberg, Karl-Erik Årzén and Mikael Johansson gave very useful input on several versions of the manuscript.

Finally we would like to express our deep gratitude to the Swedish National Board of Industrial and Technical Development (NUTEK) who have supported our research.

KARL JOHAN ÅSTRÖM
TORE HÄGGLUND

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

karl_johan.astrom@control.lth.se
tore.hagglund@control.lth.se

Table of Contents

Introduction

The PID controller has several important functions: it provides feedback; it has the ability to eliminate steady state offsets through integral action; it can anticipate the future through derivative action. PID controllers are sufficient for many control problems, particularly when process dynamics are benign and the performance requirements are modest. PID controllers are found in large numbers in all industries. The controllers come in many different forms. There are standalone systems in boxes for one or a few loops, which are manufactured by the hundred thousands yearly. PID control is an important ingredient of a distributed control system. The controllers are also embedded in many special-purpose control systems. In process control, more than 95% of the control loops are of PID type, most loops are actually PI control. Many useful features of PID control have not been widely disseminated because they have been considered trade secrets. Typical examples are techniques for mode switches and anti-windup.

PID control is often combined with logic, sequential machines, selectors, and simple function blocks to build the complicated automation systems used for energy production, transportation, and manufacturing. Many sophisticated control strategies, such as model predictive control, are also organized hierarchically. PID control is used at the lowest level; the multivariable controller gives the setpoints to the controllers at the lower level. The PID controller can thus be said to be the "bread and butter" of control engineering. It is an important component in every control engineer's toolbox.

PID controllers have survived many changes in technology ranging from pneumatics to microprocessors via electronic tubes, transistors, integrated circuits. The microprocessor has had a dramatic influence on the PID controller. Practically all PID controllers made today are based on microprocessors. This has given opportunities to provide additional features like automatic tuning, gain scheduling, and continuous adaptation. The terminology in these areas is not well-established. For purposes of this book, *auto-tuning* means that the controller parameters are tuned automatically on demand from an operator or an external signal, and *adaptation* means that the parameters of a controller are continuously updated. Practically all

new PID controllers that are announced today have some capability for automatic tuning. Tuning and adaptation can be done in many different ways. The simple controller has in fact become a test bench for many new ideas in control.

The emergence of the fieldbus is another important development. This will drastically influence the architecture of future distributed control systems. The PID controller is an important ingredient of the fieldbus concept. It may also be standardized as a result of the fieldbus development.

A large cadre of instrument and process engineers are familiar with PID control. There is a well-established practice of installing, tuning, and using the controllers. In spite of this there are substantial potentials for improving PID control. Evidence for this can be found in the control rooms of any industry. Many controllers are put in manual mode, and among those controllers that are in automatic mode, derivative action is frequently switched off for the simple reason that it is difficult to tune properly. The key reasons for poor performance is equipment problems in valves and sensors, and bad tuning practice. The valve problems include wrong sizing, hysteresis, and stiction. The measurement problems include: poor or no anti-aliasing filters; excessive filtering in "smart" sensors, excessive noise and improper calibration. Substantial improvements can be made. The incentive for improvement is emphasized by demands for improved quality, which is manifested by standards such as ISO 9000. Knowledge and understanding are the key elements for improving performance of the control loop. Specific process knowledge is required as well as knowledge about PID control.

Based on our experience, we believe that a new era of PID control is emerging. This book will take stock of the development, assess its potential, and try to speed up the development by sharing our experiences in this exciting and useful field of automatic control. The goal of the book is to provide the technical background for understanding PID control. Such knowledge can directly contribute to better product quality.

Process dynamics is a key for understanding any control problem. Chapter 2 presents different ways to model process dynamics that are useful for PID control. Methods based on step tests are discussed together with techniques based on frequency response. It is attempted to provide a good understanding of the relations between the different approaches. Different ways to obtain parameters in simple transfer function models based on the tests are also given. Two dimensionfree parameters are introduced: the normalized dead time and the gain ratio are useful to characterize dynamic properties of systems commonly found in process control. Methods for parameter estimation are also discussed. A brief description of disturbance modeling is also

given.

An in depth presentation of the PID controller is given in Chapter 3. This includes principles as well as many implementation details, such as limitation of derivative gain, anti-windup, improvement of set point response, etc. The PID controller can be structured in different ways. Commonly used forms are the series and the parallel forms. The differences between these and the controller parameters used in the different structures are treated in detail. Implementation of PID controllers using digital computers is also discussed. The underlying concepts of sampling, choice of sampling intervals, and antialiasing filters are treated thoroughly. The limitations of PID control are also described. Typical cases where more complex controllers are worthwhile are systems with long dead time and oscillatory systems. Extensions of PID control to deal with such systems are discussed briefly.

Chapter 4 describes methods for the design of PID controllers. Specifications are discussed in detail. Particular attention is given to the information required to use the methods. Many different methods for tuning PID controllers that have been developed over the years are then presented. Their properties are discussed thoroughly. A reasonable design method should consider load disturbances, model uncertainty, measurement noise, and set-point response. A drawback of many of the traditional tuning rules for PID control is that such rules do not consider all these aspects in a balanced way. New tuning techniques that do consider all these criteria are also presented.

The authors believe strongly that nothing can replace understanding and insight. In view of the large number of controllers used in industry there is a need for simple tuning methods. Such rules will at least be much better than "factory tuning," but they can always be improved by process modeling and control design. In Chapter 5 we present a collection of new tuning rules that give significant improvement over previously used rules.

In Chapter 6 we discuss some techniques for adaptation and automatic tuning of PID controllers. This includes methods based on parametric models and nonparametric techniques. A number of commercial controllers are also described to illustrate the different techniques. The possibilities of incorporating diagnosis and fault detection in the primary control loop is also discussed.

In Chapter 7 it is shown how complex control problems can be solved by combining simple controllers in different ways. The control paradigms of cascade control, feedforward control, model following, ratio control, split range control, and control with selectors are discussed. Use of currently popular techniques such as neural networks and fuzzy control are also covered briefly.

References

A treatment of PID control with many practical hints is given in (Shinskey, 1988). There is a Japanese text entirely devoted to PID control by (Suda *et al.*, 1992). Among the books on tuning of PID controllers, we can mention (McMillan, 1983) and (Corripio, 1990), which are published by ISA.

There are several studies that indicate the state of the art of industrial practice of control. The Japan Electric Measuring Instrument Manufacturers' Association conducted a survey of the state of process control systems in 1989, see (Yamamoto and Hashimoto, 1991). According to the survey more than than 90% of the control loops were of the PID type.

The paper, (Bialkowski, 1993), which describes audits of paper mills in Canada, shows that a typical mill has more than 2000 control loops and that 97% use PI control. Only 20% of the control loops were found to work well and decrease process variability. Reasons for poor performance were poor tuning (30%) and valve problems (30%). The remaining 20% of the controllers functioned poorly for a variety of reasons such as: sensor problems, bad choice of sampling rates, and anti-aliasing filters. Similar observations are given in (Ender, 1993), where it is claimed that 30% of installed process controllers operate in manual, that 20% of the loops use "factory tuning," i.e., default parameters set by the controller manufacturer, and that 30% of the loops function poorly because of equipment problems in valves and sensors.

Process Models

2.1 Introduction

A block diagram of a simple control loop is shown in Figure 2.1. The system has two major components, the process and the controller, represented as boxes with arrows denoting the causal relation between inputs and outputs. The process has one input, the manipulated variable, also called the control variable. It is denoted by u. The process output is called process variable (PV) and is denoted by y. This variable is measured by a sensor. The desired value of the process variable is called the setpoint (SP) or the reference value. It is denoted by y_{sp}. The control error e is the difference between the setpoint and the process variable, i.e., $e = y_{sp} - y$. The controller in Figure 2.1 has one input, the error, and one output, the control variable. The figure shows that the process and the controller are connected in a closed feedback loop.

The purpose of the system is to keep the process variable close to the desired value in spite of disturbances. This is achieved by the feedback loop, which works as follows. Assume that the system is in equilibrium and that a disturbance occurs so that the process variable becomes larger than the setpoint. The error is then negative and the controller output decreases which in turn causes the process output to decrease. This type of feedback is called negative feedback, because the manipulated variable moves in direction opposite to the process variable.

The controller has several parameters that can be adjusted. The control loop performs well if the parameters are chosen properly. It performs poorly otherwise, e.g., the system may become unstable. The procedure of finding the controller parameters is called tuning. This can be done in two different ways. One approach is to choose some controller parameters, to observe the behavior of the feedback system, and to modify the parameters until the desired behavior is obtained. Another approach is to first develop a mathematical model that describes the behavior of the process. The parameters of the controller are then determined using some method for control design.

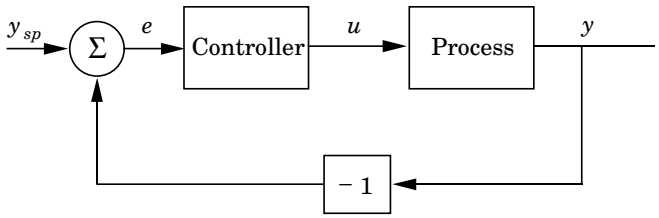

Figure 2.1 Block diagram of a simple feedback system.

An understanding of techniques for determining process dynamics is a necessary background for both methods for controller tuning. This chapter will present such techniques.

Static models are discussed in the next section. Dynamic models are discussed in Section 2.3. Transient response methods, which are useful for determining simple dynamic models of the process, are presented in Section 2.4. Section 2.5 treats methods based on moments. These methods are less sensitive to measurement noise and, furthermore, are not restricted to any specific input signal. The frequency response methods, described in Section 2.6, can be used to obtain both simple models and more detailed descriptions. Methods based on estimation of parametric models are more complex methods that require more computations but less restrictions on the experiments. These methods are presented in Section 2.7. The models discussed so far describe the relation between the process input and output. It is also important to model the disturbances acting on the system. This is discussed in Section 2.8. Section 2.9 treats methods to simplify a complex model and the problem of unmodeled dynamics and modeling errors. Conclusions and references are given in Sections 2.10 and 2.11.

2.2 Static Models

The static process characteristic is a curve that gives the steady state relation between process input signal u and process output y. See Figure 2.2. Notice that the curve has a physical interpretation only for a stable process.

All process investigations should start by a determination of the static process model. It can be used to determine the range of control signals required to change the process output over the desired range, to size actuators, and to select sensor resolution. It can also be used to assess whether static gain variations are so large that they have to be accounted for in the control design.

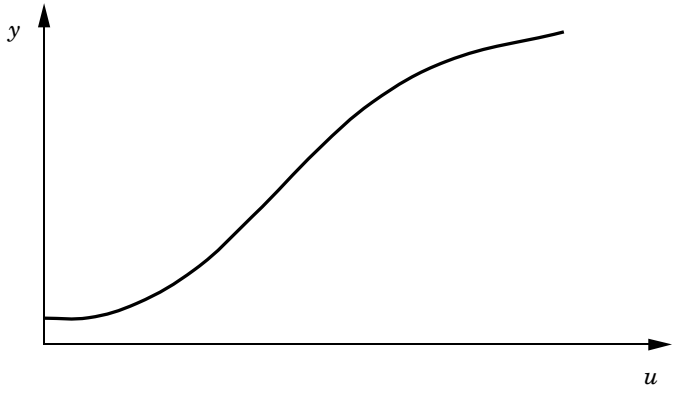

Figure 2.2 Static process characteristic. Shows process output y as a function of process input u under static conditions.

The static model can be obtained in several ways. It can be determined by an open-loop experiment where the input signal is set to a constant value and the process output is measured when it has reached steady state. This gives one point on the process characteristics. The experiment is then repeated to cover the full range of inputs.

An alternative procedure is to make a closed-loop experiment. The setpoint is then given a constant value and the corresponding control variable is measured in steady state. The experiment is then repeated to cover the full range of setpoints.

The experiments required to determine the static process model often give a good intuitive feel for how easy it is to control the process, if it is stable, and if there are many disturbances.

Sometimes process operations do not permit the experiments to be done as described above. Small perturbations are normally permitted, but it may not be possible to move the process over the full operating range. In such a case the experiment must be done over a long period of time.

Process Noise

Process disturbances are easily determined by logging the process output when the control signal is constant. Such a measurement will give a combination of measurement and load disturbances. There are many sophisticated techniques such as time-series analysis and spectral analysis that can be used to determine the characteristics of the process noise. Crude estimates of the noise characteristics are obtained simply by measuring the peak-to-peak value and by determining the average time between zero crossings of the error

signal. This is discussed further in Section 2.8.

2.3 Dynamic Models

A static process model like the one discussed in the previous section tells the steady state relation between the input and the output signal. A dynamic model should give the relation between the input and the output signal during transients. It is naturally much more difficult to capture dynamic behavior. This is, however, very significant when discussing control problems.

Fortunately there is a restricted class of models that can often be used. This applies to linear time-invariant systems. Such models can often be used to describe the behavior of control systems when there are small deviations from an equilibrium. The fact that a system is linear implies that the superposition principle holds. This means that if the input u_1 gives the output y_1 and the input u_2 gives the output y_2 it then follows that the input $au_1 + bu_2$ gives the output $ay_1 + by_2$. A system is time-invariant if its behavior does not change with time.

A very nice property of linear time-invariant systems is that their response to an arbitrary input can be completely characterized in terms of the response to a simple signal. Many different signals can be used to characterize a system. Broadly speaking we can differentiate between transient and frequency responses.

In a control system we typically have to deal with two signals only, the control signal and the measured variable. Process dynamics as we have discussed here only deals with the relation between those signals. The measured variable should ideally be closely related to the physical process variable that we are interested in. Since it is difficult to construct sensors it happens that there is considerable dynamics in the relation between the true process variable and the sensor. For example, it is very common that there are substantial time constants in temperature sensors. There may also be measurement noise and other imperfections. There may also be significant dynamics in the actuators. To do a good job of control, it is necessary to be aware of the physical origin the process dynamics to judge if a good response in the measured variable actually corresponds to a good response in the physical process variable.

Transient Responses

In transient response analysis the system dynamics are characterized in terms of the response to a simple signal. The particular signal is often chosen so that it is easy to generate experimentally. Typical

examples are steps, pulses, and impulses. Because of the superposition principle the amplitude of the signals can be normalized. For example, it is sufficient to consider the response to a step with unit amplitude. If $s(t)$ is the response to a unit step, the output $y(t)$ to an arbitrary input signal $u(t)$ is given by

$$y(t) = \int_{-\infty}^{t} u(\tau) \frac{ds(t-\tau)}{dt} d\tau = \int_{-\infty}^{t} u(\tau) h(t-\tau) d\tau \qquad (2.1)$$

where the impulse response $h(t)$ is introduced as the time derivative of the step response.

In early process control literature the step response was also called the reaction curve.

Pulse response analysis is common in medical and biological applications, but rather uncommon in process control. Ramp response analysis is less common. One application is the determination of the derivative part of a PID controller. In process control, the step response is the most common transient used for process identification. This is primarily because this is the type of disturbance that is easiest to generate manually. Step response methods are treated in detail in Section 2.4.

Frequency Response

Another way to characterize the dynamics of a linear time-invariant system is to use sine waves as a test signal. This idea goes back to Fourier. The idea is that the dynamics can be characterized by investigating how sine waves propagate through a system.

Consider a stable linear system. If the input signal to the system is a sinusoid, then the output signal will also be a sinusoid after a transient (see Figure 2.3). The output will have the same frequency as the input signal. Only the phase and the amplitude are different. This means that under stationary conditions, the relationship between the input and the output can be described by two numbers: the quotient (a) between the input and the output amplitude, and the phase shift (φ) between the input and the output signals. The functions $a(\omega)$ and $\varphi(\omega)$ describe a and φ for all frequencies (ω). It is convenient to view a and φ as the magnitude and the argument of a complex number

$$G(i\omega) = a(\omega) e^{i\varphi(\omega)} \qquad (2.2)$$

The function $G(i\omega)$ is called the frequency response function of the system. The function $a(\omega) = |G(i\omega)|$ is called the amplitude function, and the function $\varphi(\omega) = arg(G(i\omega))$ is called the phase function.

The complex number $G(i\omega)$ can be represented by a vector with length $a(i\omega)$ that forms angle $\varphi(i\omega)$ with the real axis (see Figure

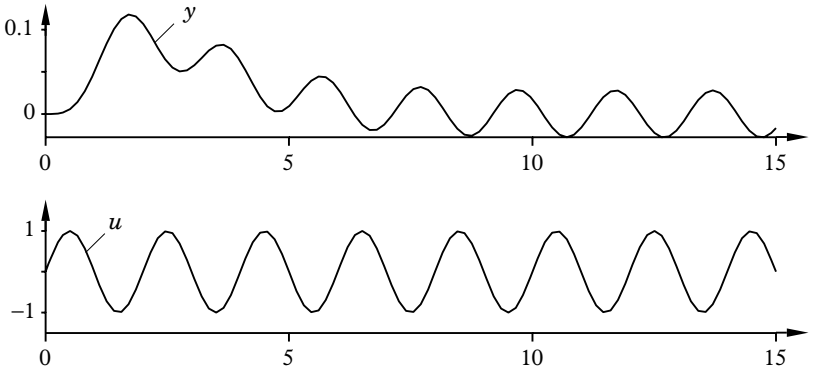

Figure 2.3 Input signal u is a sinusoid and output signal y becomes sinusoidal after a transient.

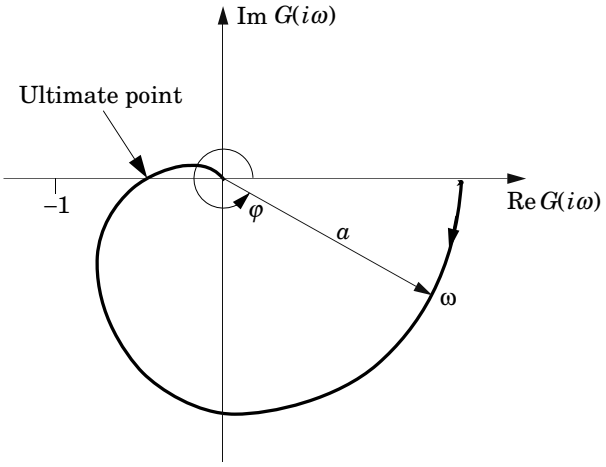

Figure 2.4 The Nyquist curve of a system.

2.4). When the frequency goes from 0 to ∞, the endpoint of the vector describes a curve in the plane, which is called the frequency curve or the Nyquist curve. The Nyquist curve gives a complete description of the system. It can be determined experimentally by sending sinusoids of different frequencies through the system. This may be time consuming. Normally, it suffices to know only parts of the Nyquist curve. For controller tuning there are some parts that are of particular interest. The lowest frequency where the phase is $-180°$ is called the ultimate frequency (ω_u). The corresponding point on the Nyquist curve is called the ultimate point. The value of $G(i\omega_u)$ is all that is needed for the tuning methods developed by Ziegler and Nichols.

The frequency response is intimately related to the Laplace trans-

form. Let $f(t)$ be a signal. The Laplace transform of the signal, $F(s)$, is then defined by

$$F(s) = \int_0^\infty e^{-st} f(t)dt \tag{2.3}$$

Let $U(s)$ and $Y(s)$ be the Laplace transforms of the input and the output of a linear time-invariant dynamical system. Assume that the system is at rest at time $t = 0$. The following relation then holds

$$Y(s) = G(s)U(s) \tag{2.4}$$

where $G(s)$ is the transfer function of the system.

It follows from Equation (2.3) that the Laplace transform of an impulse is 1. From Equation (2.4) we can conclude that $G(s)$ is the Laplace transform of the impulse response. The frequency response is simply $G(i\omega)$.

In the following sections we will show how linear system dynamics can be obtained experimentally. We will illustrate both transient and frequency response methods.

2.4 Step Response Methods

The dynamics of a process can be determined from the response of the process to pulses, steps, ramps, or other deterministic signals. The dynamics of a linear system is, in principle, uniquely given from such a transient response experiment. This requires, however, that the system is at rest before the input is applied, and that there are no measurement errors. In practice, however, it is difficult to ensure that the system is at rest. There will also be measurement errors, so the transient response method, in practice, is limited to the determination of simple models. Models obtained from a transient experiment are, however, often sufficient for PID controller tuning. The methods are also very simple to use. This section focuses on the step response method.

The Step Response

Assuming a control loop with a controller, the step response experiment can be determined as follows. Wait until the process is at rest. Set the controller to manual. Change the control variable rapidly, e.g., through the use of increase/decrease buttons. Record the process variable and scale it by dividing by the change in the control variable. The change in control variable should be as large as possible in order to get a maximum signal to noise ratio. The limit is set by permissible

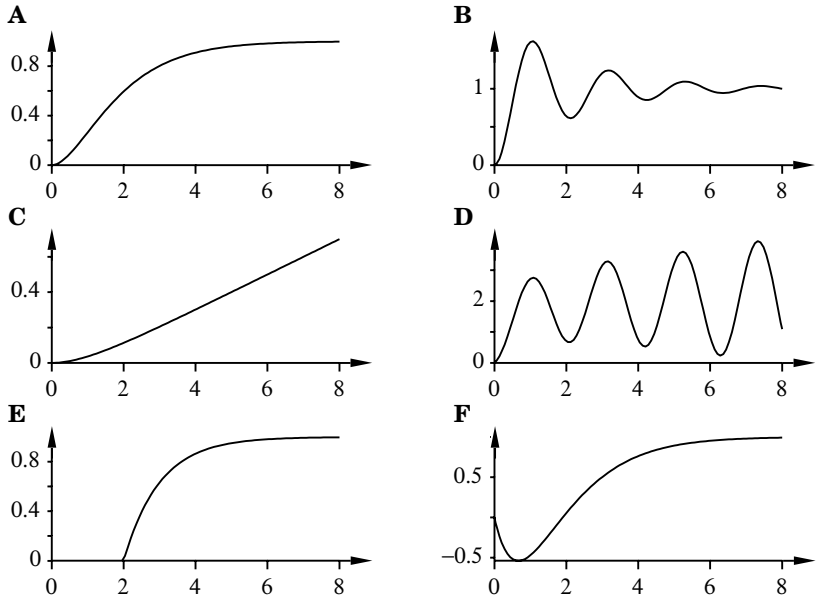

Figure 2.5 Open-loop step responses.

process operation. It is also useful to record the fluctuations in the measurement signal when the control signal is constant. This gives data about the process noise.

It is good practice to repeat the experiment for different amplitudes of the input signal and at different operating conditions. This gives an indication of the signal ranges when the model is linear. It also indicates if the process changes with the operating conditions.

Examples of open-loop step responses are shown in Figure 2.5. Many properties of the system can be read directly from the step response. In Figure 2.5A, the process output is monotonically changed to a new stationary value. This is the most common type of step response encountered in process control. In Figure 2.5B, the process output oscillates around its final stationary value. This type of process is uncommon in process control. One case where it occurs is in concentration control of recirculation fluids. In mechanical designs, however, oscillating processes are common where elastic materials are used, e.g., weak axles in servos, spring constructions, etc. The systems in Figures 2.5A and B are stable, whereas the systems shown in Figures 2.5C and 2.5D are unstable. The system in Figure 2.5C shows the output of an integrating process. Examples of integrating processes are level control, pressure control in a closed vessel, concentration control in batches, and temperature control in well isolated chambers. The common factor in all these processes is that some kind of storage occurs in them. In level, pressure and concentration control

storage of mass occurs, while in the case of temperature control there is a storage of energy. The system in Figure 2.5E has a long dead time. The dead time occurs when there are transportation delays in the process. The system in Figure 2.5F is a non-minimum phase system, where the measurement signal initially moves in the "wrong" direction. The water level in boilers often reacts like this after a step change in feed water flow.

If the system is linear, all step responses are proportional to the size of the step in the input signal. It is then convenient to normalize the responses by dividing the measurement signal by the step size of the control signal. Throughout this book we assume that this normalization is done.

The step response is a convenient way to characterize process dynamics because of its simple physical interpretation. Many tuning methods are based on it. A formal mathematical model can also be obtained from the step response. General methods for the design of control systems can then be used.

For small perturbations the static process model can be described by one parameter called the process gain. This is simply the ratio of the steady state changes of process output and process input. The gain can be obtained as the slope of the curve in Figure 2.2. It can also be obtained directly from a step response. For nonlinear systems the process gain will depend on the operating conditions. It is, however, constant for linear systems. For such systems the static properties are thus described by one parameter. Additional parameters are needed to also capture dynamics. Some simple parametric models will be described below. Stable processes with a monotone step response, as shown in Figure 2.5A, are quite common. Many methods to obtain parametric models from such a step response have been presented in the literature over the years. We will present here models with two, three, and, four parameters respectively.

Two-Parameter Models

The simplest parametric models of process dynamics have two parameters. One parameter can be process gain. The other has to capture the time behavior. The average residence time T_{ar} is a useful parameter. This is obtained as

$$T_{ar} = \frac{A_0}{K}$$

where K is the static process gain and A_0 is defined as

$$A_0 = \int_0^\infty (s(\infty) - s(t))dt$$

where $s(t)$ is the step response. Notice that $K = s(\infty)$ and that A_0 is the shaded area in Figure 2.6.

The time T_{ar} is a rough measure of the time it takes for the step response to settle. Using the static gain and the average residence time, the process can be approximated by the model

$$G_{2a}(s) = \frac{K}{1 + sT_{ar}} \tag{2.5}$$

We call this model the residence time approximation.

Another approximation to the step response that also has two parameters is given by the transfer function

$$G_{2b}(s) = \frac{a}{sL} e^{-sL} \tag{2.6}$$

This model corresponds to an integrator with dead time. This model is characterized by the two parameters, a and L, that are easily determined graphically from the step response (see Figure 2.6). The tangent to the step response $s(t)$ that has the largest slope is drawn, and the intersections of this tangent with the vertical and horizontal axes give a and L, respectively. The model given by Equation (2.6) is the basis for the Ziegler-Nichols tuning procedure discussed in Chapter 4. Notice that the model can also be fitted to unstable processes.

The properties of the approximations (2.5) and (2.6) are illustrated by an example.

EXAMPLE 2.1

The two-parameter models (2.5) and (2.6) have been fitted to the

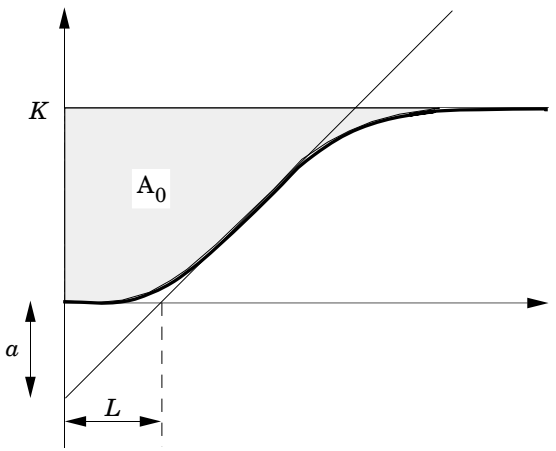

Figure 2.6 Graphical determination of a two-parameter model from a step response for a stable system with a monotone step response.

process model

$$G(s) = \frac{1}{(s+1)^8} \tag{2.7}$$

The following models were obtained

$$G_{2a}(s) = \frac{1}{1+8.0s} \qquad G_{2b}(s) = \frac{0.64}{4.3s}\, e^{-4.3s}$$

Figure 2.7 shows the step responses and the Nyquist curves of the transfer functions.

Notice that the model G_{2a} gives a good description of the step response for long times. The static gain is correct and the step response is very close to the correct one for large t. There are, however, large discrepances for small t. The system given by G_{2a} has, for example, a significant response at time $t = 2$, but the system (2.7) has barely responded at that time. The model G_{2b} has the opposite properties. It approximates the true step response very well in the interval $5 \leq t \leq 9$, but the approximation is very poor for large t.

These properties are also reflected in the Nyquist curves. They show that the average residence time approximation is quite good at low frequencies but very poor at high frequencies. The model G_{2b}, on the other hand, is poor at low frequencies but reasonable at middle range frequencies. □

Three-Parameter Models

Better approximations are obtained by increasing the number of pa-

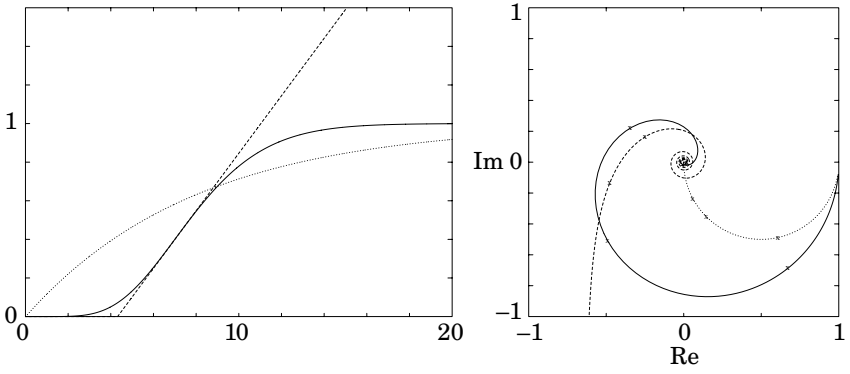

Figure 2.7 Step responses and Nyquist curves of the process $G(s) = 1/(s+1)^8$ (solid line) and the two-parameter models $G_{2a}(s)$ (dotted line) and $G_{2b}(s)$ (dashed line).

rameters. The model

$$G(s) = \frac{K}{1+sT}\, e^{-sL} \tag{2.8}$$

is characterized by three parameters: the static gain K, the time constant T, and the dead time L. This is the most common process model used in papers on PID controller tuning. The parameters L and T are often called the *apparent dead time* and the *apparent time constant*, respectively. The step response of the model (2.8) is

$$s(t) = K\left(1 - e^{-(t-L)/T}\right)$$

From this equation, it follows that the average residence time is

$$T_{ar} = \frac{\int_0^\infty (s(\infty) - s(t))dt}{K} = L + T$$

The ratio

$$\tau = \frac{L}{L+T} = \frac{L}{T_{ar}} \tag{2.9}$$

which has the property $0 \leq \tau \leq 1$, is called the *normalized dead time*. This quantity can be used to characterize the difficulty of controlling a process. It is sometimes also called the *controllability ratio*. Roughly speaking, it has been found that processes with small τ are easy to control and that the difficulty in controlling the system increases as τ increases. Systems with $\tau = 1$ correspond to pure dead-time processes, which are indeed difficult to control well.

The parameters in the model (2.8) can be determined graphically. The static gain (K) is obtained from the final steady-state level of the process output. Remember that the process output must be scaled with the change in the control variable. The intercept of the tangent to the step response that has the largest slope with the horizontal axes gives L (see Figure 2.8). The dead time L can also be obtained as the time between the onset of the step and the time $s(t)$ has reached a few percent of its final value. There are different ways to determine T. One method determines T from the distance AC in Figure 2.8, where the point C is the time when the tangent intersects the line $s(t) = K$. Another method determines T from the distance AB in Figure 2.8, where B is the time when the step response has reached the value $0.63K$. Both methods give identical results if the process dynamics are given by Equation (2.8), but they may differ significantly in other cases. The method based on the point B gives normally better approximations. The other method tends to give a too large value of T.

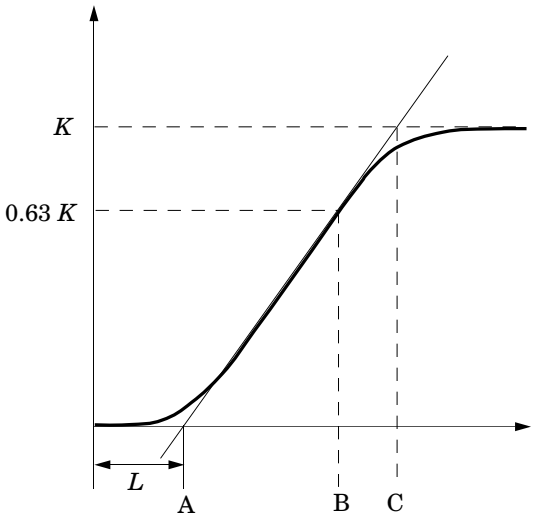

Figure 2.8 Graphical determination of three-parameter models for systems with a monotone step response.

EXAMPLE 2.2

The three-parameter models of the process model (2.7) are

$$G_{3a}(s) = \frac{1}{1+6.7s}\,e^{-4.3s} \qquad G_{3b}(s) = \frac{1}{1+4.3s}\,e^{-4.3s}$$

where the time constant T is determined from the point C in model G_{3a}, and from the point B in the model G_{3b}. Figure 2.9 shows the step responses of the true process and the models, as well as the Nyquist curves of the transfer functions. The figure shows that the time constant T is overestimated in the model G_{3a}. This overestimation is unfortunately common in this method, since most process control plants have an S-shaped step response similar to the model (2.7). Notice that the true step response and the step response of the model G_{3b} coincide at the 63% point. □

Another Model Structure

The model (2.8) is by far the most commonly used model in the papers of PID controller tuning. In spite of this, it is not a representative model. In fact, the conclusions drawn based on this model may often be misleading when applied to real processes. This will be illustrated by several examples in Chapter 4. One reason for this is that the step response of the model (2.8) is not S-shaped, or equivalently, that the frequency response of the model does not decay fast enough for high frequencies.

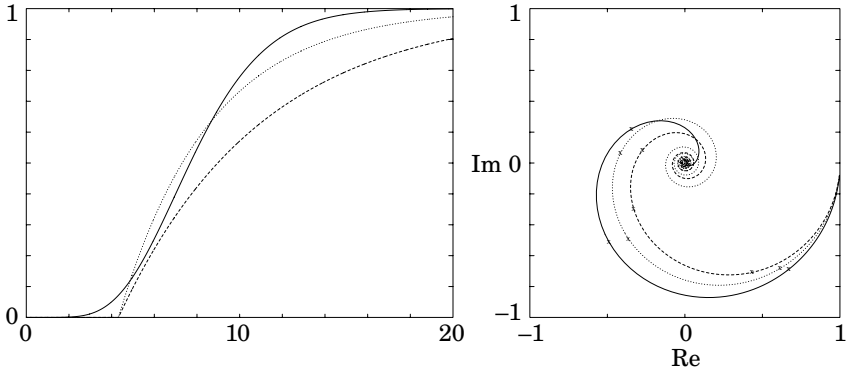

Figure 2.9 Step responses and Nyquist curves of the process $G(s) = 1/(s+1)^8$ (solid line) and the three-parameter models $G_{3a}(s)$ (dashed line) and $G_{3b}(s)$ (dotted line).

Another three-parameter model is

$$G(s) = \frac{K}{(1+sT)^2}\, e^{-sL} \tag{2.10}$$

The step response of this model is

$$s(t) = K\left(1 - \left(1 + \frac{t-L}{T}\right) e^{-(t-L)/T}\right) \tag{2.11}$$

This model has an S-shaped step response and often gives a better approximation than the first-order plus dead-time model (2.8). Static gain K and dead time L can be determined in the same way as for the model (2.8). Time constant T can then be determined from Equation 2.11 if the value of the step response at one time is known. The equation obtained must be solved numerically.

EXAMPLE 2.3

Fitting the model (2.10) to the process model (2.7) gives

$$G_{3c}(s) = \frac{1}{(1+2.0s)^2}\, e^{-4.3s}$$

The gain $K = 1$ is obtained from the steady-state value of the signal, and the dead time $L = 4.3$ is obtained from the intersection of the tangent with the largest slope and the horizontal axis as in the previous examples. The two time constants $T = 2.0$ are obtained by numerical solution of Equation (2.11). The point $s(8.6) = 0.63$ is used to obtain the additional condition. Figure 2.10 shows the step

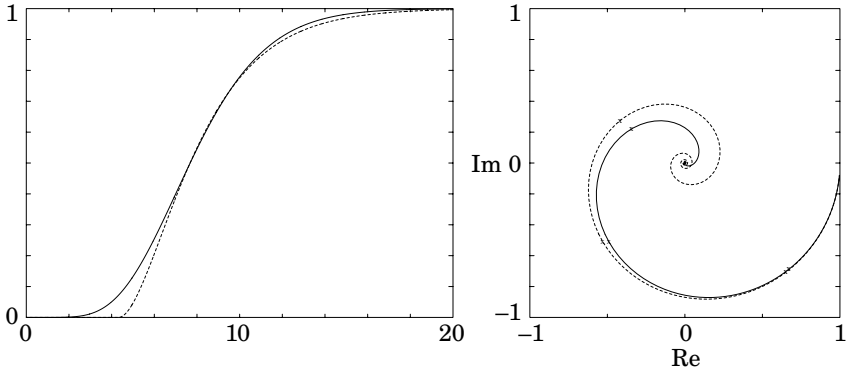

Figure 2.10 Step responses and Nyquist curves of the process $G(s) = 1/(s+1)^8$ (solid line) and the three-parameter model $G_{3c}(s)$ (dashed line).

responses of the true process model and $G_{3c}(s)$, as well as the Nyquist curves of the two transfer functions. The two step responses coincide at the 63% point. The model now has the S-shaped form because of the second-order model, and the fit is much better than the previous first-order models. □

Four-Parameter Models

An even better approximation may be obtained by the transfer function

$$G(s) = \frac{K}{(1+sT_1)(1+sT_2)}\, e^{-sL} \tag{2.12}$$

This model has four parameters: the gain K, the time constants T_1 and T_2, and the dead time L. The gain K can be determined from the steady-state value of the step response. The dead time L can also be obtained in the same way as for the three-parameter models either by drawing the tangent of maximum slope of $s(t)$ or by determining the time between the onset of the step and the time $s(t)$ has reached a few percent of its final value. The step response of the model (2.12) is

$$s(t) = K\left(1 + \frac{T_2 e^{-(t-L)/T_2} - T_1 e^{-(t-L)/T_1}}{T_1 - T_2}\right) \qquad T_1 \neq T_2 \tag{2.13}$$

The time constants (T_1) and (T_2) can be calculated from this expression by determining two points of the step response. The calculation does involve solution of transcendental equations. This must be done numerically.

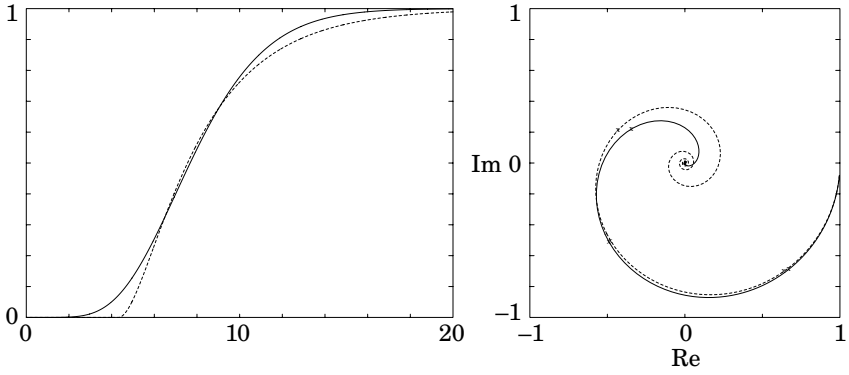

Figure 2.11 Step responses and Nyquist curves of the process $G(s) = 1/(s+1)^8$ (solid line) and the four-parameter model $G_{4a}(s)$ (dashed line).

EXAMPLE 2.4

A four-parameter model (2.12) of the process model (2.7) has been obtained in the following way. The gain $K = 1$ is determined from the steady-state values, and the dead time $L = 4.3$ is obtained from the largest slope, as in the previous examples. The time constants T_1 and T_2 are then obtained by numerically fitting the equation for the step response (2.13) to the values of the true step response at the 33% point and the 67% point. With $s(6.5) = 0.33$ and $s(8.9) = 0.67$, the time constants become $T_1 = 0.93$ and $T_2 = 3.2$. The transfer function is thus

$$G_{4a}(s) = \frac{1}{(1+0.93s)(1+3.2s)}\, e^{-4.3s}$$

Figure 2.11 shows the step responses of the true process model and $G_{4a}(s)$, as well as the Nyquist curves of the two transfer functions. Notice that the two step responses coincide at the 33% point and at the 67% point. □

In the previous example, gain K and dead time L were determined graphically from the step response, whereas time constants T_1 and T_2 were determined by numerical solution of the equation for the step response. There are several methods presented for a graphical determination of all four parameters of the model (2.12). These methods are useful when no computers are available for numerical solutions. Using computer optimization programs, however, often gives a better approximation than the graphical methods. This is illustrated in the following example.

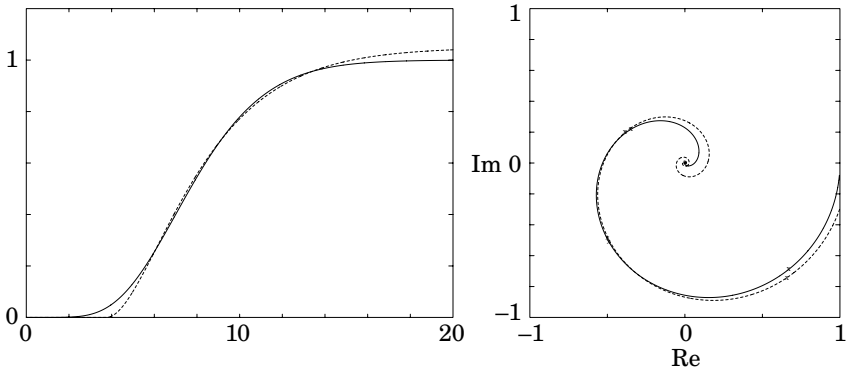

Figure 2.12 Step responses and Nyquist curves of the process $G(s) = 1/(s+1)^8$ (solid line) and the four-parameter model G_{4b} (dashed line).

EXAMPLE 2.5

The four-parameter model (2.12) has been fitted to the process model (2.7) using least squares optimization, where the aim was to obtain an accurate model in the third quadrant, i.e., where the phase shift is between $-90°$ and $-180°$. The following model was obtained.

$$G_{4b}(s) = \frac{1.05}{(1+2.39s)^2}\, e^{-3.75s}$$

Figure 2.12 shows the step responses of the true process model and $G_{4b}(s)$, as well as the Nyquist curves of the transfer functions. □

Models for Integrating Systems

There are some process control systems where the dynamics contain integration or very long time constants. Such systems will not reach a steady state under open-loop conditions. They are sometimes called systems without self regulation. For PID tuning it is useful to treat such systems separately.

Impulse Responses

For a system with integral action a steady state will not be achieved when the input signal is a step, since the output will asymptotically change at a constant rate. There will be, however, a steady state when the input is an impulse. To determine the dynamics we can, therefore, apply a short pulse to the process. After normalizing the response by dividing with the pulse area, we then get a step response that can

be modeled using the methods we have just discussed. The transfer function of a system with integral action is then obtained simply by multiplying the transfer function by $1/s$. We illustrate the procedure with an example.

EXAMPLE 2.6

Assume that a square pulse with unit height and duration τ has been applied to a process and that the model

$$G_1(s) = \frac{K}{1+sT}\, e^{-sL}$$

has been fitted to the response as described in Example 2.2. The transfer function of the process is then

$$G(s) = \frac{1}{s\tau} G_1(s) = \frac{K}{s\tau(1+sT)}\, e^{-sL} \qquad \square$$

Step Responses

Models based on step responses can also be applied to processes with integral action. One possibility is to calculate the derivative of the step response and apply the impulse response method that was just discussed.

The two-parameter model

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

that was used to model stable processes previously in this section can also be applied to integrating processes. This model gives a bad description of stable processes at high frequencies, but for integrating processes the low frequency behavior is well captured by the model.

A more sophisticated model that gives a better approximation at higher frequencies is given by the transfer function

$$G(s) = \frac{K}{s(1+sT)} e^{-sL} \tag{2.14}$$

The model is characterized by three parameters: the velocity gain K, the time constant T, and the dead time L. The step response of the model (2.14) is

$$s(t) = K\left(t - L - T\left(1 - e^{-(t-L)/T}\right)\right) \tag{2.15}$$

The gain K and the average residence time $T_{ar} = L + T$ can be determined graphically as shown in Figure 2.13.

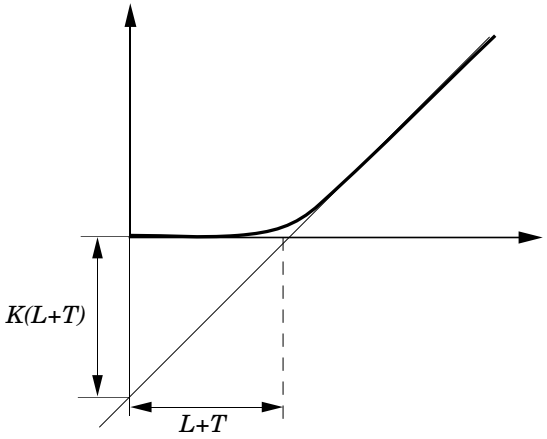

Figure 2.13 Graphical determination of a three-parameter model for an integrating process.

The dead time L and the time constant T can be determined by fitting Equation (2.15) to one point of the step response. A suitable point is

$$s(L+T) = KTe^{-1}$$

which gives

$$T = \frac{s(L+T)}{K}\, e^{1}$$

Models for Oscillatory Systems

Oscillatory systems with step responses, as shown in Figures 2.5B and D, can be crudely approximated by the two-parameter model (2.6), but this model will not capture the oscillations. None of the three- or four-parameter models presented above is suitable either. A three-parameter model that describes the oscillations is given by the transfer function

$$G(s) = \frac{K\omega^2}{s^2 + 2\zeta\omega s + \omega^2} \tag{2.16}$$

This model has three parameters: the static gain K, the natural frequency ω, and the relative damping ζ. These parameters can be determined approximately from the step response as indicated in Figure 2.14. The period of the oscillation T_p and the decay ratio d are first determined. Parameters ω and ζ are related to T_p and d as follows.

$$d = e^{-2\zeta\pi/\sqrt{1-\zeta^2}} \qquad T_p = \frac{2\pi}{\omega\sqrt{1-\zeta^2}}$$

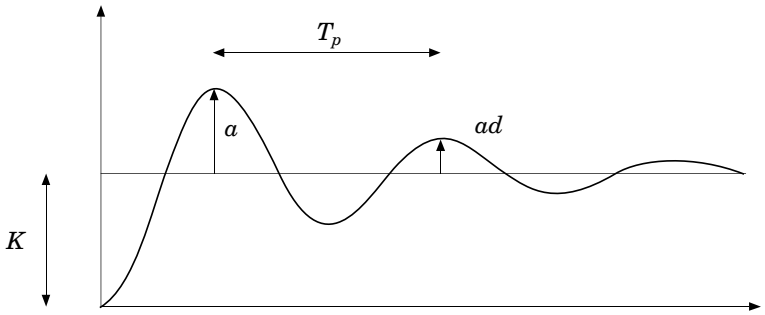

Figure 2.14 Graphical determination of mathematical models for systems with an oscillatory step response.

or

$$\zeta = \frac{1}{\sqrt{1 + (2\pi/\log d)^2}} \qquad \omega = \frac{2\pi}{T_p\sqrt{1-\zeta^2}} \tag{2.17}$$

A time delay can also be added to the model (2.16) and determined in the same way as for the previous models, e.g., by drawing the tangent of maximum slope or determining the time between the onset of the step and the time the step response has reached a few percent of its final value.

2.5 Methods of Moments

All average residence time was determined based on calculation of an area. All other methods discussed in Section 2.4 were based on evaluation of the step response at single points only. Such methods are quite sensitive to measurement noise. In this section we will discuss methods that are based on integrals of the step response.

Area Methods

We will first discuss a method that is based on area calculations. Static gain K and average residence time T_{ar} are first determined as in Figure 2.6. The area A_1 under the step response up to time T_{ar} is then determined. For a system having the transfer function

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

we have

$$A_1 = \int_0^{T_{ar}} s(t)dt = \int_0^{T} K(1 - e^{-t/T})dt = KTe^{-1}$$

The time constant is thus given by

$$T = \frac{eA_1}{K} \tag{2.18}$$

The dead time is then given by

$$L = T_{ar} - T = \frac{A_0}{K} - \frac{eA_1}{K} \tag{2.19}$$

With this method parameters L and T are both determined from computations of areas. The method is illustrated by the following example.

EXAMPLE 2.7

The method based on area determination has been applied to the process model (2.7). Static gain K is first determined from the stationary values to $K = 1$. Area A_0 is then determined to 8.0 providing the average residence time $T_{ar} = 8$. Area A_1 can be determined by integrating the step response up to time T_{ar} to $A_1 = 1.1$. From Equation (2.18), time T can be calculated to $T = 3.0$, and finally Equation (2.19) gives $L = 5.0$. To summarize, the method based on area determination gives the following three-parameter model

$$G_{3d}(s) = \frac{1}{1 + 3.0s} e^{-5.0s}$$

Figure 2.15 shows the step responses of the true process model and $G_{3d}(s)$, as well as the Nyquist curves of the two transfer functions. □

The same idea can easily be applied to a system with the transfer function

$$G(s) = \frac{K}{(1 + sT)^2} e^{-sL} \tag{2.20}$$

Parameters K and residence time T_{ar} are determined as before. In this case we have

$$T_{ar} = L + 2T$$

The area A_1 under the step response up to time T_{ar} is then determined. For a system having transfer function (2.20) we have

$$A_1 = \int_0^{T_{ar}} s(t)dt = \int_0^{2T} K\left(1 - e^{-t/T} - \frac{t}{T}e^{-t/T}\right)dt = 4KTe^{-2}$$

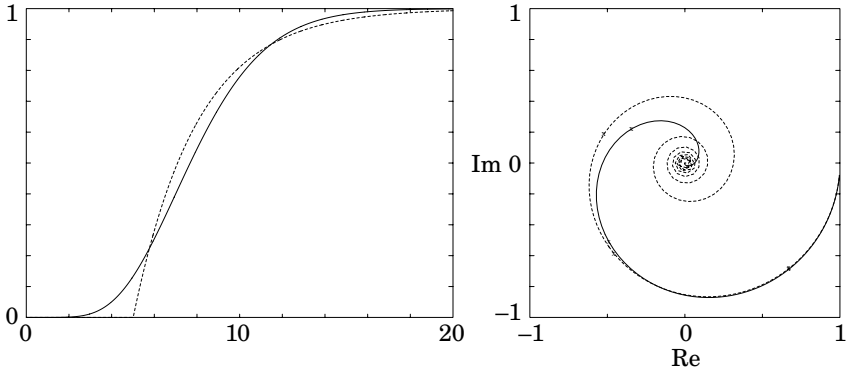

Figure 2.15 Step responses and Nyquist curves of the process $G(s) = 1/(s+1)^8$ (solid line) and the three-parameter model $G_{3d}(s)$ (dashed line).

The time constant is thus given by

$$T = \frac{A_1 e^2}{4K} \tag{2.21}$$

and the dead time is

$$L = T_{ar} - 2T = \frac{A_0}{K} - \frac{A_1 e^2}{2K} \tag{2.22}$$

The following example illustrates the properties of the method.

EXAMPLE 2.8

The three-parameter model (2.20) has been fitted to the process model (2.7) using the method of area determination. Static gain K is determined from the stationary values to $K = 1$. The area A_0 is 8.0, which gives the average residence time $T_{ar} = 8.0$. Furthermore the area A_1 is 1.1 and Equation (2.21) then gives $T = 2.0$. Equation (2.22) finally gives $L = 4.0$ and the model becomes

$$G_{3e}(s) = \frac{1}{(1+2.0s)^2} e^{-4.0s}$$

Figure 2.16 shows the step responses of the true process model and $G_{3e}(s)$, as well as the Nyquist curves of the two transfer functions. □

The methods based on area determination are less sensitive to high-frequency disturbances than the previous methods, where the model is determined from only a few values of the step response. On the other hand, they are more sensitive to low-frequency disturbances such as a change in static load.

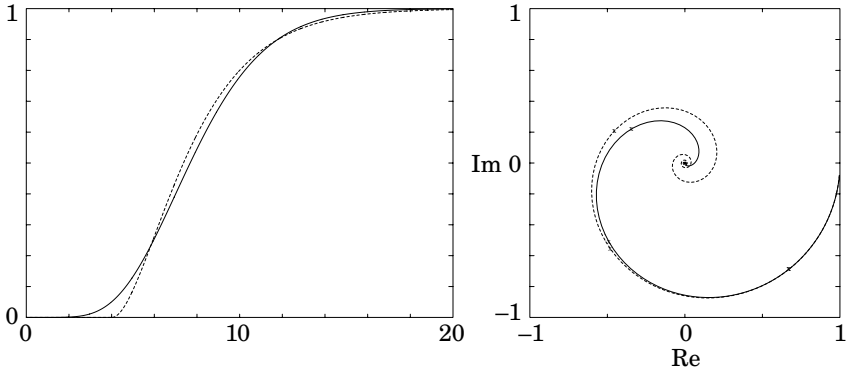

Figure 2.16 Step responses and Nyquist curves of the process $G(s) = 1/(s+1)^8$ (solid line) and the three-parameter model G_{3e}(dashed line).

The Method of Moments

A drawback with the area methods is that they require a storage of the step response. Area A_1 cannot be computed until area A_0 is determined. Therefore, some alternative methods that are also based on integration will be considered.

Let $h(t)$ be an impulse response and $G(s)$ the corresponding transfer function. The functions are related through

$$G(s) = \int_0^\infty e^{-st} h(t) dt$$

Taking derivatives with respect to s gives

$$\frac{d^n G(s)}{ds^n} = G^{(n)}(s) = (-1)^n \int_0^\infty e^{-st} t^n h(t) dt$$

Hence,

$$G^{(n)}(0) = (-1)^n \int_0^\infty t^n h(t) dt \tag{2.23}$$

The values of the transfer function and its derivatives at $s = 0$ can thus be determined from integrals of the impulse response.

The Average Residence Time

The impulse response is positive for systems with monotone step responses. It can be interpreted as the density function of a probability

distribution if it is normalized as follows:

$$f(t) = \frac{h(t)}{\int_0^\infty h(t)dt}$$

The quantity $f(t)dt$ can then be interpreted as the probability that an impulse entering the system at time 0 will leave at time t. The average residence time is then

$$T_{ar} = \int_0^\infty t f(t)dt = \frac{\int_0^\infty th(t)dt}{\int_0^\infty h(t)dt} \tag{2.24}$$

Introduce

$$g(t) = s(\infty) - s(t)$$

where $s(t)$ is the unit step response. Then

$$\frac{dg(t)}{dt} = -h(t)$$

It follows that

$$\int_0^\infty th(t)dt = \Big[-tg(t)\Big]_0^\infty + \int_0^\infty g(t)dt$$

The first term of the right-hand side is zero if $g(t)$ goes to zero at least as fast as $t^{1+\epsilon}$ for large t. The average residence time can thus also be written as

$$T_{ar} = \frac{\int_0^\infty (s(\infty) - s(t))dt}{s(\infty)}$$

which is the definition used previously.

Equation (2.23) gives a convenient way to determine parameters of different models by computing the moments. This will be illustrated by some examples.

A Three-Parameter Model

Consider the transfer function

$$G(s) = \frac{K}{1+sT} e^{-sL} \tag{2.25}$$

It follows that

$$K = G(0) = \int_0^\infty h(t)dt \tag{2.26}$$

Taking logarithms of Equation (2.25) gives

$$\log G(s) = \log K - sL - \log(1 + sT)$$

Differentiating this expression gives

$$\frac{G'(s)}{G(s)} = -L - \frac{T}{1 + sT}$$

$$\frac{G''(s)}{G(s)} - \left(\frac{G'(s)}{G(s)}\right)^2 = \frac{T^2}{(1 + sT)^2}$$

Hence

$$T_{ar} = -\frac{G'(0)}{G(0)} = L + T = \frac{\int_0^\infty th(t)dt}{\int_0^\infty h(t)dt}$$

$$T^2 = \frac{G''(0)}{G(0)} - T_{ar}^2 = \frac{\int_0^\infty t^2 h(t)dt}{\int_0^\infty h(t)dt} - T_{ar}^2 \tag{2.27}$$

Gain K is thus given by Equation (2.26) and average residence time T_{ar} and time constant T by Equation (2.27). The dead time L can then be computed to

$$L = T_{ar} - T$$

It has thus been shown that the parameters of the model can be obtained from the first two moments of the impulse response. We illustrate the procedure with an example.

EXAMPLE 2.9

Consider the process model

$$G(s) = \frac{1}{(s + 1)^8}$$

The first two derivatives with respect to s become

$$G'(s) = -\frac{8}{(s + 1)^9} \qquad G''(s) = \frac{72}{(s + 1)^{10}}$$

Hence $G(0) = 1$, $G'(0) = -8$, and $G''(0) = 72$. Equations (2.26) and (2.27) now give

$$K = 1$$

$$T_{ar} = 8$$

$$T^2 = 72 - 64 = 8$$

We thus find $T = 2\sqrt{2} \approx 2.8$ and $L = 8 - 2\sqrt{2} \approx 5.2$. This result can be compared with the previous methods in Examples 2.2 and 2.7. □

Another Three-Parameter Model

The method of moments will now be applied to determine the parameters of the transfer function

$$G(s) = \frac{K}{(1+sT)^2}\, e^{-sL}$$

We have

$$\log G(s) = \log K - sL - 2\log(1+sT)$$

Hence

$$\frac{G'(s)}{G(s)} = -L - \frac{2T}{1+sT}$$

$$\frac{G''(s)}{G(s)} - \left(\frac{G'(s)}{G(s)}\right)^2 = \frac{2T^2}{(1+sT)^2}$$

Hence

$$K = G(0) = \int_0^\infty h(t)dt$$

$$T_{ar} = -\frac{G'(0)}{G(0)} = L + 2T = \frac{\int_0^\infty th(t)dt}{\int_0^\infty h(t)dt} \tag{2.28}$$

$$T^2 = \frac{G''(0)}{2G(0)} - \frac{1}{2}T_{ar}^2 = \frac{\int_0^\infty t^2 h(t)dt}{2\int_0^\infty h(t)dt} - \frac{1}{2}T_{ar}^2$$

We illustrate the method with an example.

EXAMPLE 2.10

Consider the process model (2.7). It follows from the previous example that $G(0) = 1$, $G'(0) = -8$, and $G''(0) = 72$. We thus find $K = 1$, $T_{ar} = 8$, $T = 2$ and $L = 4$. This is the same model as the one obtained in Example 2.8. □

Other Input Signals

From a practical point of view it is a drawback to have methods that require special input signals. The method of moments can be applied to any signal provided that the system is initially at rest.

Let $U(s)$ and $Y(s)$ be the Laplace transforms of an arbitrary input and the corresponding output, respectively. Taking derivatives we get

$$\begin{aligned} Y(s) &= G(s)U(s) \\ Y'(s) &= G'(s)U(s) + G(s)U'(s) \\ Y''(s) &= G''(s)U(s) + 2G'(s)U'(s) + G(s)U''(s) \\ &\text{etc.} \end{aligned}$$

Hence,

$$\begin{aligned} Y(0) &= G(0)U(0) \\ Y'(0) &= G'(0)U(0) + G(0)U'(0) \\ Y''(0) &= G''(0)U(0) + 2G'(0)U'(0) + G(0)U''(0) \\ &\text{etc.} \end{aligned} \tag{2.29}$$

The transfer function $G(0)$ and its derivatives can thus be calculated from experiments with arbitrary inputs by calculating the following moments of the input and output

$$U^{(n)}(0) = (-1)^n \int_0^\infty t^n u(t)dt$$

$$Y^{(n)}(0) = (-1)^n \int_0^\infty t^n y(t)dt$$

and using Equation (2.29).

By using these formulas it is possible to calculate $G^{(n)}(0)$ for any signals for which the moments

$$u_n = \int_0^\infty t^n u(t)dt$$

and

$$y_n = \int_0^\infty t^n y(t)dt$$

exist. This means that the signals must decay sufficiently fast.

A typical case where the method can be used is when an experiment is performed in a closed loop with a pulse-like perturbation signal on the process input.

Weighted Moments

The method just discussed cannot be used if the signals do not go to zero or, equivalently, to *a priori* known mean values that can be subtracted in the calculations of moments, because the moments will then be infinite. There is, however, a simple modification that can be used in this case. It follows from the definition of the Laplace transform that

$$Y^{(n)}(s) = \frac{d^n Y(s)}{ds^n} = (-1)^n \int_0^\infty e^{-st} t^n y(t) dt$$

The weighted moments

$$y_n = \int_0^\infty t^n e^{-\alpha t} y(t) dt = (-1)^n Y^{(n)}(\alpha)$$

will exist provided that $y(t)$ does not grow faster than $e^{\alpha t}$ for large t. By computing y_n and the analogously defined moment u_n, we can compute $Y^{(n)}(\alpha)$ and $U^{(n)}(\alpha)$, and thus also $G^{(n)}(\alpha)$.

A Three-Parameter Model

Consider a system with the transfer function

$$G(s) = \frac{K}{1 + sT} e^{-sL} \tag{2.30}$$

We have

$$\log G(s) = \log K - sL - \log (1 + sT)$$

Hence

$$\frac{G'(s)}{G(s)} = -L - \frac{T}{1 + sT}$$

$$\frac{G''(s)}{G(s)} - \left(\frac{G'(s)}{G(s)} \right)^2 = \frac{T^2}{(1 + sT)^2}$$

Thus we get

$$\frac{T^2}{(1 + \alpha T)^2} = \frac{G''(\alpha)}{G(\alpha)} - \left(\frac{G'(\alpha)}{G(\alpha)} \right)^2 = a^2 \tag{2.31}$$

Hence,

$$\begin{aligned} T &= \frac{a}{1 - \alpha a} \\ L &= -\frac{G'(\alpha)}{G(\alpha)} - a \end{aligned} \tag{2.32}$$

The average residence time thus becomes

$$T_{ar} = L + T = -\frac{G'(\alpha)}{G(\alpha)} + \frac{\alpha a^2}{1 - \alpha a}$$

Furthermore the static gain is given by

$$K = (1 + \alpha T)G(\alpha)e^{\alpha L} \tag{2.33}$$

The formulas are illustrated by an example.

EXAMPLE 2.11

Consider a system with the transfer function

$$G(s) = \frac{1}{(s+1)^8}$$

We have

$$G(\alpha) = \frac{1}{(1+\alpha)^8} \qquad G'(\alpha) = \frac{-8}{(1+\alpha)^9} \qquad G''(\alpha) = \frac{72}{(1+\alpha)^{10}}$$

Computing the derivatives at the origin from the first terms in the Taylor series expansion gives

$$G(0) \approx \frac{1}{(1+\alpha)^8} + \frac{8\alpha}{(1+\alpha)^9} = \frac{1+9\alpha}{(1+\alpha)^9}$$

$$G'(0) \approx -\frac{8}{(1+\alpha)^9} - \frac{72\alpha}{(1+\alpha)^{10}} = -\frac{8(1+10\alpha)}{(1+\alpha)^{10}}$$

The estimate of the average residence time becomes

$$\hat{T}_{ar} = -\frac{G'(0)}{G(0)} \approx \frac{8(1+10\alpha)}{(1+\alpha)(1+9\alpha)} = \frac{8(1+10\alpha)}{1+10\alpha+9\alpha^2}$$

From these expressions it follows that α must be small in order to give reasonably good approximations. To discuss the values of α, it is reasonable to normalize and consider αT_{ar}. In this case, $T_{ar} = 8$. With $\alpha T_{ar} = 1$ we get $G(0) = 0.74$, $G'(0) = -5.54$, and $\hat{T}_{ar} = 7.53$. With $\alpha T_{ar} = 0.5$ we get $G(0) = 0.91$, $G'(0) = -7.1$, and $\hat{T}_{ar} = 7.83$, giving errors in the range of 10%. With $\alpha T_{ar} = 0.2$ we get $G(0) = 0.98$, $G'(0) = -7.81$, and $\hat{T}_{ar} = 7.96$.

It follows from Equation (2.31) that

$$a = \sqrt{\frac{72}{(1+\alpha)^2} - \frac{64}{(1+\alpha)^2}} = \frac{2\sqrt{2}}{1+\alpha}$$

It follows from Equations (2.32) and (2.33) that

$$T = \frac{2\sqrt{2}}{1 + (1 - 2\sqrt{2})\alpha}$$

$$L = \frac{8 - 2\sqrt{2}}{1 + \alpha}$$

$$K = \frac{1}{(1+\alpha)^7(1 + (1 - 2\sqrt{2})\alpha)}\, e^{(8-2\sqrt{2})\alpha/(1+\alpha)}$$

The average residence time becomes

$$\hat{T}_{ar} = T + L = 8\, \frac{1 + (2 - 2\sqrt{2})\alpha}{1 + (2 - 2\sqrt{2})\alpha + (1 - 2\sqrt{2})\alpha^2}$$

With $\alpha T_{ar} = 1$, 0.5, and 0.2, we get the estimates $\hat{T}_{ar} = 8.26$, $\hat{T}_{ar} = 8.06$, and $\hat{T}_{ar} = 8.01$, respectively. This method of estimating the average residence time gives slightly better results than the extrapolation method. □

The example shows that we can obtain reasonable estimates of the model parameters and the average residence time by using weighted moments. It also seems reasonable to choose parameter α so that αT_{ar} is in the range of 0.2 to 1. The best results are obtained for a small value of α. There is, however, an advantage in using larger values of α because there is then a less risk for disturbances to enter the system.

2.6 Frequency Responses

Two methods for determining interesting points on the Nyquist curve are presented below. Both are based on the idea of using feedback to generate sinusoids having the appropriate frequency.

The Ziegler-Nichols Frequency Response Method

Ziegler and Nichols have provided a method for determining the ultimate point on the Nyquist curve experimentally. The method is based on the observation that many systems can be made unstable under proportional feedback by choosing sufficiently high gain in the proportional feedback (see Figure 2.17). Assume that the gain is adjusted so that the process is at the stability boundary. The control signal and the process output are then sinusoids with a phase shift of $-180°$ (see

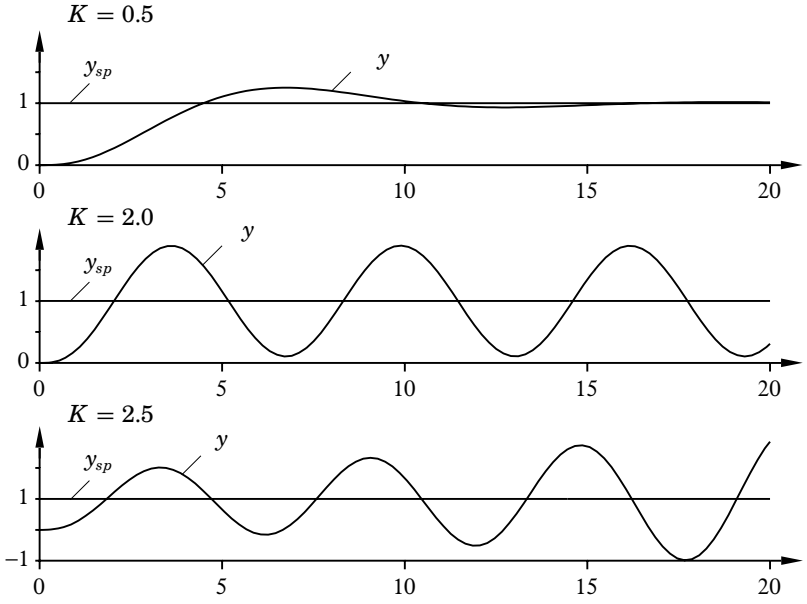

Figure 2.17 Setpoint y_{sp} and process output y for a closed-loop system with proportional feedback. The figure shows responses for three values of controller gain K.

Figure 2.18). Because of the proportional feedback they are related by

$$u = -Ky$$

For simplicity it has been assumed that the setpoint is $y_{sp} = 0$. Since the gain around the loop must be unity to maintain an oscillation, we have

$$K_u G(i\omega_u) = -1$$

where the gain, which brings the system to the stability limit, is called

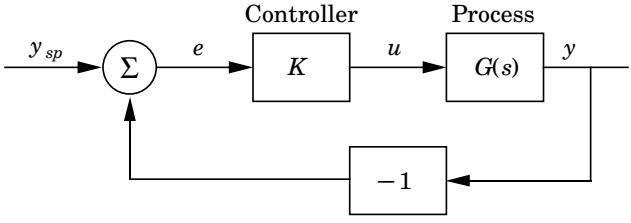

Figure 2.18 Block diagram of a closed-loop system under proportional feedback.

Table 2.1 Relations between gain ratio κ and normalized dead time τ for processes with the transfer functions $G(s) = 1/(s+1)^n$.

n	2	3	4	8
τ	0.15	0.25	0.35	0.55
κ	0	0.125	0.25	0.53

the ultimate gain (K_u). It follows from the above equation that

$$G(i\omega_u) = -\frac{1}{K_u} \tag{2.34}$$

Several design methods based only on the knowledge of $G(i\omega_u)$ are given in Chapter 4. It is convenient to introduce the gain ratio,

$$\kappa = \left| \frac{G(i\omega_u)}{G(0)} \right| \tag{2.35}$$

i.e., the gain at the ultimate frequency divided by the static gain. This parameter is an indicator of how difficult it is to control the process. Processes with a small κ are easy to control. The difficulty increases with increasing κ.

Parameter κ is also related to the normalized dead time τ, which was defined in Equation (2.9). For processes described by the transfer function (2.8) parameters τ and κ are related in the following way:

$$\tau = \frac{\pi - \arctan\sqrt{1/\kappa^2 - 1}}{\pi - \arctan\sqrt{1/\kappa^2 - 1} + \sqrt{1/\kappa^2 - 1}}$$

This relation is close to linear, it gives $\tau = 0$ for $\kappa = 0$ and $\tau = 1$ for $\kappa = 1$. For small values of κ it can be approximated by $\tau = 1.6\kappa$. This is illustrated in the following example.

EXAMPLE 2.12

To illustrate the relation between the parameters κ and τ, we give their values for systems with the transfer functions

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

The results are presented in Table 2.1. For small values of n, both κ and τ are small. These processes are easy to control. For large values of n, both κ and τ approach 1. These processes are difficult to control. □

The Ziegler-Nichols frequency response method has some advantages. It is based on a simple experiment, and the process itself is

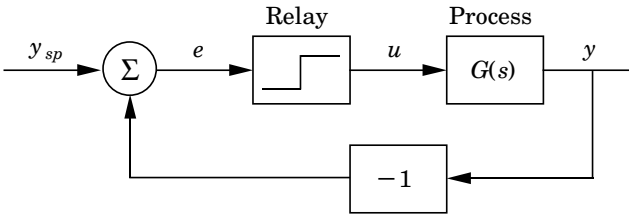

Figure 2.19 Block diagram of a process under relay feedback.

used to find the ultimate frequency. It is, however, difficult to automate this experiment or perform it in such a way that the amplitude of the oscillation is kept under control. Operating the process near instability is also dangerous and may need management authorization in an industrial plant. It is difficult to use this method for automatic tuning. An alternative method for automatic determination of specific points on the Nyquist curve is suggested below.

Relay Feedback

An alternative method to determine interesting points on the Nyquist curve is based on the observation that the appropriate oscillation can be generated by relay feedback. The system is thus connected as shown in Figure 2.19. For many systems there will then be an oscillation (as shown in Figure 2.20) where the control signal is a square wave and the process output is close to a sinusoid. Notice that the process input and output have opposite phase.

To explain how the system works, assume that the relay output is expanded in a Fourier series and that the process attenuates higher

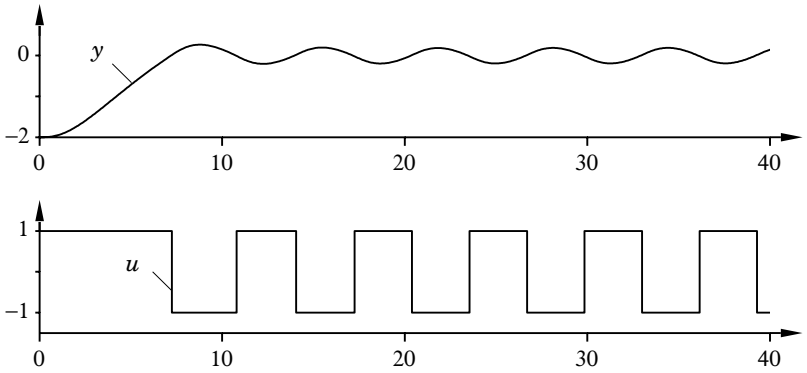

Figure 2.20 Relay output u and process output y for a system under relay feedback.

harmonics effectively. It is then sufficient to consider the first harmonic component of the input only. The input and the output then have opposite phase, which means that the frequency of the oscillation is the ultimate frequency. If d is the relay amplitude, the first harmonic of the square wave has amplitude $4d/\pi$. Let a be the amplitude of the oscillation in the process output. Then,

$$G(i\omega_u) = -\frac{\pi a}{4d} \tag{2.36}$$

Notice that the relay experiment is easily automated. Since the amplitude of the oscillation is proportional to the relay output, it is easy to control it by adjusting the relay output. Also notice in Figure 2.20 that a stable oscillation is established very quickly. The amplitude and the period can be determined after about 20 s only, in spite of the fact that the system is started so far from the equilibrium that it takes about 8 s to reach the correct level. The average residence time of the system is 12 s, which means that it would take about 40 s for a step response to reach steady state.

Describing Function Analysis

The intuitive discussion about relay oscillations can be dealt with more quantitatively using a technique called the describing function method. This is an approximate method that can be used to determine if there will be an oscillation in a nonlinear feedback system that is composed of a linear element and a static nonlinearity. To determine conditions for oscillation, the nonlinear block is described by a gain, $N(a)$, which depends on signal amplitude a at the input of the nonlinearity. This gain, which describes how a sinusoid of amplitude a propagates through the system, is called the describing function. If the process has the transfer function $G(i\omega)$, the condition

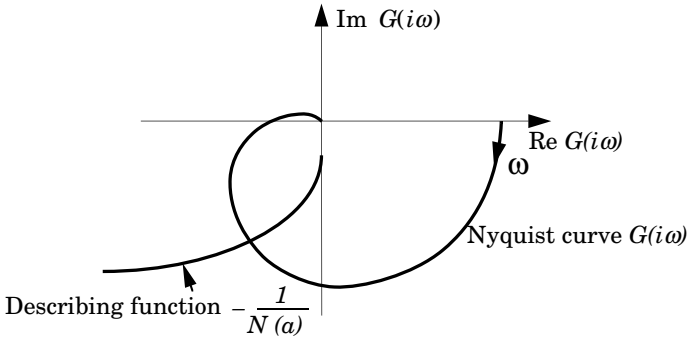

Figure 2.21 Determination of possible oscillations using the describing function method.

for oscillation is simply given by

$$N(a)G(i\omega) = -1 \tag{2.37}$$

This equation is obtained by requiring that a sine wave with frequency ω should propagate around the feedback loop with the same amplitude and phase. The equation gives two equations for determining a and ω, since N and G may be complex numbers. The equation can be solved graphically by plotting $-1/N(a)$ in the Nyquist diagram (as in Figure 2.21) together with the Nyquist curve $G(i\omega)$ of the linear system. An oscillation may occur if there is an intersection between the two curves. The amplitude and the frequency of the oscillation are the same as the parameters of the two curves at the intersection point. Therefore, measuring the amplitude and the period of the oscillation, the position of one point of the Nyquist curve can be determined.

The describing function, $N(a)$, for a relay is given by

$$N(a) = \frac{4d}{\pi a} \tag{2.38}$$

Since this function is real, an oscillation may occur if the Nyquist curve intersects the negative real axis. This explains why the experiment with relay feedback gives the point where the Nyquist curve intersects the negative real axis.

A Relay with Hysteresis

There are advantages in having a relay with hysteresis instead of a pure relay. With an ordinary relay, a small amount of noise can make the relay switch randomly. By introducing hysteresis, the noise must be larger than the hysteresis width to make the relay switch. See Figure 2.22. The describing function approach will be used to investigate the oscillations obtained. The negative inverse of the describing func-

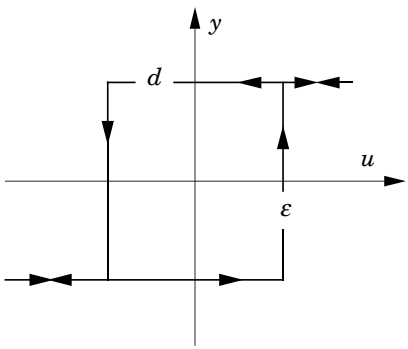

Figure 2.22 Output y from a relay with hysteresis with input u.

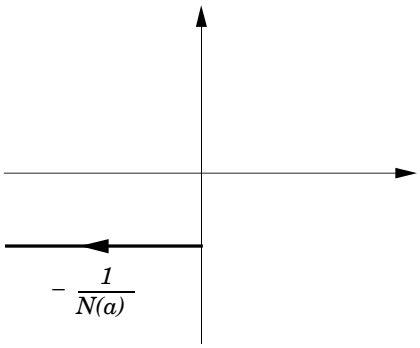

Figure 2.23 The negative reciprocal of the describing function $N(a)$ for a relay with hysteresis.

tion of such a relay is

$$-\frac{1}{N(a)} = -\frac{\pi}{4d}\sqrt{a^2 - \epsilon^2} - i\,\frac{\pi\epsilon}{4d} \tag{2.39}$$

where d is the relay amplitude and ϵ is the hysteresis width. This function can be represented as a straight line parallel to the real axis, in the complex plane (see Figure 2.23).

By choosing the relation between ϵ and d, it is therefore possible to determine a point on the Nyquist curve with a specified imaginary part. Several points on the Nyquist curve can be obtained by repeating the experiment with different relations between ϵ and d. It is easy to control the amplitude of the limit cycle to a desired level by a proper choice of the relay amplitude.

Other Uses of Relay Feedback

A slight modification of the experiment shown in Figure 2.19 gives other frequencies of interest. Figure 2.24 shows an experiment that gives the frequency ω_{90}, i.e. the frequency where the process has a phase lag of 90°. Notice that there are two different versions of the experiment depending on the order in which the integrator and the relay are connected.

Closed Loop Experiments

Relay feedback can also be applied to closed-loop systems. Figure 2.25 shows an experiment that can be used to determine the amplitude margin on-line. Let G_ℓ be the loop transfer function, i.e., the combined transfer function of the controller and the process. The closed-loop

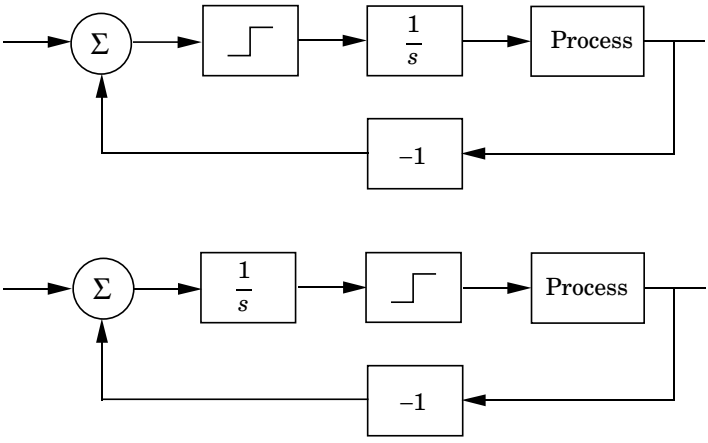

Figure 2.24 Using relay feedback to determine the frequency ω_{90}.

transfer function is then

$$G_{cl}(s) = \frac{G_\ell(s)}{1 + G_\ell(s)} \tag{2.40}$$

The experiment with relay feedback then gives an oscillation with the frequency such that the phase lag of $G_{cl}(i\omega)$ is 180°. It then follows from Equation (2.40) that this is also the frequency where $G_\ell(i\omega)$ has a phase lag of 180°, i.e., the ultimate frequency. If m is the magnitude of G_{cl} at that frequency, we find that an estimate of the amplitude margin of the closed-loop system is given by

$$\hat{A}_m = \frac{m}{1 - m}$$

If the relay has hysteresis, a conformal mapping argument shows that the experiment gives the frequency, where the loop transfer function

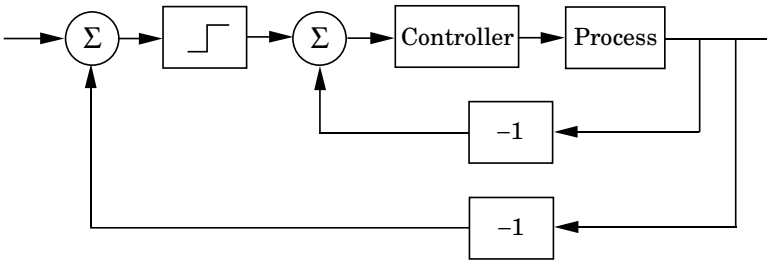

Figure 2.25 Using relay feedback to determine the amplitude margin of the closed-loop system.

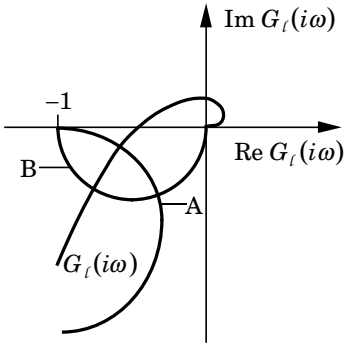

Figure 2.26 Experiments with relay feedback give the points where the curve $G_\ell(i\omega)$ intersects the circles.

intersects part of the circle,

$$\left| G_\ell(i\omega) - 1 + i\,\frac{1}{2a} \right| = \frac{1}{2a}$$

which is shown as curve A in in Figure 2.26. By introducing an integrator in series with the relay, the frequency where $G_{cl}(i\omega)$ has a phase lag of 90° is obtained. This occurs for loop transfer functions G_ℓ with the property

$$\arg \frac{G_\ell}{1 + G_\ell} = \arg G_\ell - \arg (1 + G_\ell) = \frac{\pi}{2}$$

This corresponds to the circle,

$$\left| G_\ell(i\omega) + \frac{1}{2} \right| = \frac{1}{2} \tag{2.41}$$

which is shown as curve B in Figure 2.26. The experiment will thus give the point where the loop transfer function G_ℓ of the closed-loop system intersects the circle given by Equation (2.41). Combining this result with the result from the experiment in Figure 2.24, it is also possible to approximately determine the maximum sensitivity M_s.

Many controllers use a two-degree-of-freedom configuration instead of pure error feedback. This is discussed in Chapter 3. This means that the control law is given by

$$U(s) = G_{ff}(s)Y_{sp}(s) - G_{fb}(s)Y(s)$$

The experiment shown in Figure 2.25 must then be modified by introducing a block with the transfer function G_{fb}/G_{ff} in series with the relay.

It has thus been demonstrated that several of the quantities needed to make an assessment of control performance can be obtained from experiments with relay feedback.

2.7 Parameter Estimation

A mathematical model of the process can also be obtained by fitting the parameters of a model to experimental data. For example, a model of the type given by Equation 2.8 can be obtained by adjusting the parameters so that they match observed input/output data. The advantage of such an approach is that any type of input/output data can be used. However, parameter estimation requires more computations than the methods discussed previously.

Parametric Models

Since the calculations will typically be made using a digital computer, the input/output data will typically be sampled. It is then convenient to operate with a discrete time model based on signals that are sampled periodically. Moreover, if the experimental data is also computer-generated, it is reasonable to assume that the input to the process is constant between the sampling instants. Let the sampling period be h. Assume that time delay L is less than h. The model (2.8) can then be described as

$$y(kh) = ay(kh-h) + b_1u(kh-h) + b_2u(kh-2h) \tag{2.42}$$

where

$$a = e^{-h/T}$$

$$b_1 = K\left(1 - e^{-(h-L)/T}\right)$$

$$b_2 = Ke^{-h/T}\left(e^{L/T} - 1\right)$$

For arbitrary time delays L, the model becomes instead

$$y(kh) = ay(kh-h) + b_1u(kh-nh) + b_2u(kh-nh-h) \tag{2.43}$$

where parameters a, b_1, and b_2 are given as above with $n = L \operatorname{div} h$ and $\tau = L \bmod h$ replacing L. The model can be given a convenient representation by introducing a shift operator q, defined by

$$qy(kh) = y(kh+h)$$

The model (2.43) can then be written as

$$q^n(q-a)y(kh) = (b_1q + b_2)u(kh)$$

If the complex variable z (similar to the Laplace transform variable s) is introduced, the process can also be described by the pulse transfer function:

$$H(z) = \frac{b_1z + b_2}{z^n(z-a)} \tag{2.44}$$

Notice that the transfer function is a ratio of two polynomials even if the corresponding physical process has time delays.

The discussion can be extended to systems of higher order, and the result is then an input/output relation of the form:

$$\begin{aligned} &y(kh) + a_1 y(kh-h) + \cdots + a_n y(kh-nh) \\ &= b_1 u(kh-h) + \cdots + b_n u(kh-nh) \end{aligned}$$

This equation can be written compactly as

$$A(q)y(kh) = B(q)u(kh) \tag{2.45}$$

where $A(q)$ and $B(q)$ are polynomials:

$$\begin{aligned} A(q) &= q^n + a_1 q^{n-1} + \cdots + a_n \\ B(q) &= b_1 q^{n-1} + b_2 q^{n-2} + \cdots + b_n \end{aligned}$$

The corresponding transfer function is then

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_1 z^{n-1} + b_2 z^{n-2} + \cdots + b_n}{z^n + a_1 z^{n-1} + \cdots + a_n}$$

Parameter Estimation

There are many ways to estimate the parameters of the discrete time model (Equation 2.45). A simple method is as follows. Assume that a sequence of input/output pairs ($\{u(kh), y(kh), k = 1, 2, \ldots, N\}$) have been observed. The parameters can then be determined in such a way that Equation (2.45) fits the data as well as possible in the least squares sense. The sum of the squares of the errors is

$$V(\theta) = \sum_{k=n+1}^{N} e^2(kh) \tag{2.46}$$

where

$$e(kh+nh) = A(q)y(kh) - B(q)u(kh), \qquad k = 1, \cdots, N-n$$

Notice that the error is linear in parameters a_i and b_i of the model and that the sum of squares of the errors is a quadratic function. This means that the minimum of the loss function can be computed analytically. Rather than showing the solution to the optimization problem, a convenient way of computing the parameters recursively is presented below.

Recursive Computations

In a tuning experiment, a new input/output pair is normally obtained in each sampling. It is then convenient to compute the parameter estimates recursively. All parameters are grouped together in the vector:

$$\theta = (a_1\, a_2 \ldots a_n\, b_1\, b_2 \ldots b_n)^T$$

Introduce the regression vector defined by

$$\varphi_{k-1} = (-y(kh-h) \ldots -y(kh-nh)\, u(kh-h) \ldots u(kh-nh))^T$$

The estimate can then be calculated recursively by

$$e_k = y(kh) - \varphi_{k-1}^T \theta_{k-1} \tag{2.47A}$$

$$P_k = P_{k-1} - \frac{P_{k-1}\varphi_{k-1}\varphi_{k-1}^T P_{k-1}}{1 + \varphi_{k-1}^T P_{k-1}\varphi_{k-1}} \tag{2.47B}$$

$$\theta_k = \theta_{k-1} + P_k \varphi_{k-1} e_k \tag{2.47C}$$

These equations have good physical interpretations. The new estimate θ_k is obtained by adding a correction term $P\varphi e$ to the old estimate θ_{k-1}. The correction term is a product of three quantities: P, φ, and e. The error e is the difference between the last measurement $y(kh)$ and the prediction $\varphi^T\theta$ of this measurement based on old estimates. Regression vector φ can be interpreted as the gradient of the error with respect to the parameters. This vector tells how the scalar error is distributed to give corrections in all parameters.

Equation (2.47B) may be interpreted as follows. Matrix P_k is proportional to the covariance matrix of the estimates; the last term in Equation (2.47B) is the reduction in uncertainty due to the last measurement.

The equations have to be initialized. The initial value of parameter vector θ can be chosen as the best initial guesses of the parameters. The initial value of matrix P is typically chosen as the identity matrix multiplied by a large number.

Computer Code

Recursive least squares estimation is an essential part of many schemes for automatic tuning. The following is a computer code that implements the algorithm.

```
{The recursive least squares algorithm}
  e=y
  for i=1 to 2*n do e=e-φ[i]*θ[i]
{Compute estimator gain}
  for i=1 to 2*n do
```

```
      begin
      s=0
      d=1
      for j=1 to 2*n do
        begin
        s=s+P[i,j]*φ[j]
        d=d+s*φ[j]
      end
      r[i]=s
    end
{Update estimates}
  for i=1 to 2*n do θ[i]=θ[i]+r[i]*e/d
{Update P matrix}
  for i=1 to 2*n do
    begin
    for j=i to 2*n do P[i,j]=P[i,j]-r[i]*r[j]/d
    for j=i+1 to 2*n do P[j,i]=P[i,j]
    end
{Update φ-vector}
   for i=1 to 2*n-1 do φ[2*n-i+1]=φ[2*n-i]
   φ[1]=-y
   φ[n+1]=u
```

The code description is given in "pidgin" Pascal, and it is assumed that the variables have been properly declared. There are many refinements to the algorithm; for instance, its numerical properties can be improved by using a so-called square root algorithm. It is also common practice to bandpass filter the signals before introducing them into the algorithm to get rid of static levels and high frequency disturbances. There are also many variations of the algorithm to discount past data. The code gives an indication of the type of algorithms that are used in recursive parameter estimation.

2.8 Disturbance Models

So far, we have only discussed modeling of process dynamics. Disturbances is another important side of the control problem. In fact, without disturbances and process uncertainty there would be no need for feedback. There is a special branch of control, stochastic control theory, that deals explicitly with disturbances. This has had little impact on tuning and design of PID controllers. For PID control, disturbances have mostly been considered indirectly, e.g., by introducing integral action. As our ambitions increase and we strive for control systems with improved performances it will be useful to consider disturbances explicitly. In this section, therefore, we will present some models that can be used for this purpose.

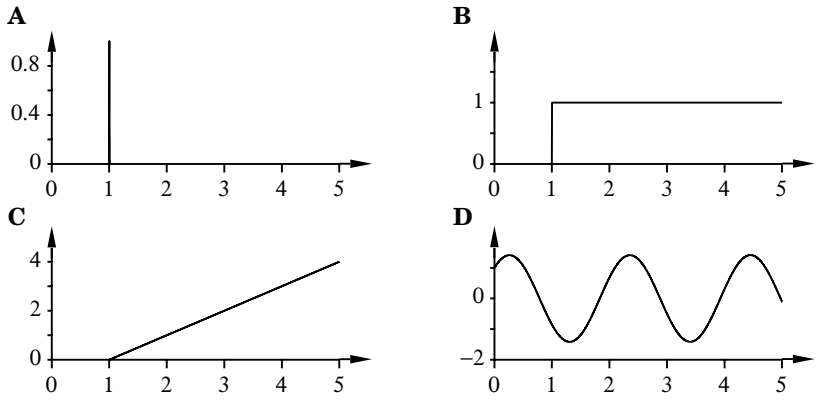

Figure 2.27 Prototype disturbances, A impulse, B step, C ramp, and D sinusoid.

There are some fundamental problems in dealing with disturbances. An inherent property of disturbances is that they cannot be predicted exactly. Most mathematical models, however, do have the property that they give signals that can be predicted exactly. Some care must be exercised when interpreting models and results.

Simple Models

Simple mathematical models were found to be very useful when dealing with process dynamics in the previous sections. We will now try to make models that also can be used to characterize disturbances.

Examples of some simple disturbances—impulse, step, ramp, and sinusoid—are given in Figure 2.27. The impulse is a mathematical idealization of a pulse whose duration is short in comparison with the time scale. The signals are essentially deterministic. The only uncertain elements in the impulse, step, and ramp are the times when they start and the signal amplitude. The uncertain elements of the sinusoid are frequency, amplitude, and phase.

More complex disturbances can be obtained by combining the simple disturbances, as shown in Figure 2.28. These disturbances are obtained by repeating a number of impulses, by combining steps and ramps, and by changing the amplitude and phase of the sinusoid.

Noise

There are also other types of disturbances that are much more irregular than the signals shown in Figure 2.28. Some examples are given in Figure 2.29. To characterize signals like the ones shown

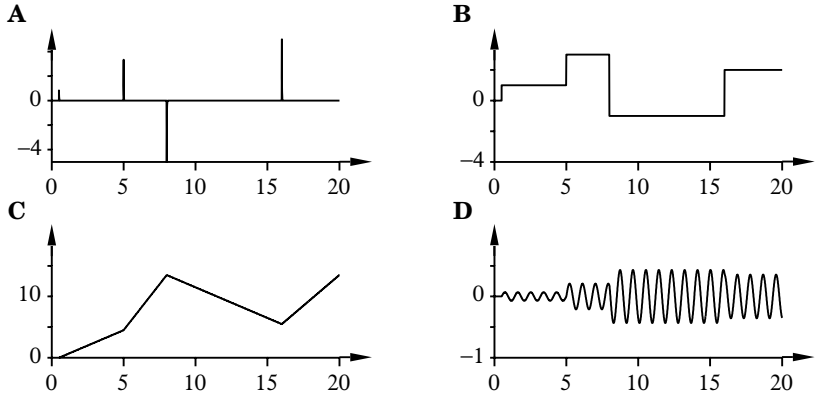

Figure 2.28 Disturbances that are obtained by combining the simple prototype disturbances.

in Figure 2.29, it is necessary to describe both the amplitude and the time characteristics. A distinction between stationary and nonstationary behavior must first be made. A signal is stationary if its behavior is essentially the same for all times. The amplitude properties of a stationary signal can be described by giving a histogram that tells the fraction of time when the signal has a given amplitude.

The mean value, the standard deviation or the variance, and the peak-to-peak values are simple ways to characterize the amplitude distribution. If the amplitude distribution is normal, the distribution is uniquely given by the mean value m and the standard deviation σ. The probability for the signal to be outside the 3σ limits is about 0.0026.

The time behavior of a stationary signal can be described by the spectral density function $\phi(\omega)$. This function characterizes the frequency content of a signal. The value

$$\frac{1}{2\pi}\phi(\omega)\Delta\omega$$

is the average energy of a signal in a narrow band of width $\Delta\omega$ centered around ω. The average energy is then

$$\sigma^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty}\phi(\omega)d\omega$$

A signal where $\phi(\omega)$ is constant is called white noise. Such a signal has its energy equally distributed among all frequencies.

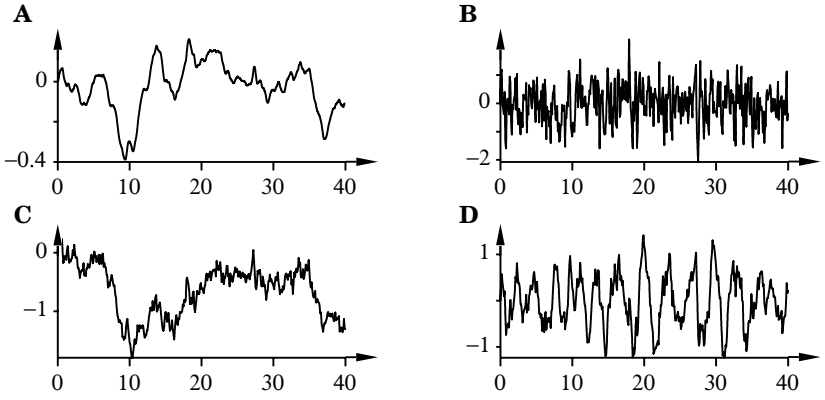

Figure 2.29 Examples of noise signals.

Measuring Noise Characteristics

The noise characteristics can be determined in several different ways. There are simple methods that can be used for crude estimates and more sophisticated methods that give more precise descriptions.

A simple way to estimate the amplitude characteristics is to measure the average

$$\bar{m} = \frac{1}{T}\int_0^T y(t)dt$$

and the mean square error

$$\bar{\sigma}^2 = \frac{1}{T}\int_0^T (y(t) - \bar{m})^2\, dt$$

To evaluate the integrals it is necessary to know a reasonable value of T, which requires knowledge about the time scale. An alternative is to compute the peak-to-peak value y_{pp}. The standard deviation can then be estimated as

$$\bar{\sigma} = \frac{1}{6}\, y_{pp}$$

Notice that it is also necessary to know the time scales in order to determine the time interval over which the peak-to-peak value is computed.

The energy in a given frequency band can be determined by computing the mean square average or the peak-to-peak value of a filtered signal.

Useful information about the frequency content in a signal can also be determined from the zero crossings. For a stationary signal

the average number of zero crossings per second can be determined from

$$N = \frac{1}{\pi}\left(\frac{\int_{-\infty}^{\infty} \omega^2 \phi(\omega) d\omega}{\int_{-\infty}^{\infty} \phi(\omega) d\omega}\right)^{1/2}$$

where $\phi(\omega)$ is the spectral density. Notice that this formula has similarities with the formula for determining the average residence time, Equation (2.24).

For a spectral density that is uniform over the interval (ω_1, ω_2), we get

$$N = \frac{1}{\pi}\left(\frac{\omega_2^3 - \omega_1^3}{3(\omega_2 - \omega_1)}\right)^{1/2} = \frac{1}{\pi}\left(\frac{\omega_1^2 + \omega_1\omega_2 + \omega_2^2}{3}\right)^{1/2}$$

For an ideal low-pass filter we have $\omega_1 = 0$ and $\omega_2 = \omega_b$, where ω_b is the band width. In this case we get

$$N = \frac{\omega_b}{\pi\sqrt{3}} = \frac{2\pi f_b}{\pi\sqrt{3}} \approx 1.16 f_b$$

The average rate of zero crossings per second is thus approximately equal to the bandwidth measured in Hz. Measurement of zero crossings can easily be combined with computation of the peak-to-peak value. More accurate determination of the spectral characteristics can be done by using a spectral analyzer or by recording a data set and computing the spectrum numerically.

Representation of Disturbances

It is often very convenient to consider signals as generated from a dynamic system with simple inputs as shown in Figure 2.30. For example, the signals shown in Figure 2.27 can be represented by sending an impulse through a dynamic system. The transfer function of the systems for the different signals are

Impulse $G(s) = 1$

Step $G(s) = \dfrac{1}{s}$

Ramp $G(s) = \dfrac{1}{s^2}$

Sinusoid $G(s) = \dfrac{\omega^2}{s^2 + \omega^2}$

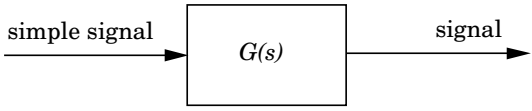

Figure 2.30 Signals represented as outputs of dynamic systems.

Similarly the signals in Figure 2.28 can be generated by sending a sequence of pulses through the same systems.

The same idea can be extended to describe noise. In this case the generating signal is white noise. We illustrate the idea.

EXAMPLE 2.13

The so called RC noise has the spectral density

$$\phi(\omega) = \frac{a^2}{\omega^2 + a^2}$$

It can be represented by sending white noise through a system with the transfer function

$$G(s) = \frac{a}{s + a}$$ □

The possibility of representing signals in this way also gives a possibility of dealing with nonstationary signals. The process obtained by sending white noise through an integrator, for example, is a drifting process that is called a random walk or a Wiener process.

The representation of signals in this way also makes it possible to see similarities between signals of different type. It turns out that signals that are generated from the same system have many similarities. For example, a step signal (Figure 2.27B), a piece-wise constant signal (Figure 2.28B), and a random walk are all generated by sending primitive signals through an integrator. The primitive signals are an impulse, a sequence of impulses for the piece-wise constant signals, or white noise for a random walk. A consequence of this is that a controller that is designed to work well for one of these signals will work well for all of them. A step disturbance is thus not as special as it may first appear.

2.9 Approximate Models and Unmodeled Dynamics

In this chapter, we have discussed various ways to model the process to be controlled. We end the chapter with a discussion about what is *not* captured by the models. Typical examples are nonlinearities and

process variations. Furthermore, the parametric low-order models give a satisfactory description of the behavior of the true system for signals with a limited frequency range only.

The process models describe the relation between the process input signal and the process output signal only, but the control system consists of other signals that influence the control performance. The characteristics of the setpoint changes, load disturbances, and measurement noise must also be taken into account in the design of the control system.

Many design methods for PID controllers require that the process model be of low order. Some methods to approximate higher-order process models with low-order models are also presented in this section.

Nonlinearities

All dynamic models presented in this chapter are linear, whereas most processes in practice are nonlinear. Nonlinear valves, actuators or sensors result in the process having different dynamics at different operating points. A linear model, obtained by transient or frequency response analysis of a nonlinear process, is only valid at the actual operating point. This means that a controller that is tuned based on this model may work well only at this operating point.

There are several ways to overcome the problem. A simple way is to tune the controller for the worst case and accept degraded performance at other operating conditions. If the characteristics of the nonlinearity are known, it can be compensated by feeding signals through a function module that forms the inverse of the nonlinearity. An example is a flow meter based on measurement of difference pressure. The flow is proportional to the square of the difference pressure. A linear relation between the flow and the output signal from the flow meter can be obtained by feeding the measured signal through a square root function.

Another way to compensate for nonlinearities is to divide the operating range into several smaller ranges where the process can be well approximated by linear models. A controller with satisfactory behavior for the full operating range can be obtained by determining one model for each operating range and changing the controller parameters with the operating condition. This approach is called *gain scheduling*.

Parametric Models

If the process is linear, a step response reveals all information about the process dynamics. In the same way, a Nyquist curve or frequency

response gives a complete description of the process dynamics. Information is lost when going from these graphical descriptions to parametric models. The parametric models derived for PID controller tuning are normally of low order. This means that quite a lot of information is lost. It is, therefore, particularly important that these simple models are derived properly and that their limitations are kept in mind when using them for controller tuning.

The parametric models based on step response analysis are often accurate at low frequencies, whereas they become more uncertain at higher frequencies. The simple models based on frequency response analysis, described in Section 2.6, are accurate at the frequencies of the input signals, but not at other frequencies. The basic relay method thus is accurate around the ultimate frequency ω_u, but not for higher and lower frequencies.

Process Variations

The model is valid at the time the experiment is performed. If the process dynamics change with time, it may not be valid at a later time. This problem can be handled in the same way as the nonlinearities described above.

A simple solution is to base the controller tuning on the model that describes the worst case. Gain scheduling can be used if the time variations can be related to some measurable variable. Adaptive control can be used if the process variations are random in the sense that they cannot be related to any measurable variable. Such a controller will adapt itself automatically to the actual process dynamics.

Disturbances

There are always disturbances acting on a control system. We distinguish between three types of disturbances, namely, setpoint changes (y_{sp}), load disturbances (l), and measurement noise (n) (see Figure 2.31).

Setpoint Changes

In process control, most control loops have a constant setpoint. (An exception is the controller sitting in the inner loop in cascade control.) The setpoint may change at certain time instances because of desires to change operating conditions such as production rates. The setpoint is, as a result, typically piece-wise constant with changes occurring rarely. It is, therefore, suitable to model the setpoint as a step function.

Since the setpoint is a disturbance that we have access to, it is possible to feed it through a low-pass filter or a ramping module before it enters the PID controller. In this way, the step function can be made smoother. This property is useful, since many control design methods giving a good rejection of load disturbances give too large overshoots after a sudden change in the setpoint.

Load Disturbances

Load disturbances are disturbances that enters the control loop somewhere in the process and drive the system away from its desired operating point. They may be caused by quality variations in a feed flow or variations in the demand flow, for example. These disturbances are the most common and the most important disturbances in process control. When discussing controller design in Chapter 4, we will, therefore, focus on the behavior with respect to these disturbances.

The load disturbance is typically a low-frequency disturbance, and it will, furthermore, be more or less low-pass filtered by the process depending on where in the process it enters. Consequently, it usually results in a low-frequency disturbance in the process output. To obtain this characteristic in the process output, we model the load disturbance as a step function added to the control signal at the process input (see Figure 2.31).

Measurement Noise

Measurement noise represents disturbances that distort the information about the process variables obtained from the sensors. Measurement noise may be of different character. It may be high-frequency fluctuations and it may be low-frequency calibration errors. With several sensors it is possible to reduce calibration errors. With only one sensor nothing can be done about calibration errors; we, therefore,

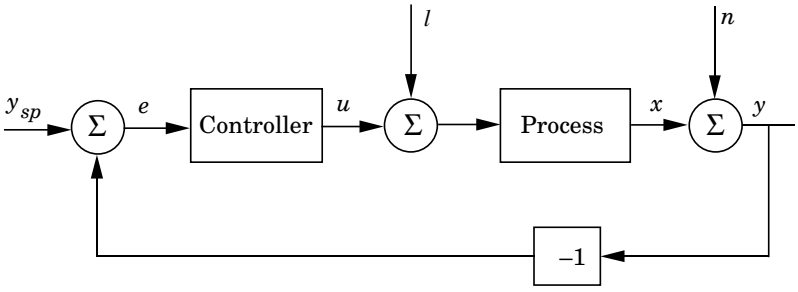

Figure 2.31 Block diagram of a simple feedback loop with three types of disturbances: Setpoint changes (y_{sp}), load disturbances (l) and measurement noise (n).

will model measurement noise as a high-frequency signal added to the process output.

Since measurement noise does not contain any information about the status of the process, it should be filtered out. Furthermore, high-frequency components in the measurement signal might be amplified by the controller and cause wear on the actuator. Filtering does introduce additional dynamics. It is therefore important to take the filter dynamics into account in the controlling design. We will model the measurement noise as an impulse function.

Approximating Complex Models

In modeling it is often convenient to split a system into interconnected subsystems. An example could be to divide a system into actuator, process, and sensor. Another example occurs when general control loops are cascaded. We may be interested then in obtaining a simplified description of the closed loop. Even if the model for each part is simple, the complete model may then be quite complex. Since many of the design methods for PID controllers are based on simple models, we need a procedure to simplify a complex model. Some ways to make such approximations are discussed below.

To perform the approximations it is necessary to determine the frequency range where the approximation should be valid. We do this simply by saying that the models should describe the system well around the frequency ω_0. This frequency should be approximately the same as the frequency of the dominant closed-loop poles of the desired system or the desired bandwidth of the closed-loop system. (The notion of dominant poles is discussed in Chapter 4.) Having restricted the modeling to a rather narrow frequency range, low-order models can now be determined by fitting them to experimental data, as described previously in this section.

Another possibility is to start with a complex model of the form

$$G_p(s) = K \frac{1 + b_1 s + b_2 s^2 + \cdots + b_n s^n}{1 + a_1 s + a_2 s^2 + \cdots + a_n s^n} e^{-sL}$$

and approximate it. The approximation is done in the following way. Poles and zeros that are much slower than ω_0 are approximated by integrators, poles, and zeros of the same order as ω_0 are retained, and poles and zeros that are much faster than ω_0 are neglected or approximated by a small time lag. A dead time such that $\omega_0 L \ll 1$ is neglected or approximated by a time constant. The approximation of fast poles and zeros by a first order system is illustrated by an example.

EXAMPLE 2.14 Approximation of fast modes

Consider the transfer function

$$G(s) = \frac{K(1+sT_1)(1+sT_2)}{(1+sT_3)(1+sT_4)(1+sT_5)(1+sT_6)}\, e^{-sL}$$

where

$$T = T_3 + T_4 + T_5 + T_6 - T_1 - T_2 - L > 0$$

and it is assumed that $L \ll T$. The transfer function G can be approximated by

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

□

EXAMPLE 2.15 Approximation of fast and slow modes

Consider the same system as in Example 2.14. Assume that

$$T_3 > T_4 > T_5 > T_6$$

and that

$$T_5 > \max\,(T_1, T_2, L)$$

Furthermore, let it be desired to obtain a model that describes the process well in the frequency range

$$\frac{1}{T_4} < \omega_0 < \frac{1}{T_5}$$

The time constant T_3 is slower than T_4 and T_5, and it will therefore be approximated by an integrator, i.e.,

$$\frac{1}{1+sT_3} \approx \frac{1}{sT_3}$$

The time constants T_1, T_2, T_6, and the time delay L are all smaller than T_5. They will be approximated by a single time constant

$$T = T_6 - T_1 - T_2 - L$$

If T is positive the system is then approximated by

$$G(s) = \frac{K}{sT_3(1+st_4)(1+sT_5)(1+sT)}$$

If T is negative the transfer function is instead approximated by

$$G(s) = \frac{K(1+sT)}{sT_3(1+sT_4)(1+sT_5)}$$

□

Summary

To summarize: When deriving a simple model to be used for PID controller tuning, it is important to ensure that the model describes the process well for the typical input signals obtained during the process operations. The amplitude and frequency distribution of the signal is of importance. Model accuracy may be poor if the process is nonlinear or time varying. Control quality can be improved by gain scheduling or adaptive control. It is also important to know what kind of disturbances are acting on the system and which limitation they impose.

2.10 Conclusions

Modeling is an important aspect of controller tuning. The models we need should describe how the process reacts to control signals. They should also describe the properties of the disturbances that enter the system. Most work on tuning of PID controllers have focused on the process dynamics, which is also reflected in the presentation in this chapter.

A number of methods for determining the dynamics of a process have been presented in this chapter. Some are very simple: they are based on a direct measurement of the step response and simple graphical constructions. Others are based on the frequency response. It has been shown that very useful information can be generated from relay feedback experiments. Such experiments are particularly useful because the process is brought into self-oscillation at the ultimate frequency, which is of considerable interest for design of controllers.

The simple methods are useful in field work when a controller has to be tuned and few tools are available. The methods are also useful to provide understanding as well as being references when more complicated methods are assessed. We have also presented more complicated methods that require significant computations.

Models of different complexity have been presented. Many models were characterized by a few parameters. Such models are useful for many purposes and are discussed in Chapter 4. When using such models it should be kept in mind that they are approximations.

When deriving the models we also introduced two dimension-free quantities, the normalized dead time τ and the gain ratio κ. These parameters make it possible to make a crude assessment of the difficulty of controlling the process. Processes with small values are easy to control. The difficulty increases as the values approach 1. Tuning rules based on τ and κ are provided in Chapter 5.

2.11 References

Process modeling is a key element in understanding and solving a control problem. Good presentations of modeling are found in standard textbooks on control, such as (Buckley, 1964), (Smith, 1972), (Seborg *et al.*, 1989), and (Luyben, 1990). These books have much material on many different modeling techniques. Similar presentations are given in (Gille *et al.*, 1959), (Harriott, 1964), (Oppelt, 1964), (Takahashi *et al.*, 1972), (Deshpande and Ash, 1981), (Shinskey, 1988), (Stephanopoulos, 1984), and (Hägglund, 1991). There are also books that specialize in modeling for control system design, see (Wellstead, 1979), (Nicholson, 1980), (Nicholson, 1981), and (Close and Frederick, 1993).

In the early work much effort was devoted to characterize dynamics by the step response, which at that time was called the response curve. See (Ziegler *et al.*, 1943) and the books (Tucker and Wills, 1960) and (Lloyd and Anderson, 1971), which were written by practitioners in control companies. A nice overview of step and frequency response methods is given in the paper (Rake, 1980). Additional details are given in (Strejc, 1959) and (Anderssen and White, 1971). Frequency response methods are presented in (Anderssen and White, 1970).

The relay method is treated in (Åström and Hägglund, 1984), (Hägglund and Åström, 1991), and (Schei, 1992). The describing function method is well documented in (Atherton, 1975) and (Gelb and Velde, 1968). There are many books on parameter estimation, the book (Johansson, 1993) is quite accessible. More mathematical details are found in (Ljung, 1987), (Ljung and Söderström, 1983), and (Söderström and Stoica, 1988). Many useful practical aspects on system identification are given in (Isermann, 1980).

PID Control

3.1 Introduction

The PID controller is by far the most common control algorithm. Most feedback loops are controlled by this algorithm or minor variations of it. It is implemented in many different forms, as a stand-alone controller or as a part of a DDC (Direct Digital Control) package or a hierarchical distributed process control system. Many thousands of instrument and control engineers worldwide are using such controllers in their daily work. The PID algorithm can be approached from many different directions. It can be viewed as a device that can be operated with a few rules of thumb, but it can also be approached analytically.

This chapter gives an introduction to PID control. The basic algorithm and various representations are presented in detail. A description of the properties of the controller in a closed loop based on intuitive arguments is given. The phenomenon of reset windup, which occurs when a controller with integral action is connected to a process with a saturating actuator, is discussed, including several methods to avoid it.

Some important aspects of digital computer implementation of PID controllers are given: issues such as prefiltering, different digital approximations, noise filtering, and computer code for good implementation. Operational aspects, such as bumpless transfer between manual and automatic mode and between different parameter sets, are also presented. The chapter ends with some aspects on the use and misuse of PID control with examples of systems where PID control works well and where it does not.

3.2 The Feedback Principle

The idea of feedback is deceptively simple and, yet, extremely powerful. It has had a profound influence on technology. Application of the feedback principle has resulted in major breakthroughs in control, communication, and instrumentation. Many patents have been granted on the idea. Assume for simplicity that the process is such that the process variable increases when the manipulated variable is increased. The principle of feedback can then be expressed as follows:

> Increase the manipulated variable when the process variable is smaller than the setpoint and decrease the manipulated variable when the process variable is larger than the setpoint.

This type of feedback is called *negative feedback* because the manipulated variable moves in opposite direction to the process variable. The feedback principle can be illustrated by the block diagram shown in Figure 3.1. In this diagram the process and the controller are represented as boxes with arrows denoting inputs and outputs. Notice also that there is a special symbol to denote the summation of signals. The block diagram shows that the process and the controller are connected in a closed feedback loop. The presence of the sign-reversing block indicates that the feedback is negative.

The reason why feedback systems are of interest is that feedback makes the process variable close to the setpoint in spite of disturbances and variation of the process characteristics.

On-Off Control

The feedback can be arranged in many different ways. A simple feedback mechanism can be described mathematically as follows:

$$u = \begin{cases} u_{\max} \text{ if } e > 0 \\ u_{\min} \text{ if } e < 0 \end{cases} \tag{3.1}$$

where $e = y_{sp} - y$ is the control error. This control law implies that maximum corrective action is always used. The manipulated variable,

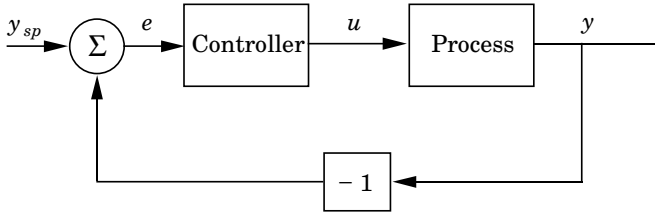

Figure 3.1 Block diagram of a process with a feedback controller.

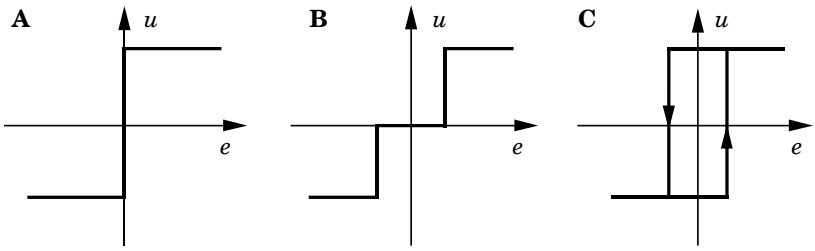

Figure 3.2 Controller characteristics for ideal on-off control (A), and modifications with dead zone (B) and hysteresis (C).

thus, has its largest value when the error is positive, and its smallest value when the error is negative. This type of feedback is called *on-off control*. It is simple and there are no parameters to choose. On-off control often succeeds in keeping the process variable close to the setpoint, but it will typically result in a system where the variables oscillate. Notice that in Equation (3.1) the control variable is not defined when the error is zero. It is common to have some modifications either by introducing hysteresis or a dead zone (see Figure 3.2).

Proportional Control

The reason why on-off control often gives rise to oscillations is that the system overreacts because a small change in the error will make the manipulated variable change over the full range. This effect is avoided in proportional control where the characteristic of the controller is proportional to the control error for small errors. Figure 3.3 shows the characteristic of a proportional controller. The controller is thus characterized by the nonlinear function $u = f_c(e)$ shown in the figure.

To describe the characteristic of a proportional controller we must of course give the limits $u_{\max}$ and $u_{\min}$ of the control variable. The linear range can be specified either by giving the slope of the characteristic (controller gain K) or by giving the range where the characteristic is linear (proportional band P_b). This range is normally centered around the setpoint. The proportional band and the controller gain are related through

$$u_{\max} - u_{\min} = K\, P_b \tag{3.2}$$

It is normally assumed that $u_{\max} - u_{\min} = 100$ %, which implies that

$$K = \frac{100}{P_b} \tag{3.3}$$

Notice that a proportional controller acts like an on-off controller for large errors.

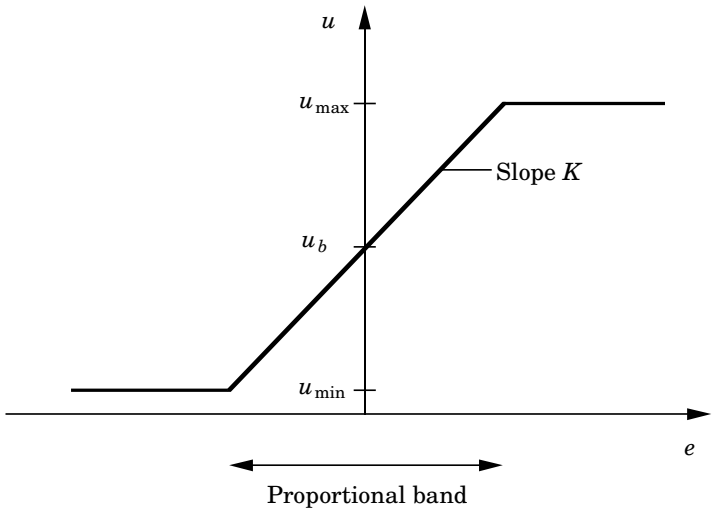

Figure 3.3 Characteristic of a proportional controller. The input is control error e and the output is control signal u.

Static Analysis of Feedback Systems

Some properties of a control system can be understood by a simple static analysis. To do this we introduce the static process characteristic, which is a curve that shows the stationary value of process output y as a function of process input u. See Figure 3.4. Notice that the curve has a physical interpretation only for a stable process. The static process characteristic is very important. It can be used to determine the range of control signals required to change the process output over the desired range, to size actuators, and to select sensor resolution. It can also be used to assess whether static gain variations are so large that they must be accounted for in the control design.

Proportional Control

Consider a process under proportional control. Let the controller characteristic be

$$u = f_c(y_{sp} - y) \tag{3.4}$$

Introducing the inverse controller characteristic f_c^{-1}, this can be written as

$$y_{sp} - y = f_c^{-1}(u)$$

Further introducing the static process characteristic,

$$y = f_p(u) \tag{3.5}$$

we find that the equilibrium value of u satisfies the equation

$$y_{sp} - f_c^{-1}(u) = f_p(u) \tag{3.6}$$

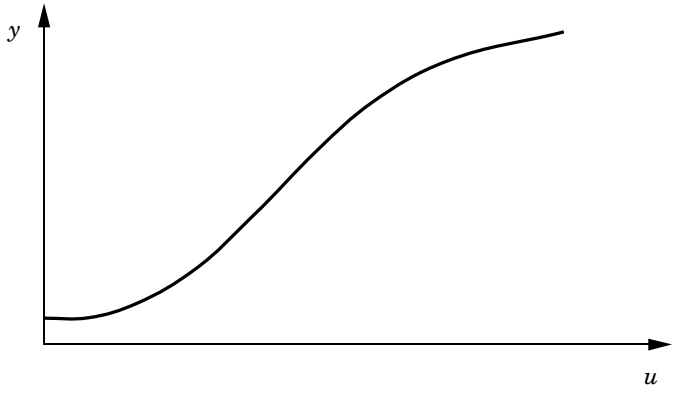

Figure 3.4 Static process characteristic. Shows process output y as a function of process input u under static conditions.

This equation can be solved graphically by finding the intersection between the graphs of the functions $f_p(u)$ and $y_{sp} - f_c^{-1}(u)$ as shown in Figure 3.5. The intersection is unique if the static characteristics are monotone. The equilibrium value of process output y is obtained simply as the y-coordinate of the intersection. In the graphical construction, it is easy to see how the equilibrium is influenced by the setpoint and the controller gain. The equilibrium agrees with the setpoint only if

$$y_{sp} = y_0 \stackrel{\text{def}}{=} f_p(u_b) \tag{3.7}$$

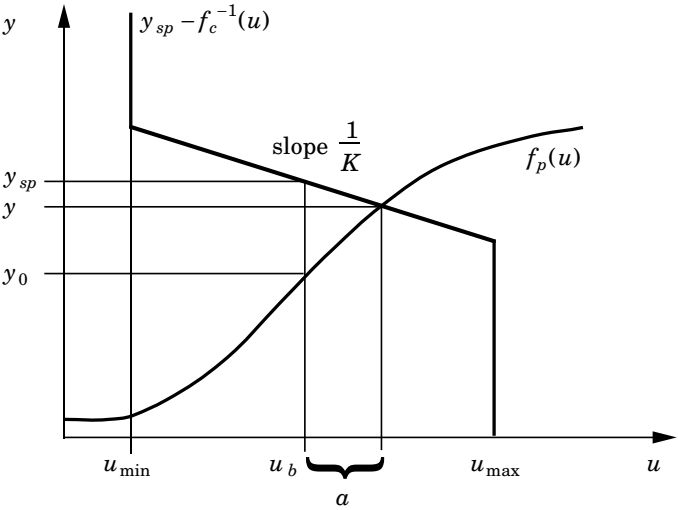

Figure 3.5 Determination of equilibrium from static process and controller characteristics.

For all other values of the setpoint there will be a deviation. If the process characteristic is approximated by a straight line with slope K_p, and the controller gain is K, the deviation can easily be computed. Introducing the parameter a shown in Figure 3.5, we find that

$$y_{sp} - y_0 = \left(K_p + \frac{1}{K} \right) a$$

and

$$y_{sp} - y = \frac{1}{K} a$$

This implies that the steady-state error is given by

$$e = y_{sp} - y = \frac{1}{1 + K_p K} (y_{sp} - y_0) \tag{3.8}$$

The smaller the deviation, the larger is the loop gain $K_p K$.

3.3 PID Control

In the previous section we saw that proportional control had the drawback that it mostly results in a static or steady state error. The control algorithms used in practice are, therefore, usually more complex than the proportional controller. It has been found empirically that a so-called PID controller is a useful structure. Inside the proportional band the behaviour of the "textbook" version of the PID algorithm can be described as:

$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \tag{3.9}$$

where u is the control variable and e is the control error ($e = y_{sp} - y$). The control variable is thus a sum of three terms: the P-term (which is proportional to the error), the I-term (which is proportional to the integral of the error), and the D-term (which is proportional to the derivative of the error). The controller parameters are proportional gain K, integral time T_i, and derivative time T_d.

Proportional Action

In the case of pure proportional control, the control law of Equation (3.9) reduces to

$$u(t) = K e(t) + u_b \tag{3.10}$$

The control action is simply proportional to the control error. The variable u_b is a bias or a reset. When the control error e is zero, the

control variable takes the value $u(t) = u_b$. Bias u_b is often fixed to $(u_{\max} + u_{\min})/2$, but can sometimes be adjusted manually so that the stationary control error is zero at a given setpoint.

Static Analysis

Several properties of proportional control can be understood by the following argument, which is based on pure static considerations. Consider the simple feedback loop, shown in Figure 3.6, and composed of a process and a controller. Assume that the controller has proportional action and that the process is modeled by the static model

$$x = K_p(u + l) \tag{3.11}$$

where x is the process variable, u is the control variable, l is a load disturbance, and K_p is the static process gain. The following equations are obtained from the block diagram.

$$\begin{aligned} y &= x + n \\ x &= K_p(u + l) \\ u &= K(y_{sp} - y) + u_b \end{aligned} \tag{3.12}$$

Elimination of intermediate variables gives the following relation between process variable x, setpoint y_{sp}, load disturbance l, and measurement noise n:

$$x = \frac{KK_p}{1 + KK_p}(y_{sp} - n) + \frac{K_p}{1 + KK_p}(l + u_b) \tag{3.13}$$

Compare with Equation (3.8) of the previous section. Product KK_p is a dimensionless number called the *loop gain*. Several interesting properties of the closed-loop system can be read from Equation (3.13). First assume that n and u_b are zero. Then the loop gain should be high in order to ensure that process output x is close to setpoint y_{sp}. A high value of the loop gain will also make the system insensitive to

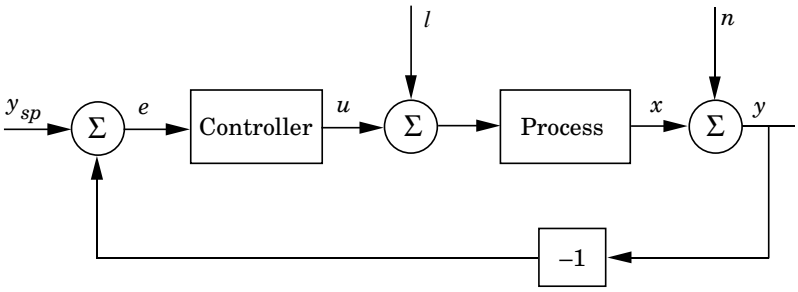

Figure 3.6 Block diagram of a simple feedback loop.

load disturbance l. However, if n is nonzero, it follows from Equation (3.13) that measurement noise n influences the process output in the same way as setpoint y_{sp}. To avoid making the system sensitive to measurement noise, the loop gain should not be made too large. Further, the controller bias u_b influences the system in the same way as a load disturbance. It is, therefore, obvious that the design of the loop gain is a trade-off between different control objectives, and that there is no simple answer to what loop gain is the best. This will depend on which control objective is the most important.

It also follows from Equation (3.13) that there will normally be a steady-state error with proportional control. This can be deduced intuitively from the observation following from Equation (3.12) that the control error is zero only when $u = u_b$ in stationarity. The error, therefore, can be made zero at a given operating condition by a proper choice of the controller bias u_b.

The static analysis given above is based on the assumption that the process can be described by a static model. This leaves out some important properties of the closed-loop system dynamics. The most important one is that the closed-loop system will normally be unstable for high-loop gains if the process dynamics are considered. In practice, the maximum loop gain is thus determined by the process dynamics. One way to describe process dynamics leads to descriptions like Equation (3.11) where the process gain is frequency-dependent. (This was discussed in Chapter 2.)

A typical example of proportional control is illustrated in Figure 3.7. The figure shows the behaviour of the process output and the

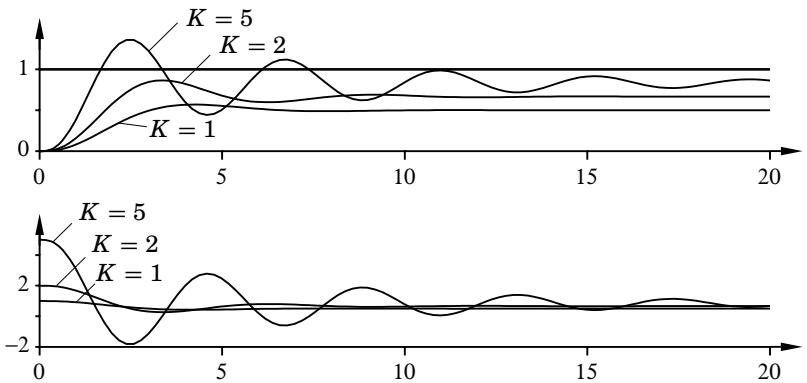

Figure 3.7 Simulation of a closed-loop system with proportional control. The process transfer function is $G(s) = (s+1)^{-3}$. The upper diagram shows setpoint $y_{sp} = 1$ and process output y for different values of controller gain K. The lower diagram shows control signal u for different controller gains.

control signal after a step change in the setpoint. The steady state error can be computed from Equation (3.13). The bias term u_b, the load l, and the noise n are all zero in the simulation. With a controller gain $K = 1$ and a static process gain $K_p = 1$, the error is therefore 50%. The figure shows that the steady state error decreases with increasing controller gain as predicted by Equation (3.13). Notice also that the response becomes more oscillatory with increasing controller gain. This is due to the process dynamics.

Integral Action

The main function of the integral action is to make sure that the process output agrees with the setpoint in steady state. With proportional control, there is normally a control error in steady state. With integral action, a small positive error will always lead to an increasing control signal, and a negative error will give a decreasing control signal no matter how small the error is.

The following simple argument shows that the steady-state error will always be zero with integral action. Assume that the system is in steady state with a constant control signal (u_0) and a constant error (e_0). It follows from Equation (3.9) that the control signal is then given by

$$u_0 = K\left(e_0 + \frac{e_0}{T_i}t\right)$$

As long as $e_0 \neq 0$, this clearly contradicts the assumption that the control signal u_0 is constant. A controller with integral action will always give zero steady-state error.

Integral action can also be visualized as a device that automatically resets the bias term u_b of a proportional controller. This is illustrated in the block diagram in Figure 3.8, which shows a proportional controller with a reset that is adjusted automatically. The adjustment is made by feeding back a signal, which is a filtered value of the output, to the summing point of the controller. This was actually one of the early inventions of integral action, or "automatic reset," as it was also called.

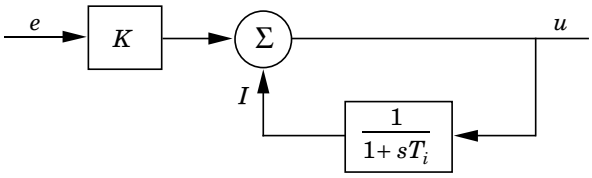

Figure 3.8 Implementation of integral action as positive feedback around a lag.

The implementation shown in Figure 3.8 is still used by many manufacturers. A simple calculation shows that the controller gives the desired results. The following equations follow from the block diagram:

$$u = Ke + I$$

$$T_i \frac{dI}{dt} + I = u$$

Elimination of u between these equations gives

$$T_i \frac{dI}{dt} + I = Ke + I$$

Hence,

$$T_i \frac{dI}{dt} = Ke$$

which shows that the controller in Figure 3.8 is, in fact, a PI controller.

The properties of integral action are illustrated in Figure 3.9, which shows a simulation of a system with PI control. The proportional gain is constant, $K = 1$ in all curves, and the integral time is changed. The case $T_i = \infty$ corresponds to pure proportional control. This case is identical to the case $K = 1$ in Figure 3.7, where the steady state error is 50%. The steady state error is removed when T_i has finite values. For large values of the integration time, the response creeps slowly towards the setpoint. The approach is approximately exponential with time constant T_i/KK_p. The approach is faster for smaller values of T_i; and it is also more oscillatory.

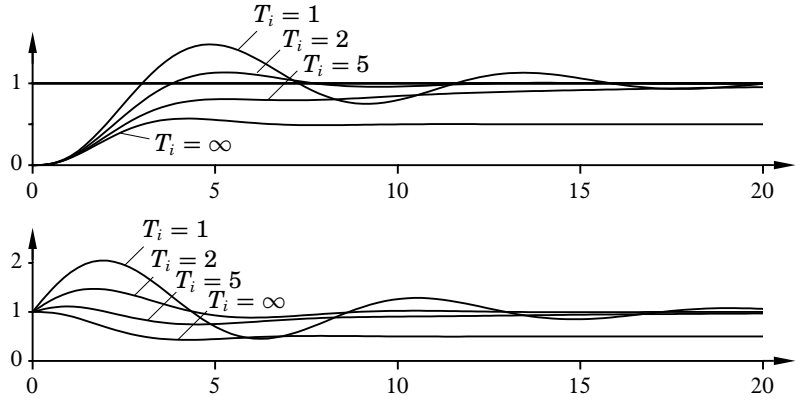

Figure 3.9 Simulation of a closed-loop system with proportional and integral control. The process transfer function is $G(s) = (s + 1)^{-3}$, and the controller gain is $K = 1$. The upper diagram shows setpoint $y_{sp} = 1$ and process output y for different values of integral time T_i. The lower diagram shows control signal u for different integral times.

Derivative Action

The purpose of the derivative action is to improve the closed-loop stability. The instability mechanism can be described intuitively as follows. Because of the process dynamics, it will take some time before a change in the control variable is noticeable in the process output. Thus, the control system will be late in correcting for an error. The action of a controller with proportional and derivative action may be interpreted as if the control is made proportional to the *predicted* process output, where the prediction is made by extrapolating the error by the tangent to the error curve (see Figure 3.10). The basic structure of a PD controller is

$$u(t) = K\left(e(t) + T_d \frac{de(t)}{dt}\right)$$

A Taylor series expansion of $e(t + T_d)$ gives

$$e(t + T_d) \approx e(t) + T_d \frac{de(t)}{dt}$$

The control signal is thus proportional to an estimate of the control error at time T_d ahead, where the estimate is obtained by linear extrapolation.

The properties of derivative action are illustrated in Figure 3.11, which shows a simulation of a system with PID control. Controller gain and integration time are kept constant, $K = 3$ and $T_i = 2$, and derivative time T_d is changed. For $T_d = 0$ we have pure PI control. The closed-loop system is oscillatory with the chosen parameters. Initially damping increases with increasing derivative time, but decreases again when derivative time becomes too large.

Summary

The PID controller has three terms. The proportional term P corresponds to proportional control. The integral term I gives a control

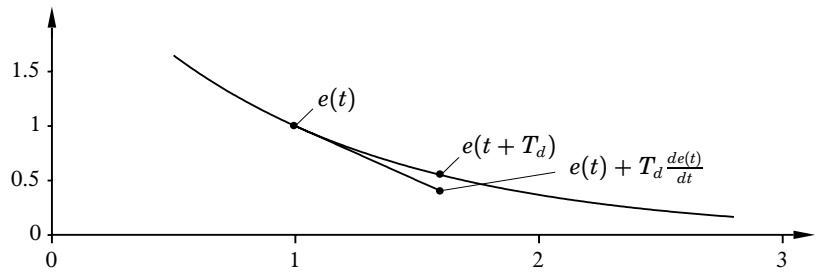

Figure 3.10 Interpretation of derivative action as predictive control, where the prediction is obtained by linear extrapolation.

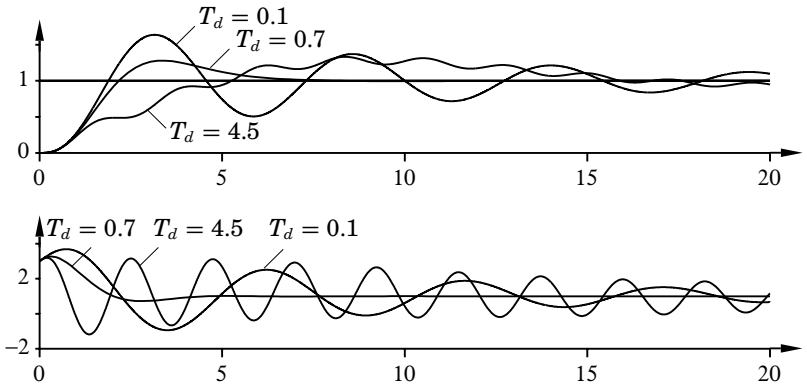

Figure 3.11 Simulation of a closed-loop system with proportional, integral and derivative control. The process transfer function is $G(s) = (s + 1)^{-3}$, the controller gain is $K = 3$, and the integral time is $T_i = 2$. The upper diagram shows setpoint $y_{sp} = 1$ and process output y for different values of derivative time T_d. The lower diagram shows control signal u for different derivative times.

action that is proportional to the time integral of the error. This ensures that the steady state error becomes zero. The derivative term D is proportional to the time derivative of the control error. This term allows prediction of the future error. There are many variations of the basic PID algorithm that will substantially improve its performance and operability. They are discussed in the next section.

3.4 Modifications of the PID Algorithm

The PID algorithm was given by Equation (3.9) in the previous section. This "textbook" algorithm is seldom used in practice because much better performance is obtained by the modified algorithm discussed in this section.

Alternative Representations

The PID algorithm given by (Equation 3.9) can be represented by the transfer function

$$G(s) = K\left(1 + \frac{1}{sT_i} + sT_d\right) \tag{3.14}$$

A slightly different version is most common in commercial controllers.

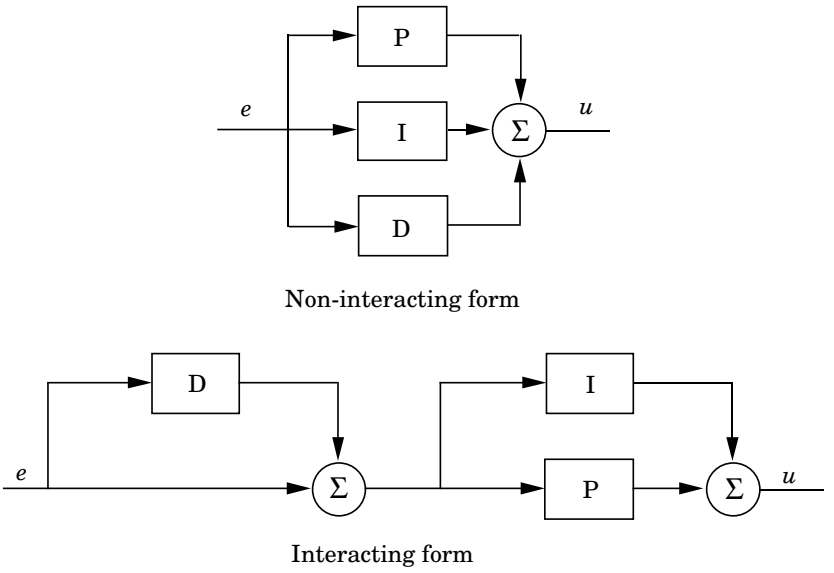

Figure 3.12 Interacting and non-interacting form of the PID algorithm.

This controller is described by

$$G'(s) = K' \left(1 + \frac{1}{sT_i'}\right)(1 + sT_d') \tag{3.15}$$

The two controller structures are presented in block diagram form in Figure 3.12. The controller given by Equation (3.14) is called non-interacting, and the one given by Equation (3.15) interacting. The reason for this nomenclature is that in the controller (3.14) the integral time T_i does not influence the derivative part, and the derivative time T_d does not influence the integral part (see Equation (3.14)). The parts are thus non-interacting. In the interacting controller, the derivative time T_d' does influence the integral part. Therefore, the parts are interacting.

The interacting controller (3.15) can always be represented as a non-interacting controller (3.14), whose coefficients are given by

$$\begin{aligned} K &= K' \frac{T_i' + T_d'}{T_i'} \\ T_i &= T_i' + T_d' \\ T_d &= \frac{T_i' T_d'}{T_i' + T_d'} \end{aligned} \tag{3.16}$$

An interacting controller of the form (3.15) that corresponds to a non-

interacting controller (3.14) can be found only if

$$T_i \geq 4T_d$$

Then,

$$\begin{aligned} K' &= \frac{K}{2}\left(1 + \sqrt{1 - 4T_d/T_i}\right) \\ T_i' &= \frac{T_i}{2}\left(1 + \sqrt{1 - 4T_d/T_i}\right) \\ T_d' &= \frac{T_i}{2}\left(1 - \sqrt{1 - 4T_d/T_i}\right) \end{aligned} \tag{3.17}$$

The non-interacting controller given by Equation (3.14) is more general, and we will use that in the future. It is, however, claimed that the interacting controller is easier to tune manually.

There is also an historical reason for preferring the interacting controller. Early pneumatic controllers were easier to build using the interacting form. When the controller manufacturers changed technology from pneumatic to analog electric and, finally, to digital technique, they kept the interactive form. Therefore, the interacting form is most common among single-loop controllers.

It is important to keep in mind that different controllers may have different structures. It means that if a controller in a certain control loop is replaced by another type of controller, the controller parameters may have to be changed. Note, however, that the interacting and the non-interacting forms are different only when both the I and the D parts of the controller are used. If we only use the controller as a P, PI, or PD controller, the two forms are equivalent. Yet another representation of the PID algorithm is given by

$$G''(s) = k + \frac{k_i}{s} + sk_d \tag{3.18}$$

The parameters are related to the parameters of standard form through

$$\begin{aligned} k &= K \\ k_i &= \frac{K}{T_i} \\ k_d &= KT_d \end{aligned}$$

The representation (3.18) is equivalent to the standard form, but the parameter values are quite different. This may cause great difficulties for anyone who is not aware of the differences, particularly if parameter $1/k_i$ is called integral time and k_d derivative time. The form given by Equation (3.18) is often useful in analytical calculations because the parameters appear linearly. The representation also has the advantage that it is possible to obtain pure proportional, integral, or derivative action by finite values of the parameters.

Summarizing we have thus found that there are three different forms of the PID controller.

- The standard or non-interacting form given by Equation (3.14).
- The series or interacting form given by Equation (3.15).
- The parallel form given by Equation (3.18).

The standard form is sometimes called the ISA algorithm, or the ideal algorithm. The proportional, integral, and derivative actions are noninteracting in the time domain. This algorithm admits complex zeros, which is useful when controlling systems with oscillatory poles.

The series form is also called the classical form. This representation is obtained naturally when a controller is implemented as an analog device based on a pneumatic force balance system. The name classical reflects this. The series form has an attractive interpretation in the frequency domain because the zeros correspond to the inverse values of the derivative and integral times. All zeros of the controller are real. Pure integral or proportional action can not be obtained with finite values of the controller parameters. Most controllers use this form.

The parallel form is the most general form, because pure proportional or integral action can be obtained with finite parameters. The controller can also have complex zeros. In this way it is the most flexible form. However, it is also the form where the parameters have little physical interpretation.

Setpoint Weighting

A common form of a control system is shown in Figure 3.6. The system is characterized by forming an error that is the difference between the setpoint and the process output. The controller generates a control signal by operating on the error. This control signal is then applied to the process. Such a system is called a "system with error feedback" because the controller operates on the error signal. A more flexible structure is obtained by treating the setpoint and the process output separately. A PID-controller of this form is given by

$$u(t) = K\left(e_p + \frac{1}{T_i}\int_0^t e(s)ds + T_d\frac{de_d}{dt}\right) \tag{3.19}$$

where the error in the proportional part is

$$e_p = by_{sp} - y \tag{3.20}$$

and the error in the derivative part is

$$e_d = cy_{sp} - y \tag{3.21}$$

The error in the integral part must be the true control error

$$e = y_{sp} - y$$

to avoid steady-state control errors. The controllers obtained for different values of b and c will respond to load disturbances and measurement noise in the same way. The response to setpoint changes will depend, however, on the values of b and c. This is illustrated in Figure 3.13, which shows the response of a PID controller to setpoint changes, load disturbances, and measurement errors for different values of b. The figure shows clearly the effect of changing b. The overshoot for setpoint changes is smallest for $b = 0$, which is the case where the reference is only introduced in the integral term, and increases with increasing b. Notice that a simulation like the one in Figure 3.13 is useful in order to give a quick assessment of the responses of a closed-loop system to setpoint changes, load disturbances, and measurement errors.

The parameter c is normally chosen equal to zero to avoid large transients in the control signal due to sudden changes in the setpoint. An exception is when the controller is the secondary controller in a cascade coupling (see Section 7.2). In this case, the setpoint changes smoothly, because it is given by the primary controller output. Notice that if the integral action is implemented with positive feedback around a lag as in Figure 3.8, the parameter b is equal to one.

The controller with $b = 0$ and $c = 0$ is sometimes called an I-PD controller, and the controller with $b = 1$ and $c = 0$ is sometimes called a PI-D controller. We prefer to stick to the generic use of PID and give the parameters b and c, thereby making a small contribution towards reduction of three-letter abbreviations.

In general, a control system has many different requirements. It should have good transient response to setpoint changes, and it should reject load disturbances and measurement noise. For a system with error feedback only, an attempt is made to satisfy all demands with the same mechanism. Such systems are called one-degree of freedom systems. By having different signal paths for the setpoint and the process output (two-degree of freedom systems), there is more flexibility to satisfy the design compromise. This is carried much further in more sophisticated control systems.

In the block diagram in Figure 3.6, the controller output is generated from the error $e = y_{sp} - y$. Notice that this diagram is no longer valid when the control law given by Equation (3.19) and the error definitions (3.20) and (3.21) are used. A block diagram for a system with PID control is now given by Figure 3.14.

Notice that the transfer function from the setpoint y_{sp} to the

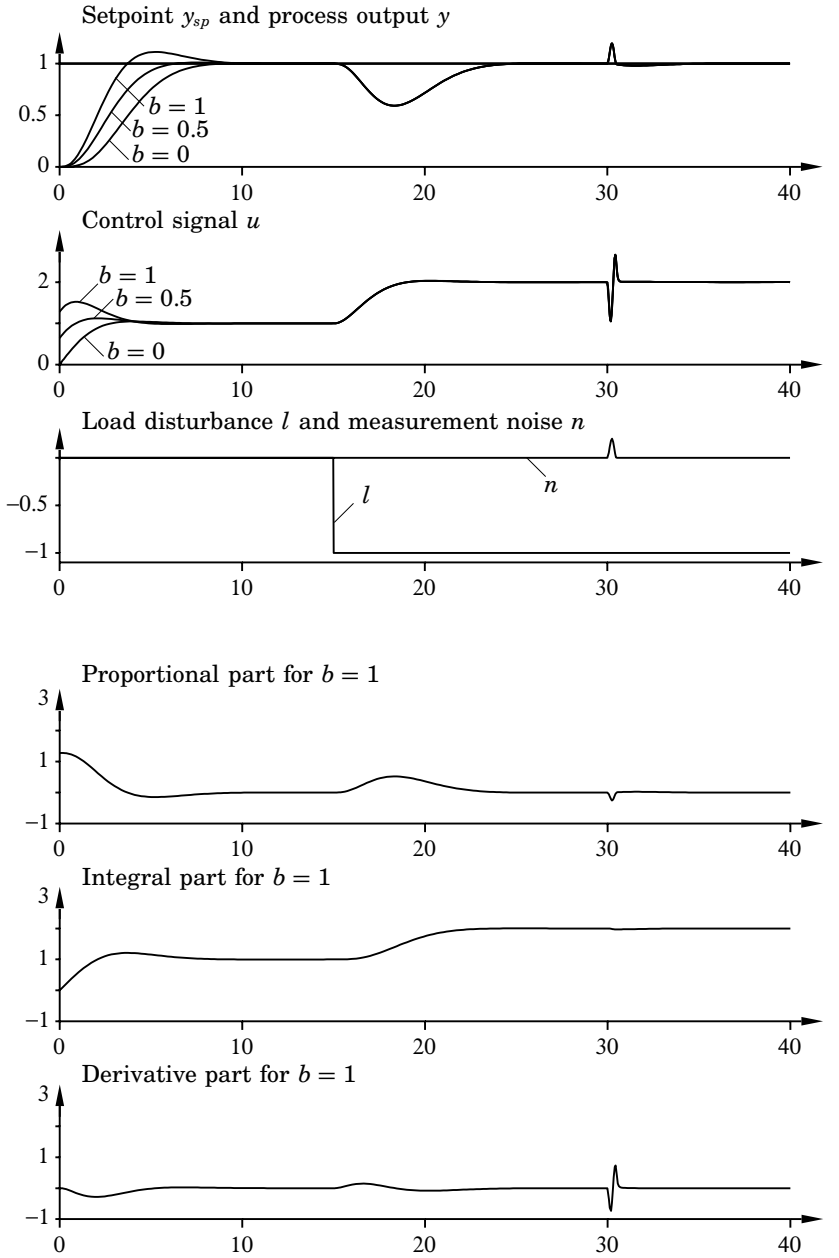

Figure 3.13 The response to setpoint changes, load disturbances, and measurement errors for different values of setpoint weighting b. The lower diagrams show the proportional, integral, and derivative parts of the control signal.

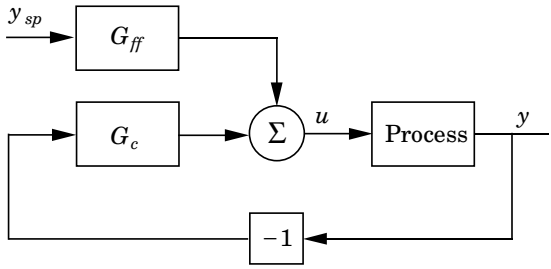

Figure 3.14 Block diagram of a simple feedback loop whith a PID controller having a two-degree-of-freedom structure.

control signal u is given by

$$G_{ff} = K\left(b + \frac{1}{sT_i} + csT_d\right)$$

and the transfer function from the process variable y to the control variable u is given by

$$G_c = K\left(1 + \frac{1}{sT_i} + sT_d\right)$$

and that the transfer functions are different.

Limitation of the Derivative Gain

The derivative action may result in difficulties, if there is high-frequency measurement noise. A sinusoidal measurement noise

$$n = a \sin \omega t$$

gives the following contribution to the derivative term of the control signal:

$$u_n = KT_d \frac{dn}{dt} = aKT_d\omega \cos \omega t$$

The amplitude of the control signal can thus be arbitrarily large if the noise has a sufficiently high frequency (ω). The high-frequency gain of the derivative term is therefore limited to avoid this difficulty. This can be done by implementing the derivative term as

$$D = -\frac{T_d}{N}\frac{dD}{dt} - KT_d\frac{dy}{dt} \tag{3.22}$$

It follows from this equation that the modified derivative term can be represented as follows:

$$D = -\frac{sKT_d}{1 + sT_d/N}y$$

The modification can be interpreted as the ideal derivative filtered by a first-order system with the time constant T_d/N. The approximation acts as a derivative for low-frequency signal components. The gain, however, is limited to KN. This means that high-frequency measurement noise is amplified at most by a factor KN. Typical values of N are 8 to 20.

Error-Squared Controllers

In the standard form of PID control, the control error enters linearly in the control algorithm, see Equation (3.9). It is sometimes desirable to have higher controller gains when the control error is large, and smaller gains when the control error is small. One common way of obtaining this property is to use the square of the control error, i.e., the control error is substituted by

$$e_{\text{squared}} = e|e|$$

The square of the error is mostly used only in the proportional term, sometimes in the integral term, but seldom in the derivative term.

One reason for using error-squared controllers is to reduce the effects of low-frequency disturbances in the measurement signal. These disturbances cannot be filtered out, but the use of error-squared control gives a small amplification of the noise when the control error is small, and an effective control when the control error is large.

Another application of error-squared controllers is surge tank control. Here, the main control objective is to keep the control signal smooth. On the other hand, the level must not deviate too much from the setpoint. This is obtained efficiently by error-squared control.

Special Controller Outputs

The inputs and outputs of a controller are normally analog signals, typically 0–20 mA or 4–20 mA. The main reason for using 4 mA instead of 0 mA as the lower limit is that many transmitters are designed for two-wire connection. This means that the same wire is used for both driving the sensor and transmitting the information from the sensor. It would not be possible to drive the sensor with a current of 0 mA. The main reason for using current instead of voltage is to avoid the influence of voltage drops along the wire due to resistance in the (perhaps long) wire.

Thyristors and Triacs

In temperature controllers it is common practice to integrate the power amplifier with the controller. The power amplifier could be a

thyristor or a triac. With a thyristor, an AC voltage is switched to the load at a given angle of the AC voltage. Since the relation between angle and power is nonlinear, it is crucial to use a transformation to maintain a linear relationship. A triac is also a device that implements switching of an AC signal, but only at the zero crossing. Such a device is similar to a pulse output.

Pulse Width Modulation

In some cases, such as with the triac, there is an extreme quantization in the sense that the actuator only accepts two values, on or off. In such a case, a cycle time T_{cycle} is specified, and the controller gives a pulse with width

$$T_{\text{pulse}}(t) = \frac{u(t) - u_{\min}}{u_{\max} - u_{\min}} T_{\text{cycle}} \tag{3.23}$$

A similar, but slightly different, situation occurs when the actuator has three levels: max, min, and zero. A typical example is a motor-

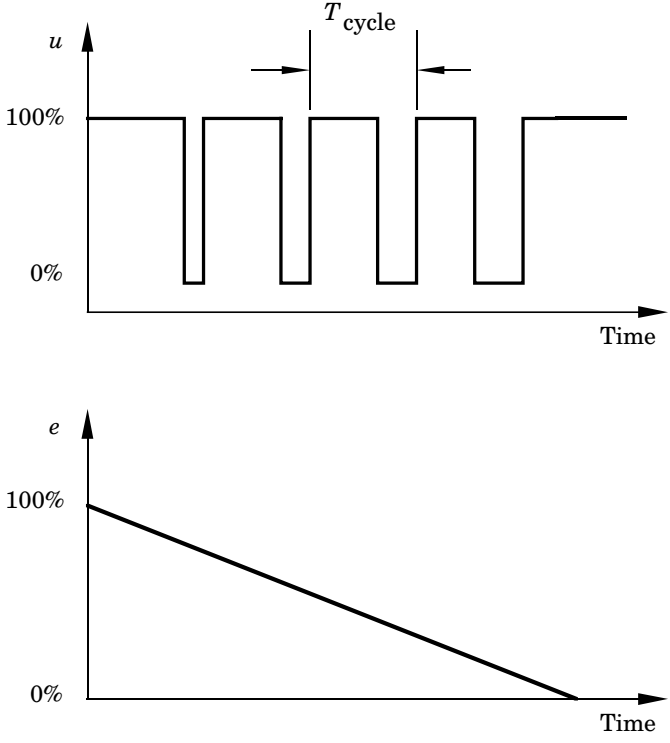

Figure 3.15 Illustration of controller output based on pulse width modulation.

driven valve where the motor can stand still, go forward, or go backward.

Figure 3.15 illustrates the pulse width modulation. The figure shows the output from a P controller with pulse width modulation for different values of the control error.

Velocity Algorithms

The algorithms described so far are called positional algorithms because the output of the algorithms is the control variable. In certain cases the control system is arranged in such a way that the control signal is driven directly by an integrator, e.g., a motor. It is then natural to arrange the algorithm in such a way that it gives the velocity of the control variable. The control variable is then obtained by integrating its velocity. An algorithm of this type is called a velocity algorithm. A block diagram of a velocity algorithm for a PID controller is shown in Figure 3.16. Velocity algorithms were commonly used in many early controllers that were built around motors. In several cases, the structure was retained by the manufacturers when technology was changed in order to maintain functional compatibility with older equipment. Another reason is that many practical issues, like wind-up protection and bumpless parameter changes, are easy to implement using the velocity algorithm. This is discussed further in Sections 3.5 and 3.6. In digital implementations velocity algorithms are also called incremental algorithms.

A Difficulty with Velocity Algorithms

A velocity algorithm cannot be used directly for a controller without integral action, because such a controller cannot keep the stationary

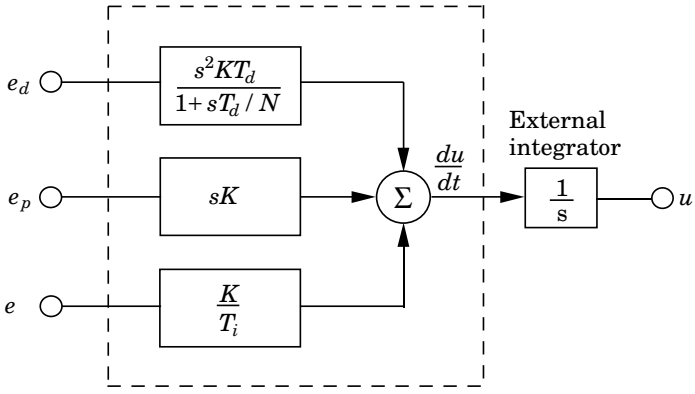

Figure 3.16 Block diagram of a PID algorithm in velocity form.

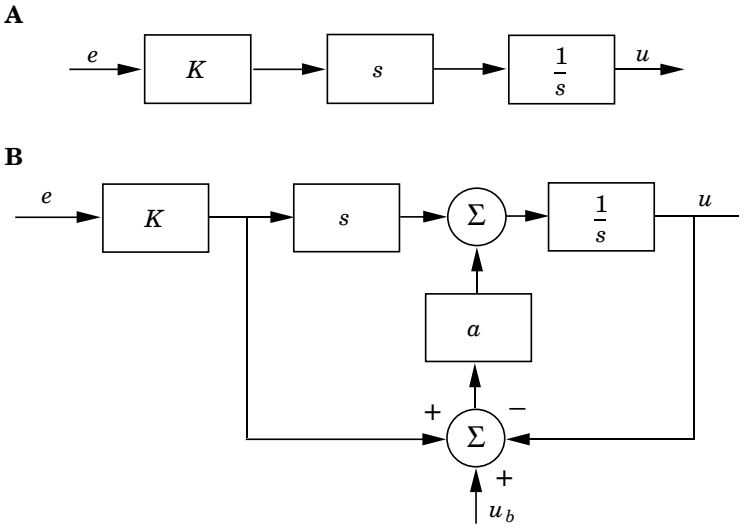

Figure 3.17 Illustrates the difficulty with a proportional controller in velocity form (A) and a way to avoid it (B).

value. This can be understood from the block diagram in Figure 3.17A, which shows a proportional controller in velocity form. Stationarity can be obtained for any value of the control error e, since the output from the derivation block is zero for any constant input. The problem can be avoided with the modification shown in Figure 3.17B. Here, stationarity is only obtained when $u = Ke + u_b$.

If a sampled PID controller is used, a simple version of the method illustrated in figure 3.17B is obtained by implementing the P controller as

$$\Delta u(t) = u(t) - u(t-h) = Ke(t) + u_b - u(t-h)$$

where h is the sampling period.

3.5 Integrator Windup

Although many aspects of a control system can be understood based on linear theory, some nonlinear effects must be accounted for. All actuators have limitations: a motor has limited speed, a valve cannot be more than fully opened or fully closed, etc. For a control system with a wide range of operating conditions, it may happen that the control variable reaches the actuator limits. When this happens the feedback loop is broken and the system runs as an open loop because the actuator will remain at its limit independently of the process

output. If a controller with integrating action is used, the error will continue to be integrated. This means that the integral term may become very large or, colloquially, it "winds up". It is then required that the error has opposite sign for a long period before things return to normal. The consequence is that any controller with integral action may give large transients when the actuator saturates.

EXAMPLE 3.1 Illustration of integrator windup

The wind-up phenomenon is illustrated in Figure 3.18, which shows control of an integrating process with a PI controller. The initial setpoint change is so large that the actuator saturates at the high limit. The integral term increases initially because the error is positive; it reaches its largest value at time $t = 10$ when the error goes through zero. The output remains saturated at this point because of the large value of the integral term. It does not leave the saturation limit until the error has been negative for a sufficiently long time to let the integral part come down to a small level. Notice that the control signal bounces between its limits several times. The net effect is a large overshoot and a damped oscillation where the control signal flips from one extreme to the other as in relay oscillation. The output finally comes so close to the setpoint that the actuator does not saturate. The system then behaves linearly and settles. □

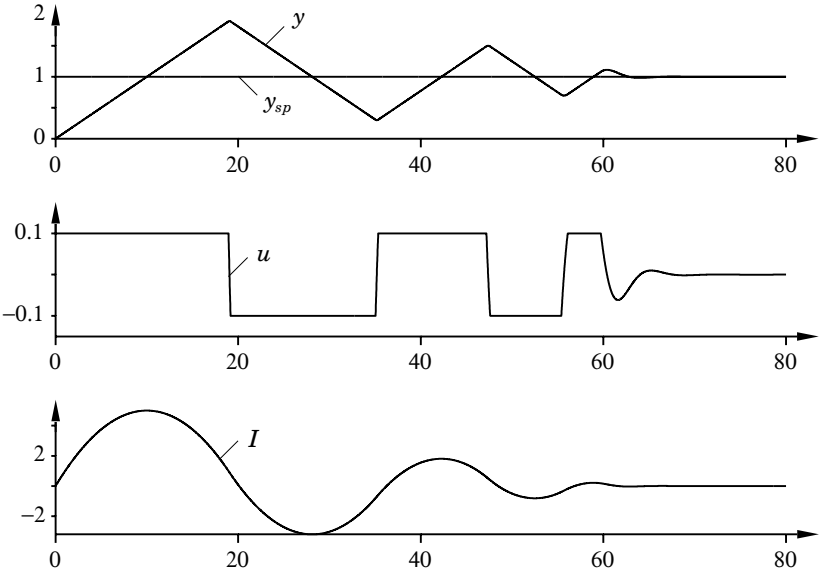

Figure 3.18 Illustration of integrator windup. The diagrams show process output y, setpoint y_{sp}, control signal u, and integral part I.

Integrator windup may occur in connection with large setpoint changes or it may be caused by large disturbances or equipment malfunctions. Windup can also occur when selectors are used so that several controllers are driving one actuator. In cascade control, windup may occur in the primary controller when the secondary controller is switched to manual mode, uses its local setpoint, or if its control signal saturates. See Section 7.2.

The phenomenon of windup was well known to manufacturers of analog controllers who invented several tricks to avoid it. They were described under labels like preloading, batch unit, etc. Although the problem was well understood, there were often limits imposed because of the analog implementations. The ideas were often kept as trade secrets and not much spoken about. The problem of windup was rediscovered when controllers were implemented digitally and several methods to avoid windup were presented in the literature. In the following section we describe several of the ideas.

Setpoint Limitation

One way to try to avoid integrator windup is to introduce limiters on the setpoint variations so that the controller output will never reach the actuator bounds. This often leads to conservative bounds and limitations on controller performance. Further, it does not avoid windup caused by disturbances.

Incremental Algorithms

In the early phases of feedback control, integral action was integrated with the actuator by having a motor drive the valve directly. In this case windup is handled automatically because integration stops when the valve stops. When controllers were implemented by analog techniques, and later with computers, many manufacturers used a configuration that was an analog of the old mechanical design. This led to the so-called velocity algorithms discussed in Section 3.4. In this algorithm the rate of change of the control signal is first computed and then fed to an integrator. In some cases this integrator is a motor directly connected to the actuator. In other cases the integrator is implemented internally in the controller. With this approach it is easy to handle mode changes and windup. Windup is avoided by inhibiting the integration whenever the output saturates. This method is equivalent to back-calculation, which is described below. If the actuator output is not measured, a model that computes the saturated output can be used. It is also easy to limit the rate of change of the control signal.

Back-Calculation and Tracking

Back-calculation works as follows: When the output saturates, the integral is recomputed so that its new value gives an output at the saturation limit. It is advantageous not to reset the integrator instantaneously but dynamically with a time constant T_t.

Figure 3.19 shows a block diagram of a PID controller with anti-windup based on back-calculation. The system has an extra feedback path that is generated by measuring the actual actuator output and forming an error signal (e_s) as the difference between the output of the controller (v) and the actuator output (u). Signal e_s is fed to the input of the integrator through gain $1/T_t$. The signal is zero when

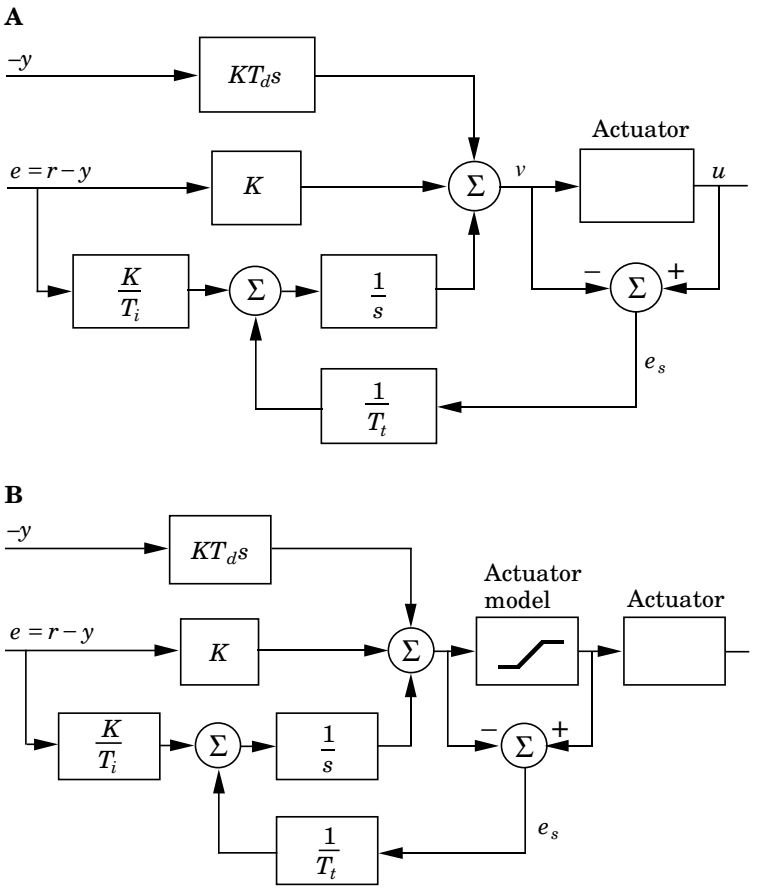

Figure 3.19 Controller with anti-windup. A system where the actuator output is measured is shown in A and a system where the actuator output is estimated from a mathematical model is shown in B.

there is no saturation. Thus, it will not have any effect on the normal operation when the actuator does not saturate. When the actuator saturates, the signal e_s is different from zero. The normal feedback path around the process is broken because the process input remains constant. There is, however, a feedback path around the integrator. Because of this, the integrator output is driven towards a value such that the integrator input becomes zero. The integrator input is

$$\frac{1}{T_t} e_s + \frac{K}{T_i} e$$

where e is the control error. Hence,

$$e_s = -\frac{K T_t}{T_i} e$$

in steady state. Since $e_s = u - v$, it follows that

$$v = u_{\lim} + \frac{K T_t}{T_i} e$$

where u_{lim} is the saturating value of the control variable. Since the signals e and $u_{\lim}$ have the same sign, it follows that v is always larger than $u_{\lim}$ in magnitude. This prevents the integrator from winding up.

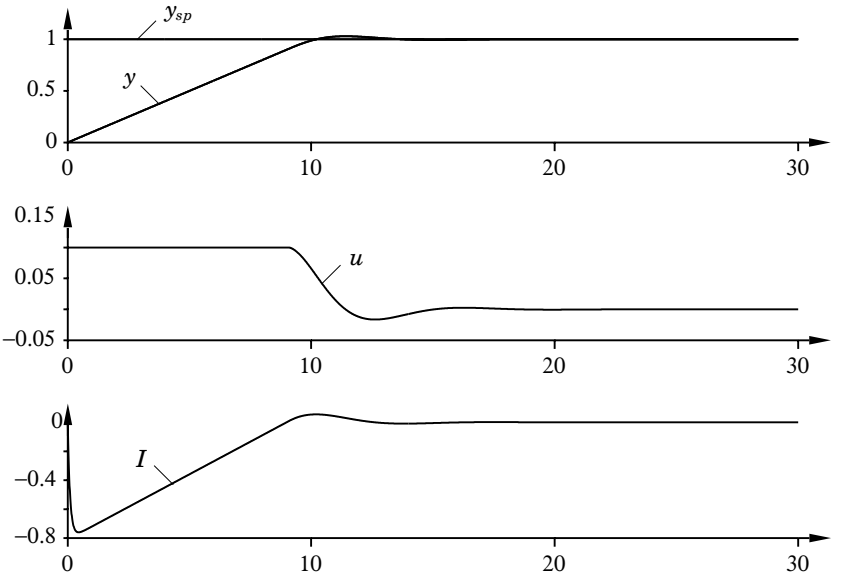

Figure 3.20 Controller with anti-windup applied to the system of Figure 3.18. The diagrams show process output y, setpoint y_{sp}, control signal u, and integral part I.

The rate at which the controller output is reset is governed by the feedback gain, $1/T_t$, where T_t can be interpreted as the time constant, which determines how quickly the integral is reset. We call this the tracking time constant.

It frequently happens that the actuator output cannot be measured. The anti-windup scheme just described can be applied by incorporating a mathematical model of the saturating actuator, as is illustrated in Figure 3.19B.

Figure 3.20 shows what happens when a controller with anti-windup is applied to the system simulated in Figure 3.18. Notice that the output of the integrator is quickly reset to a value such that the controller output is at the saturation limit, and the integral has a negative value during the initial phase when the actuator is saturated. This behavior is drastically different from that in Figure 3.18, where the integral has a positive value during the initial transient. Also notice the drastic improvement in performance compared to the ordinary PI controller used in Figure 3.18.

The effect of changing the values of the tracking time constant is illustrated in Figure 3.21. From this figure, it may thus seem advantageous to always choose a very small value of the time constant because the integrator is then reset quickly. However, some care must be exercised when introducing anti-windup in systems with derivative action. If the time constant is chosen too small, spurious errors can cause saturation of the output, which accidentally resets the integrator. The tracking time constant T_t should be larger than T_d and smaller than T_i. A rule of thumb that has been suggested is to choose $T_t = \sqrt{T_i T_d}$.

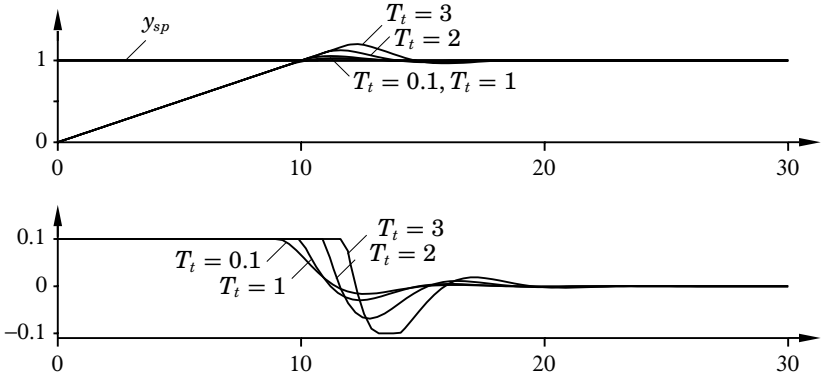

Figure 3.21 The step response of the system in Figure 3.18 for different values of the tracking time constant T_t. The upper curve shows process ouput y and setpoint y_{sp}, and the lower curve shows control signal u.

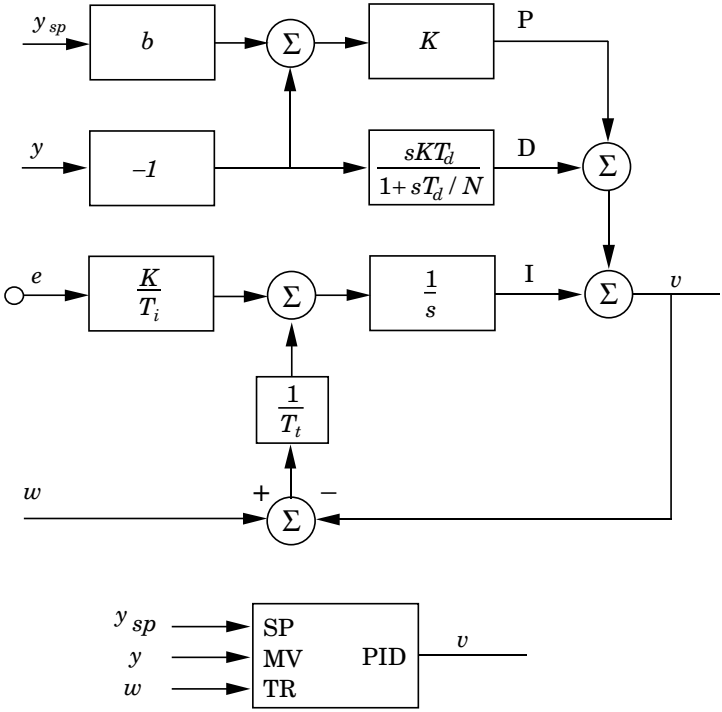

Figure 3.22 Block diagram and simplified representation of PID module with tracking signal.

Controllers with a Tracking Mode

A controller with back-calculation can be interpreted as having two modes: the normal *control mode*, when it operates like an ordinary controller, and a *tracking mode*, when the integrator is tracking so that it matches given inputs and outputs. Since a controller with tracking can operate in two modes, we may expect that it is necessary to have a logical signal for mode switching. However, this is not necessary, because tracking is automatically inhibited when the tracking signal w is equal to the controller output. This can be used with great advantage when building up complex systems with selectors and cascade control.

Figure 3.22 shows a PID module with a tracking signal. The module has three inputs: the setpoint, the measured output, and a tracking signal. The new input TR is called a tracking signal because the controller output will follow this signal. Notice that tracking is inhibited when $w = v$. Using the module the system shown in Figure 3.19 can be presented as shown in Figure 3.23.

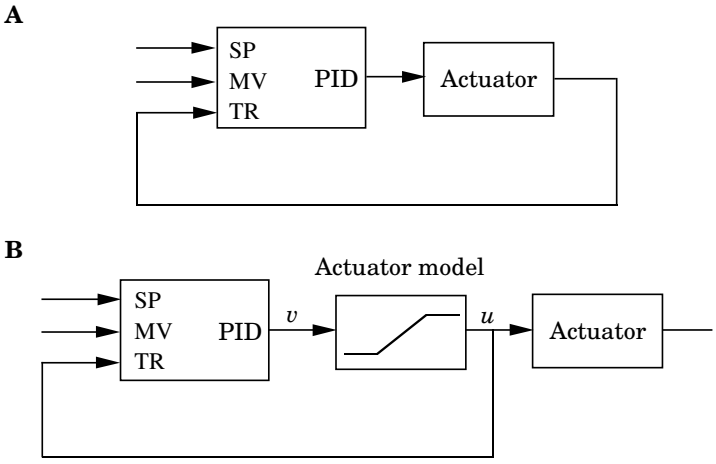

Figure 3.23 Representation of the controllers with anti-windup in Figure 3.19 using the basic control module with tracking shown in Figure 3.22.

The Proportional Band

The notion of proportional band is useful in order to understand the wind-up effect and to explain schemes for anti-windup. The *proportional band* is an interval such that the actuator does not saturate if the instantaneous value of the process output or its predicted value is in the interval. For PID control without derivative gain limitation, the control signal is given by

$$u = K(by_{sp} - y) + I - KT_d \frac{dy}{dt} \tag{3.24}$$

Solving for the predicted process output

$$y_p = y + T_d \frac{dy}{dt}$$

gives the proportional band (y_l, y_h) as

$$\begin{aligned} y_l &= by_{sp} + \frac{I - u_{\max}}{K} \\ y_h &= by_{sp} + \frac{I - u_{\min}}{K} \end{aligned} \tag{3.25}$$

and $u_{\min}$, $u_{\max}$ are the values of the control signal for which the actuator saturates. The controller operates in the linear mode, if the predicted output is in the proportional band. The control signal saturates when the predicted output is outside the proportional band. Notice that the proportional band can be shifted by changing the integral term.

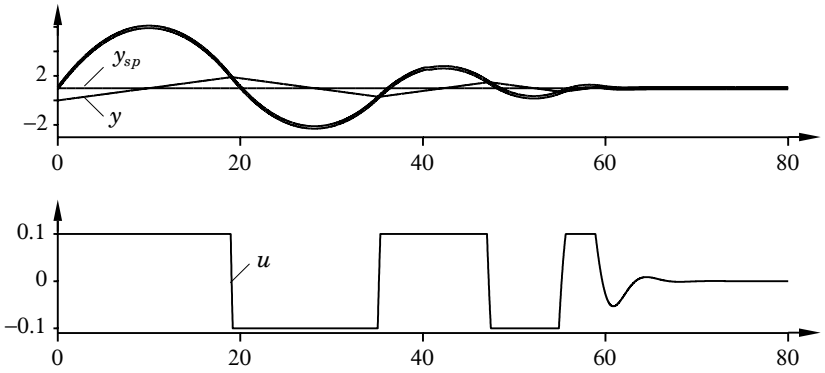

Figure 3.24 The proportional band for the system in Example 3.1. The upper diagram shows process output y and the proportional band. The lower diagram shows control signal u.

To illustrate that the proportional band is useful in understanding windup, we show the proportional band in Figure 3.24 for the system discussed in Example 3.1. (Compare with Figure 3.18.) The figure shows that the proportional band starts to move upwards because the integral term increases. This implies that the output does not reach the proportional band until it is much larger than the setpoint. When the proportional band is reached the control signal decreases rapidly. The proportional band changes so rapidly, however, that the output very quickly moves through the band, and this process repeats several times.

The notion of proportional band helps to understand several schemes for anti-windup. Figure 3.25 shows the proportional band for the system with tracking for different values of the tracking time constant T_t. The figure shows that the tracking time constant has a significant influence on the proportional band. Because of the tracking, the proportional band is moved closer to the process output. How rapidly it does this is governed by the tracking time constant T_t. Notice that there may be a disadvantage in moving it too rapidly, since the predicted output may then move into the proportional band because of noise, and cause the control signal to decrease unnecessarily.

Conditional Integration

Conditional integration is an alternative to back-calculation or tracking. In this method integration is switched off when the control is far from steady state. Integral action is thus only used when certain conditions are fulfilled, otherwise the integral term is kept constant. The method is also called integrator clamping.

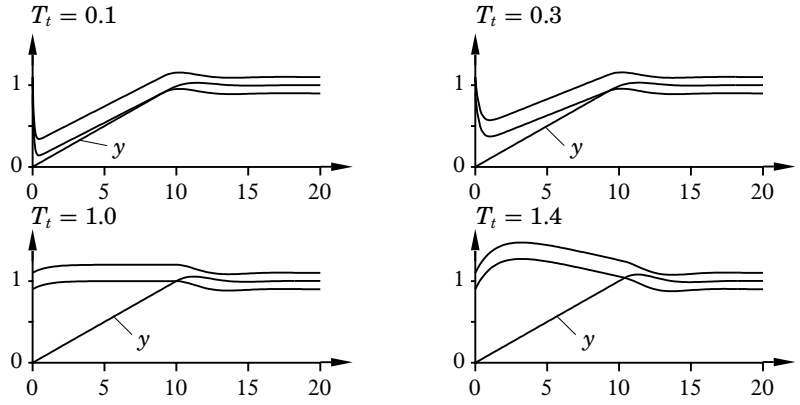

Figure 3.25 The proportional band and the process output y for a system with conditional integration and tracking with different tracking time constants T_t.

The conditions when integration is inhibited can be expressed in many different ways. Figure 3.26 shows a simulation of the system in Example 3.1 with conditional integration such that the integral term is kept constant during saturation. A comparison with Figure 3.25 shows that, in this particular case, there is very little difference in performance between conditional integration and tracking. The different wind-up schemes do, however, move the proportional bands differently.

A few different switching conditions are now considered. One simple approach is to switch off integration when the control error is large. Another approach is to switch off integration during saturation. Both these methods have the disadvantage that the controller may get stuck at a non-zero control error if the integral term has a large value at the time of switch off.

A method without this disadvantage is the following. Integration is switched off when the controller is saturated *and* the integrator update is such that it causes the control signal to become more saturated. Suppose, for example, that the controller becomes saturated at the upper saturation. Integration is then switched off if the control error is positive, but not if it is negative.

Some conditional integration methods are intended mainly for startup of batch processes, when there may be large changes in the setpoint. One particular version, used in temperature control, sets the proportional band outside the setpoint when there are large control deviations. The offset can be used to adjust the transient response obtained during start up of the process. The parameters used are called cut-back or preload (see Figure 3.27). In this system the proportional band is positioned with one end at the setpoint and the other end

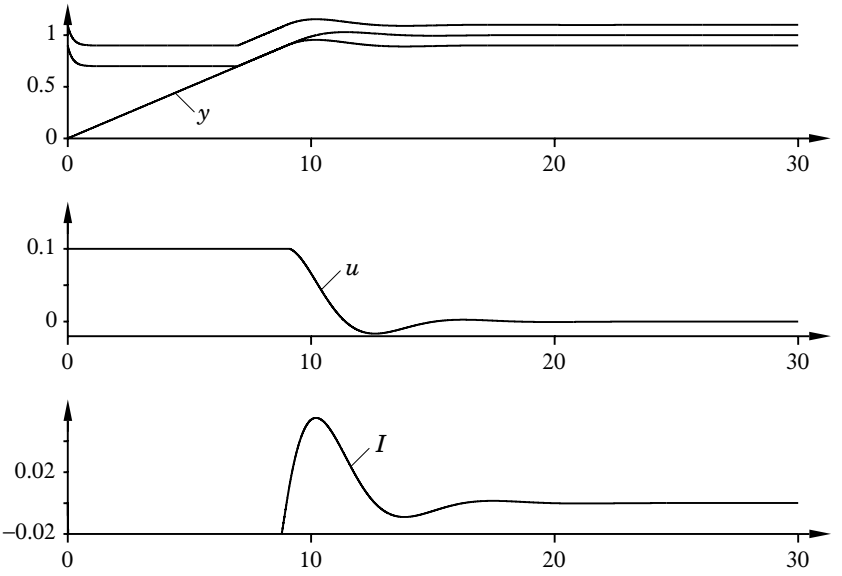

Figure 3.26 Simulation of the system in Example 3.1 with conditional integration. The diagrams show the proportional band, process output y, control signal u, and integral part I.

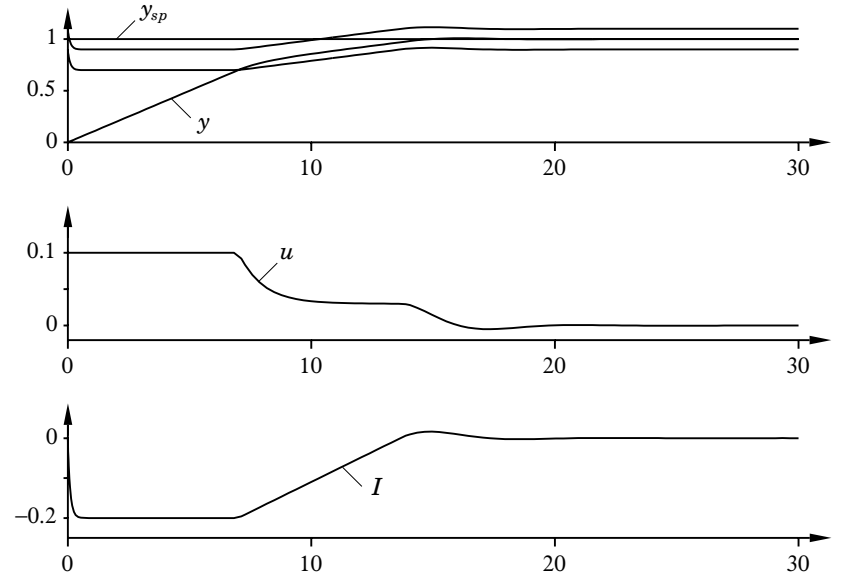

Figure 3.27 Adjustment of the proportional band using cut-back parameters. The diagrams show the proportional band, setpoint y_{sp}, process output y, control signal u, and integral part I.

towards the measured value when there are large variations. These methods may give wind-up during disturbances.

Series Implementation

In Figure 3.8, we showed a special implementation of a controller in interacting form. To avoid windup in this controller we can incorporate a model of the saturation in the system as shown in Figure 3.28A. Notice that in this implementation the tracking time constant T_t is the same as the integration time T_i. This value of the tracking time constant is often too large.

In Figure 3.28A, the model of the saturation will limit the control signal directly. It is important, therefore, to have a good model of the physical saturation. Too hard a limitation will cause an unnecessary limitation of the control action. Too weak a limitation will cause windup.

More flexibility is provided if the saturation is positioned according to Figure 3.28B. In this case, the saturation will not influence the proportional part of the controller. With this structure it is also possible to force the integral part to assume other preload values during saturation. This is achieved by replacing the saturation function by the nonlinearity shown in Figure 3.29. This anti-windup procedure is sometimes called a "batch unit" and may be regarded as a type of conditional integration. It is mainly used for adjusting the

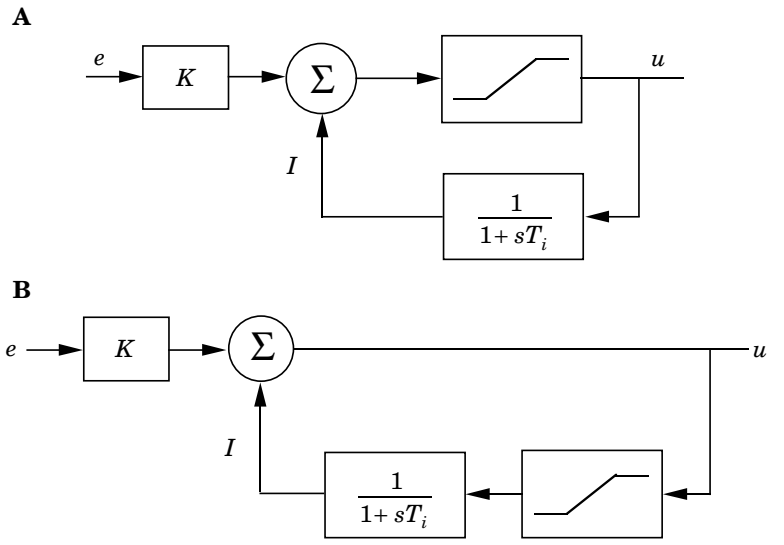

Figure 3.28 Two ways to provide anti-windup in the controller in Figure 3.8 where integral action is generated as automatic reset.

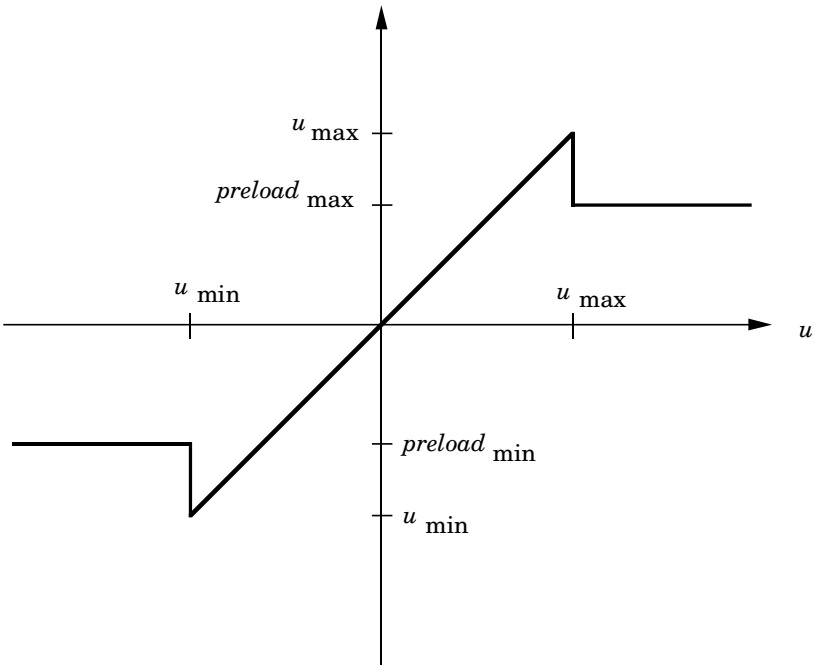

Figure 3.29 A "batch unit" used to provide anti-windup in the controller in Figure 3.8.

overshoot during startup when there is a large setpoint change. In early single-loop controllers the batch unit was supplied as a special add-on hardware.

Combined Schemes

Tracking and conditional integration can also be combined. In (Howes, 1986) it is suggested to manipulate the proportional band explicitly for batch control. This is done by introducing so-called *cutback points*. The high cutback is above the setpoint and the low cutback is below. The integrator is clamped when the predicted process output is outside the cutback interval. Integration is performed with a specified tracking time constant when the process output is between the cutback points. The cutback points are considered as controller parameters that are adjusted to influence the response to large setpoint changes. A similar method is proposed in (Dreinhofer, 1988), where conditional integration is combined with back-calculation. In (Shinskey, 1988), the integrator is given a prescribed value $i = i_0$ during saturation. The value of i_0 is tuned to give satisfactory overshoot at startup. This approach is also called preloading.

3.6 Digital Implementation

PID controllers were originally implemented using analog techniques. Early systems used pneumatic relays, bellows, and needle-valve constrictions. Electric motors with relays and feedback circuits and operational amplifiers were used later. Many of the features like antiwindup and derivation of process output instead of control error were incorporated as "tricks" in the implementation. It is now common practice to implement PID controllers using microprocessors, and some of the old tricks have been rediscovered. Several issues must be considered in connection with digital implementations. The most important ones have to do with sampling, discretization, and quantization.

Sampling

When a digital computer is used to implement a control law, all signal processing is done at discrete instances of time. The sequence of operations is as follows:

(1) Wait for clock interrupt
(2) Read analog input
(3) Compute control signal
(4) Set analog output
(5) Update controller variables
(6) Go to 1

The control actions are based on the values of the process output at discrete times only. This procedure is called *sampling*. The normal case is that the signals are sampled periodically with period h. The sampling mechanism introduces some unexpected phenomena, which must be taken into account in a good digital implementation of a PID controller. To explain these, consider the signals

$$s(t) = \cos(n\omega_s t \pm \omega t)$$

and

$$s_a(t) = \cos(\omega t)$$

where $\omega_s = 2\pi/h$ [rad/s] is the sampling frequency. Well-known formulas for the cosine function imply that the values of the signals at the sampling instants $[kh, k = 0, 1, 2, ...]$ have the property

$$s(kh) = \cos(nkh\omega_s \pm \omega kh) = \cos(\omega kh) = s_a(\omega kh)$$

The signals s and s_a thus have the same values at the sampling instants. This means that there is no way to separate the signals if only their values at the sampling instants are known. Signal s_a

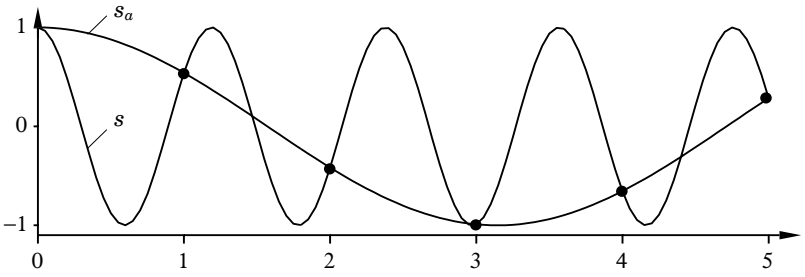

Figure 3.30 Illustration of the aliasing effect. The diagram shows signal s and its alias s_a.

is, therefore, called an *alias* of signal s. This is illustrated in Figure 3.30. A consequence of the aliasing effect is that a high-frequency disturbance after sampling may appear as a low-frequency signal. In Figure 3.30 the sampling period is 1 s and the sinusoidal disturbance has a period of 6/5 s. After sampling, the disturbance appear as a sinusoid with the frequency

$$f_a = 1 - \frac{5}{6} = 1/6 \text{ Hz}$$

This low-frequency signal with time period 6 s is seen in the figure.

Prefiltering

The aliasing effect can create significant difficulties if proper precautions are not taken. High frequencies, which in analog controllers normally are effectively eliminated by low-pass filtering, may, because of aliasing, appear as low-frequency signals in the bandwidth of the sampled control system. To avoid these difficulties, an analog prefilter (which effectively eliminates all signal components with frequencies above half the sampling frequency) should be introduced. Such a filter is called an antialiasing filter. A second-order Butterworth filter is a common antialiasing filter. Higher-order filters are also used in critical applications. An implementation of such a filter using operational amplifiers is shown in Figure 3.31. The selection of the filter bandwidth is illustrated by the following example.

EXAMPLE 3.2 Selection of prefilter bandwidth

Assume it is desired that the prefilter attenuate signals by a factor of 16 at half the sampling frequency. If the filter bandwidth is ω_b and the sampling frequency is ω_s, we get

$$(\omega_s/2\omega_b)^2 = 16$$

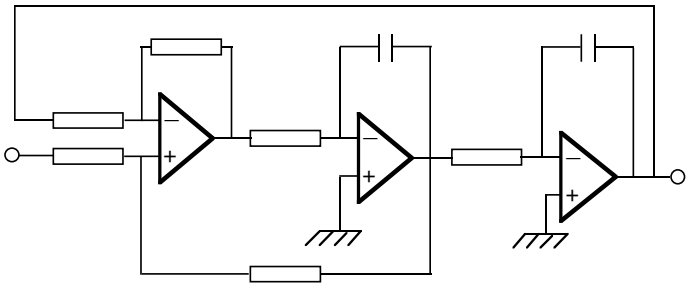

Figure 3.31 Circuit diagram of a second-order Butterworth filter.

Hence,

$$\omega_b = \frac{1}{8}\,\omega_s$$ □

Notice that the dynamics of the prefilter will be combined with the process dynamics.

Discretization

To implement a continuous-time control law, such as a PID controller in a digital computer, it is necessary to approximate the derivatives and the integral that appear in the control law. A few different ways to do this are presented below.

Proportional Action

The proportional term is

$$P = K(by_{sp} - y)$$

This term is implemented simply by replacing the continuous variables with their sampled versions. Hence,

$$P(t_k) = K\,(by_{sp}(t_k) - y(t_k)) \tag{3.26}$$

where $\{t_k\}$ denotes the sampling instants, i.e., the times when the computer reads the analog input.

Integral Action

The integral term is given by

$$I(t) = \frac{K}{T_i}\int_0^t e(s)ds$$

It follows that

$$\frac{dI}{dt} = \frac{K}{T_i}\, e \tag{3.27}$$

There are several ways of approximating this equation.

Forward differences: Approximating the derivative by a forward difference gives

$$\frac{I(t_{k+1}) - I(t_k)}{h} = \frac{K}{T_i}\, e(t_k)$$

This leads to the following recursive equation for the integral term

$$I(t_{k+1}) = I(t_k) + \frac{Kh}{T_i}\, e(t_k) \tag{3.28}$$

Backward differences: If the derivative in Equation (3.27) is approximated instead by a backward difference, the following is obtained:

$$\frac{I(t_k) - I(t_k - 1)}{h} = \frac{K}{T_i}\, e(t_k)$$

This leads to the following recursive equation for the integral term

$$I(t_{k+1}) = I(t_k) + \frac{Kh}{T_i}\, e(t_{k+1}) \tag{3.29}$$

Tustin's approximation and ramp equivalence: Another simple approximation method is due to Tustin. This approximation is

$$I(t_{k+1}) = I(t_k) + \frac{Kh}{T_i}\, \frac{e(t_{k+1}) + e(t_k)}{2} \tag{3.30}$$

Yet another method is called ramp equivalence. This method gives exact outputs at the sampling instants, if the input signal is continuous and piece-wise linear between the sampling instants. The ramp equivalence method gives the same approximation of the integral term as the Tustin approximation, i.e., Equation (3.30).

Notice that all approximations have the same form, i.e.,

$$I(t_{k+1}) = I(t_k) + b_{i1} e(t_{k+1}) + b_{i2} e(t_k) \tag{3.31}$$

but with different values of parameters b_{i1} and b_{i2}.

Derivative Action

The derivative term is given by Equation (3.22), i.e.,

$$\frac{T_d}{N}\, \frac{dD}{dt} + D = -K T_d\, \frac{dy}{dt} \tag{3.32}$$

This equation can be approximated in the same way as the integral term.

Forward differences: Approximating the derivative by a forward difference gives

$$\frac{T_d}{N}\frac{D(t_{k+1}) - D(t_k)}{h} + D(t_k) = -KT_d\frac{y(t_{k+1}) - y(t_k)}{h}$$

This can be rewritten as

$$D(t_{k+1}) = \left(1 - \frac{Nh}{T_d}\right) D(t_k) - KN\left(y(t_{k+1}) - y(t_k)\right) \tag{3.33}$$

Backward differences: If the derivative in Equation (3.32) is approximated by a backward difference, the following equation is obtained:

$$\frac{T_d}{N}\frac{D(t_k) - D(t_{k-1})}{h} + D(t_k) = -KT_d\frac{y(t_k) - y(t_{k-1})}{h}$$

This can be rewritten as

$$D(t_k) = \frac{T_d}{T_d + Nh} D(t_{k-1}) - \frac{KT_dN}{T_d + Nh}\left(y(t_k) - y(t_{k-1})\right) \tag{3.34}$$

Tustin's approximation: Using the Tustin approximation to approximate the derivative term gives

$$D(t_k) = \frac{2T_d - Nh}{2T_d + Nh} D(t_{k-1}) - \frac{2KT_dN}{2T_d + Nh}\left(y(t_k) - y(t_{k-1})\right) \tag{3.35}$$

Ramp equivalence: Finally, the ramp equivalence approximation is

$$D(t_k) = e^{-Nh/T_d} D(t_{k-1}) - \frac{KT_d(1 - e^{-Nh/T_d})}{h}\left(y(t_k) - y(t_{k-1})\right) \tag{3.36}$$

All approximations have the same form,

$$D(t_k) = a_d D(t_{k-1}) - b_d\left(y(t_k) - y(t_{k-1})\right) \tag{3.37}$$

but with different values of parameters a_d and b_d.

The approximations of the derivative term are stable only when $|a_d| < 1$. The forward difference approximation requires that $T_d > Nh/2$. The approximation becomes unstable for small values of T_d. The other approximations are stable for all values of T_d. Notice, however, that Tustin's approximation and the forward difference approximation give negative values of a_d if T_d is small. This is undesirable because the approximation will then exhibit ringing. The backward difference approximation will give good results for all values of T_d. It is also easier to compute than the ramp equivalence approximation and is, therefore, the most common method.

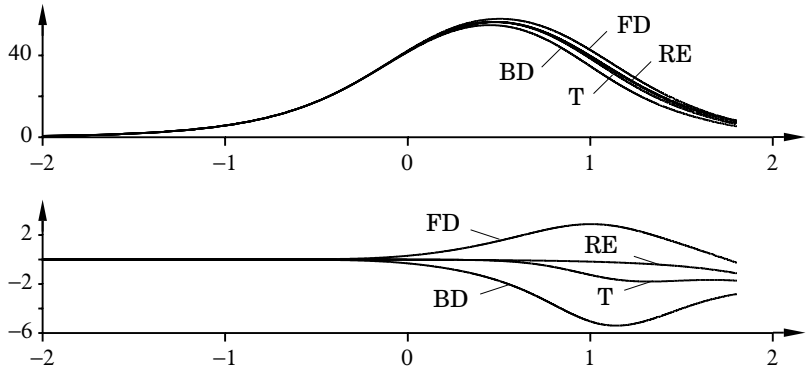

Figure 3.32 Phase curves for PD controllers obtained by different discretizations of the derivative term $sT_d/(1 + sT_d/N)$ with $T_d = 1, N = 10$ and a sampling period 0.02. The discretizations are forward differences (FD), backward differences (BD), Tustin's approximation (T), and ramp equivalence (RE). The lower diagram shows the differences between the approximations and the true phase curve.

Figure 3.32 shows the phase curves for the different discrete time approximations. Tustin's approximation and the ramp equivalence approximation give the best agreement with the continuous time case, the backward approximation gives less phase advance, and the forward approximation gives more phase advance. The forward approximation is seldom used because of the problems with instability for small values of derivative time T_d. Tustin's algorithm is used quite frequently because of its simplicity and its close agreement with the continuous time transfer function. The backward difference is used when an algorithm that is well behaved for small T_d is needed.

All approximations of the PID controller can be represented as

$$R(q)u(kh) = T(q)y_{sp}(kh) - S(q)y(kh) \tag{3.38}$$

where q is the forward shift operator, and the polynomials R, S, and T are of second order. The polynomials R, S, and T have the forms

$$\begin{aligned} R(q) &= (q-1)(q-a_d) \\ S(q) &= s_0q^2 + s_1q + s_2 \\ T(q) &= t_0q^2 + t_1q + t_2 \end{aligned} \tag{3.39}$$

which means that Equation (3.38) can be written as

$$\begin{aligned} u(kh) = {} & t_0y_{sp}(kh) + t_1y_{sp}(kh-h) + t_2y_{sp}(kh-2h) \\ & - s_0y(kh) - s_1y(kh-h) - s_2y(kj-2h) \\ & + (1+a_d)u(kh-h) - a_du(kh-h) \end{aligned}$$

Table 3.1 Coefficients in different approximations of the continuous time PID controller.

	Forward	Backward	Tustin	Ramp equivalence
b_{i1}	0	$\frac{Kh}{T_i}$	$\frac{Kh}{2T_i}$	$\frac{Kh}{2T_i}$
b_{i2}	$\frac{Kh}{T_i}$	0	$\frac{Kh}{2T_i}$	$\frac{Kh}{2T_i}$
a_d	$1-\frac{Nh}{T_d}$	$\frac{T_d}{T_d+Nh}$	$\frac{2T_d-Nh}{2T_d+Nh}$	e^{-Nh/T_d}
b_d	KN	$\frac{KT_dN}{T_d+Nh}$	$\frac{2KT_dN}{2T_d+Nh}$	$\frac{KT_d(1-e^{-Nh/T_d})}{h}$

The coefficients in the S and T polynomials are

$$\begin{aligned} s_0 &= K + b_{i1} + b_d \\ s_1 &= -K(1+a_d) - b_{i1}a_d + b_{i2} - 2b_d \\ s_2 &= Ka_d - b_{i2}a_d + b_d \\ t_0 &= Kb + b_{i1} \\ t_1 &= -Kb(1+a_d) - b_{i1}a_d + b_{i2} \\ t_2 &= Kba_d - b_{i2}a_d \end{aligned} \tag{3.40}$$

The coefficients in the polynomials for different approximation methods are given in Table 3.1.

Incremental Form

The algorithms described so far are called positional algorithms because they give the output of the controller directly. In digital implementations it is common to also use velocity algorithms. The discrete time version of such an algorithm is also called an incremental algorithm. This form is obtained by computing the time differences of the controller output and adding the increments.

$$\Delta u(t_k) = u(t_k) - u(t_{k-1}) = \Delta P(t_k) + \Delta I(t_k) + \Delta D(t_k)$$

In some cases integration is performed externally. This is natural when a stepper motor is used. The output of the controller should then represent the increments of the control signal, and the motor implements the integrator. The increments of the proportional part, the integral part, and the derivative part are easily calculated

from Equations (3.26), (3.31) and (3.37):

$$\Delta P(t_k) = P(t_k) - P(t_{k-1}) = K\,(by_{sp}(t_k) - y(t_k) - by_{sp}(t_{k-1}) + y(t_{k-1}))$$
$$\Delta I(t_k) = I(t_k) - I(t_{k-1}) = b_{i1}\,e(t_k) + b_{i2}\,e(t_{k-1})$$
$$\Delta D(t_k) = D(t_k) - D(t_{k-1}) = a_d \Delta D(t_{k-1}) - b_d\,(y(t_k) - 2y(t_{k-1}) + y(t_{k-2}))$$

One advantage with the incremental algorithm is that most of the computations are done using increments only. Short word-length calculations can often be used. It is only in the final stage where the increments are added that precision is needed. Another advantage is that the controller output is driven directly from an integrator. This makes it very easy to deal with windup and mode switches. A problem with the incremental algorithm is that it cannot be used for controllers with P or PD action only. Therefore, ΔP has to be calculated in the following way when integral action is not used (see Section 3.4).

$$\Delta P(t_k) = K\big(by_{sp}(t_k) - y(t_k)\big) + u_b - u(t_{k-1})$$

Quantization and Word Length

A digital computer allows only finite precision in the calculations. It is sometimes difficult to implement the integral term on computers with a short word length. The difficulty is understood from Equation (3.31) for the integral term. The correction terms $b_{i1}e(t_{k+1}) + b_{i2}e(t_k)$ are normally small in comparison to $I(t_k)$, and they may be rounded off unless the word length is sufficiently large. This rounding-off effect gives an offset, called integration offset. To get a feel for the orders of magnitude involved, assume that we use the backward approximation and that all signals are normalized to have a largest magnitude of one. The correction term $Kh/T_i \cdot e$ in Equation (3.29) then has the largest magnitude Kh/T_i. Let the sampling period h be 0.02 s, the integral time $T_i = 20$ min $= 1200$ s and the gain $K = 0.1$. Then,

$$\frac{Kh}{T_i} = 1.7\ 10^{-6} = 2^{-19.2}$$

To avoid rounding off the correction term, it is thus necessary to have a precision of at least 20 bits. More bits are required to obtain meaningful numerical values. The situation is particularly important when a stepping motor that outputs increments is used. It is then necessary to resort to special tricks to avoid rounding off the integral. One simple way is to use a longer sampling period for the integral term. For instance, if a sampling period of 1 s is used instead of 0.02 s in the previous example, a precision of 14 bits is sufficient.

Three-Position Pulse Output

In Section 3.4, it was mentioned that the PID controller may have different types of outputs. We now return to the three-position pulse output and give a more detailed description of its implementation.

If a valve is driven by a constant-speed electrical motor, the valve can be in three states: "increase," "stop," and "decrease." Control of valves with electrical actuators is performed with a controller output that can be in three states. Three-position pulse output is performed using two digital outputs from the controller. When the first output is conducting, the valve position will increase. When the second output is conducting, the valve position will decrease. If none of the outputs are conducting, the valve position is constant. The two outputs must never be conducting at the same time.

There is normally both a dead zone and a dead time in the controller to ensure that the change of direction of the motor is not too frequent and not too fast. It means that the controller output is constant as long as the magnitude of the control error is within the dead zone, and that the output is stopped for a few seconds before it is allowed to change direction.

A servomotor is characterized by its running time T_{run}, which is the time it takes for the motor to go from one end position to the other. Since the servomotor has a constant speed, it introduces an integrator in the control loop, where the integration time is determined by T_{run}. A block diagram describing a PID controller with three-position pulse output combined with an electrical actuator is shown in Figure 3.33. Suppose that we have a steady-state situation, where the output from the PID controller u is equal to the position v of the servo-motor. Suppose further that we suddenly want to increase the controller output by an amount Δu. As long as the increase-output is conducting, the output v from the servo-motor will increase according to

$$\Delta v = \frac{1}{T_{\text{run}}} \int_0^t 1\, dt = \frac{t}{T_{\text{run}}}$$

To have Δv equal to Δu, the integration must be stopped after time

$$t = \Delta u T_{\text{run}}$$

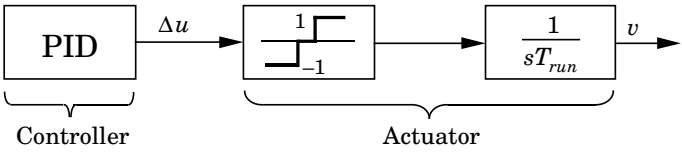

Figure 3.33 A PID controller with three-position pulse output combined with an electrical actuator.

In a digital controller, this means that the digital output corresponding to an increasing valve position is to be conducting for n sampling periods, where n is given by

$$n = \frac{\Delta u T_{\text{run}}}{h}$$

and where h is the sampling period of the controller.

To be able to perform a correct three-position pulse output, two buffers (`Buff_increase` and `Buff_decrease`) must be used to hold the number of pulses that should be sent out. The following is a computer code for three-position pulse output. For the sake of simplicity, details such as dead zone and dead time are omitted in the code.

```
if  delta_u > 0 then
    if valve_is_increasing then
        Buff_increase = Buff_increase + n;
    else
        Buff_decrease = Buff_decrease - n;
        if Buff_decrease < 0 then
            Buff_increase = - Buff_decrease;
            Buff_decrease = 0;
            valve_is_decreasing = false;
            valve_is_increasing = true;
        end;
    end;
else if delta_u < 0 then
    if valve_is_decreasing then
        Buff_decrease = Buff_decrease + n;
    else
        Buff_increase = Buff_increase - n;
        if Buff_increase < 0 then
            Buff_decrease = - Buff_increase;
            Buff_increase = 0;
            valve_is_increasing = false;
            valve_is_decreasing = true;
        end;
    end;
end;
if Buff_increase > 0 then
    Increaseoutput = 1;
    Decreaseoutput = 0;
    Buff_increase = Buff_increase - 1;
else if Buff_decrease > 0 then
    Increaseoutput = 0;
    Decreaseoutput = 1;
    Buff_decrease = Buff_decrease - 1;
end;
```

According to Figure 3.33, the controller output is Δu instead of u in the case of three-position pulse output. The integral part of the control algorithm is outside the controller, in the actuator. This solution causes no problems if the control algorithm really contains an integral part. P and PD control can not be obtained without information of the valve position (see Figure 3.17.)

3.7 Operational Aspects

Practically all controllers can be run in two modes: manual or automatic. In manual mode the controller output is manipulated directly by the operator, typically by pushing buttons that increase or decrease the controller output. A controller may also operate in combination with other controllers, such as in a cascade or ratio connection, or with nonlinear elements, such as multipliers and selectors. This gives rise to more operational modes. The controllers also have parameters that can be adjusted in operation. When there are changes of modes and parameters, it is essential to avoid switching transients. The way the mode switchings and the parameter changes are made depends on the structure chosen for the controller.

Bumpless Transfer Between Manual and Automatic

Since the controller is a dynamic system, it is necessary to make sure that the state of the system is correct when switching the controller between manual and automatic mode. When the system is in manual mode, the control algorithm produces a control signal that may be different from the manually generated control signal. It is necessary to make sure that the two outputs coincide at the time of switching. This is called *bumpless transfer*.

Bumpless transfer is easy to obtain for a controller in incremental form. This is shown in Figure 3.34. The integrator is provided with a switch so that the signals are either chosen from the manual or the automatic increments. Since the switching only influences the increments there will not be any large transients.

A similar mechanism can be used in the series, or interacting, implementation of a PID controller shown in Figure 3.8 (see Figure 3.35). In this case there will be a switching transient if the output of the PD part is not zero at the switching instant.

For controllers with parallel implementation, the integrator of the PID controller can be used to add up the changes in manual mode. The controller shown in Figure 3.36 is such a system. This system gives a smooth transition between manual and automatic mode provided

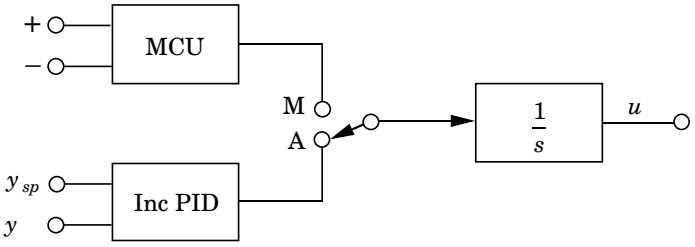

Figure 3.34 Bumpless transfer in a controller with incremental output. MCU stands for manual control unit.

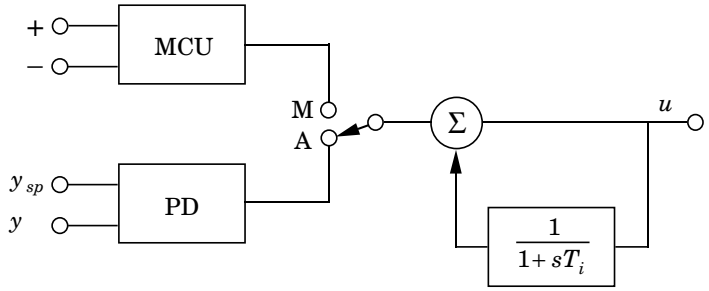

Figure 3.35 Bumpless transfer in a PID controller with a special series implementation.

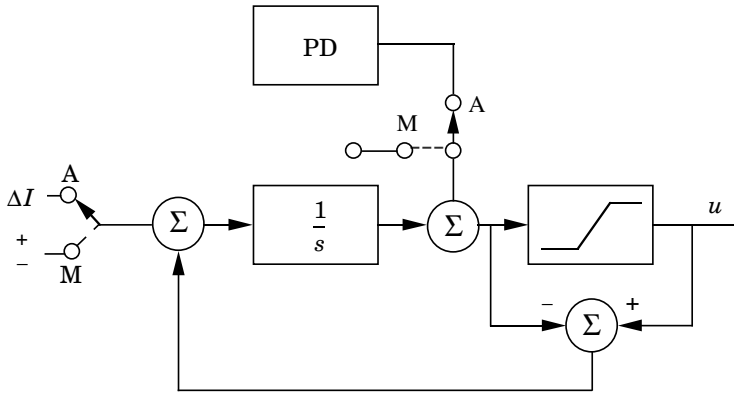

Figure 3.36 A PID controller where one integrator is used both to obtain integral action in automatic mode and to sum the incremental commands in manual mode.

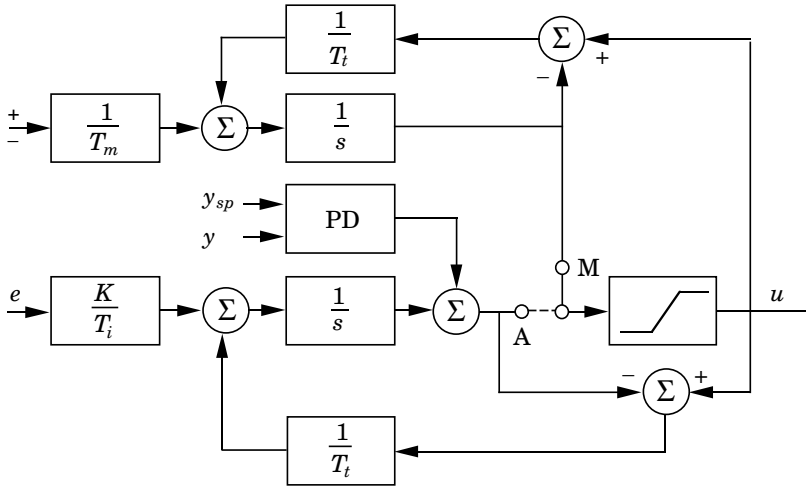

Figure 3.37 PID controller with parallel implementation that switches smoothly between manual and automatic control.

that the switch is made when the output of the PD block is zero. If this is not the case, there will be a switching transient.

It is also possible to use a separate integrator to add the incremental changes from the manual control device. To avoid switching transients in such a system, it is necessary to make sure that the integrator in the PID controller is reset to a proper value when the controller is in manual mode. Similarly, the integrator associated with manual control must be reset to a proper value when the controller is in automatic mode. This can be realized with the circuit shown in Figure 3.37. With this system the switch between manual and automatic is smooth even if the control error or its derivative is different from zero at the switching instant. When the controller operates in manual mode, as is shown in Figure 3.37, the feedback from the output v of the PID controller tracks the output u. With efficient tracking the signal v will thus be close to u at all times. There is a similar tracking mechanism that ensures that the integrator in the manual control circuit tracks the controller output.

Bumpless Parameter Changes

A controller is a dynamical system. A change of the parameters of a dynamical system will naturally result in changes of its output. Changes in the output can be avoided, in some cases, by a simultaneous change of the state of the system. The changes in the output will also depend on the chosen realization. With a PID controller it

is natural to require that there be no drastic changes in the output if the parameters are changed when the error is zero. This will hold for all incremental algorithms because the output of an incremental algorithm is zero when the input is zero, irrespective of the parameter values. It also holds for a position algorithm with the structure shown in Figure 3.8. For a position algorithm it depends, however, on the implementation. Assume that the state is chosen as

$$x_I = \int^t e(\tau)d\tau$$

when implementing the algorithm. The integral term is then

$$I = \frac{K}{T_i} x_I$$

Any change of K or T_i will then result in a change of I. To avoid bumps when the parameters are changed, it is essential that the state be chosen as

$$x_I = \int^t \frac{K(\tau)}{T_i(\tau)} e(\tau)d\tau$$

when implementing the integral term.

With sensible precautions, it is easy to ensure bumpless parameter changes if parameters are changed when the error is zero. There is, however, one case where special precautions have to be taken, namely, if setpoint weighting (Equation 3.20) is used. To have bumpless parameter changes in such a case it is necessary that the quantity $P + I$ be invariant to parameter changes. This means that when

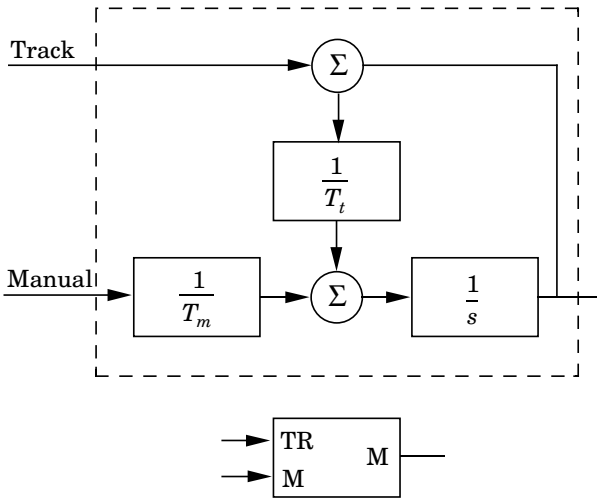

Figure 3.38 Manual control module.

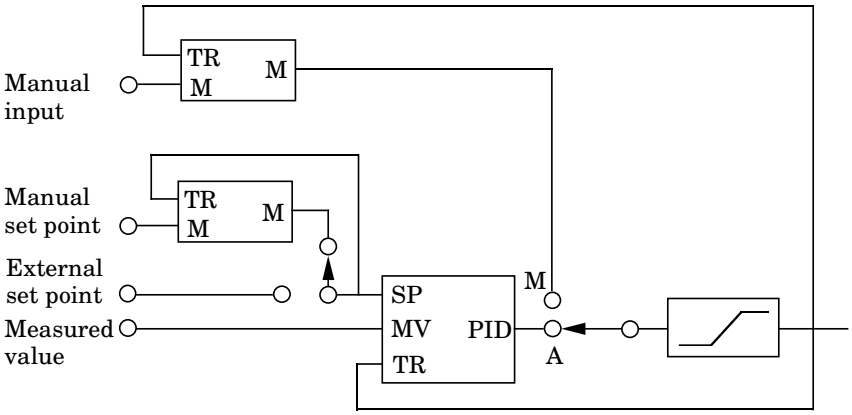

Figure 3.39 A reasonably complete PID controller with anti-windup, automatic-manual mode, and manual and external set-point.

parameters are changed, the state I should be changed as follows

$$I_{\text{new}} = I_{\text{old}} + K_{\text{old}}(b_{\text{old}}\, y_{sp} - y) - K_{\text{new}}(b_{\text{new}}\, y_{sp} - y) \tag{3.41}$$

To build automation systems it is useful to have suitable modules. Figure 3.38 shows the block diagram for a manual control module. It has two inputs, a tracking input and an input for the manual control commands. The system has two parameters, the time constant T_m for the manual control input and the reset time constant T_t. In digital implementations it is convenient to add a feature so that the command signal accelerates as long as one of the increase-decrease buttons are pushed. Using the module for PID control and the manual control module in Figure 3.38, it is straightforward to construct a complete controller. Figure 3.39 shows a PID controller with internal or external setpoints via increase/decrease buttons and manual automatic mode. Notice that the system only has two switches.

Computer Code

As an illustration, the following is a computer code for a PID algorithm. The controller handles both anti-windup and bumpless transfer.

```
"Compute controller coefficients
bi=K*h/Ti                  "integral gain
ad=(2*Td-N*h)/(2*Td+N*h)
bd=2*K*N*Td/(2*Td+N*h)     "derivative gain
a0=h/Tt
```

```
"Bumpless parameter changes
I=I+Kold*(bold*ysp-y)-Knew*(bnew*ysp-y)

"Control algorithm
r=adin(ch1)               "read setpoint from ch1
y=adin(ch2)               "read process variable from ch2
P=K*(b*ysp-y)             "compute proportional part
D=ad*D-bd*(y-yold)        "update derivative part
v=P+I+D                   "compute temporary output
u=sat(v,ulow,uhigh)       "simulate actuator saturation
daout(ch1)                "set analog output ch1
I=I+bi*(ysp-y)+ao*(u-v)   "update integral
yold=y                    "update old process output
```

The computation of the coefficients should be done only when the controller parameters are changed. Precomputation of the coefficients ad, ao, bd, and bi saves computer time in the main loop. The main program must be called once every sampling period. The program has three states: yold, I, and D. One state variable can be eliminated at the cost of a less readable code. Notice that the code includes derivation of the process output only, proportional action on part of the error only ($b \neq 1$), and anti-windup.

3.8 Commercial Controllers

Commercial PID controllers differ in the structure of the control law (standard-series-parallel, absolute-velocity), the parameterization, the limitation of high-frequency gain (filtering), and in how the setpoint is introduced. To be able to tune a controller, it is necessary to know the structure and the parameterization of the control algorithm. This information is, unfortunately, not usually available in the controller manuals. In this section, we have tried to summarize the properties of controllers from some different manufacturers.

Different structures of the PID algorithm were presented in Section 3.4. To summarize the results we introduce $U(s)$, $Y(s)$, and $Y_{sp}(s)$ as the Laplace transforms of process input u, process output y, and setpoint y_{sp}. Furthermore let $E(s) = Y_{sp}(s) - Y(s)$ denote the Laplace transform of the control error. Three different structures are used in the commercial controllers. The standard form, or ISA form, is given by

$$\text{I.} \quad U = K\left(bY_{sp} - Y + \frac{1}{sT_i}E + \frac{sT_d}{1 + sT_d/N}\left(cY_{sp} - Y\right)\right)$$

the series form is given by

$$\text{II.}\quad U = K'\left(\left(b+\frac{1}{sT_i'}\right)\frac{1+scT_d'}{1+sT_d'/N}Y_{sp} - \left(1+\frac{1}{sT_i'}\right)\frac{1+sT_d'}{1+sT_d'/N}Y\right)$$

and the parallel form by

$$\text{III.}\quad U = K''(bY_{sp}-Y) + \frac{K_i''}{s}E + \frac{K_d''s}{1+sK_d''/(NK'')}(cY_{sp}-Y)$$

The relations between the different parameters are discussed in Section 3.4. Recall that the parameters b and c are the weightings that influence the setpoint response. The values of b and c used are typically 0 or 1 in commercial controllers. This does not use the power of setpoint weighting fully as was discussed in Section 3.4. The setpoint weight factors b and c are chosen differently in different commercial controllers.

The high-frequency gain of the derivative term is limited to avoid noise amplification. This gain limitation can be parameterized in terms of the parameter N.

The sampling period is an important parameter of a digital PID controller, which limits how fast processes can be controlled. The values used in commercial controllers vary significantly.

Table 3.2 summarizes the properties of some common commercial PID controllers.

3.9 When Can PID Control Be Used?

The requirements on a control system may include many factors, such as response to command signals, insensitivity to measurement noise and process variations, and rejection of load disturbances. The design of a control system also involves aspects of process dynamics, actuator saturation, and disturbance characteristics. It may seem surprising that a controller as simple as the PID controller can work so well. The general empirical observation is that most industrial processes can be controlled reasonably well with PID control provided that the demands on the performance of the control are not too high. In the following paragraphs we delve further into this issue by first considering cases where PID control is sufficient and then discussing some generic problems where more sophisticated control is advisable.

When Is PI Control Sufficient?

Derivative action is frequently not used. It is an interesting observation that many industrial controllers only have PI action and that in others the derivative action can be (and frequently is) switched off. It can be shown that PI control is adequate for all processes where

Table 3.2 Properties of the PID algorithms in some commercial controllers. The structures of the controllers are labeled ISA (I), series (II), and ideal (III).

Controller	Structure	Setpoint weighting		Derivative gain limitation	Sampling period
		b	c	N	(s)
Allen Bradley PLC 5	I, III	1.0	1.0	none	load dependent
Bailey Net 90	II, III	0.0 or 1.0	0.0 or 1.0	10	0.25
Fisher Controls Provox	II	1.0	0.0	8	0.1, 0.25, or 1.0
Fisher Controls DPR 900, 910	II	0.0	0.0	8	0.2
Fisher Porter Micro DCI	II	1.0	0.0 or 1.0	none	0.1
Foxboro Model 761	II	1.0	0.0	10	0.25
Honeywell TDC	II	1.0	1.0	8	0.33, 0.5, or 1.0
Moore Products Type 352	II	1.0	0.0	$1-30$	0.1
Alfa Laval Automation ECA40, ECA400	II	0.0	0.0	8	0.2
Taylor Mod 30	II	0.0 or 1.0	0.0	17 or 20	0.25
Toshiba TOSDIC 200	II	1.0	1.0	$3.3-10$	0.2
Turnbull TCS 6000	II	1.0	1.0	none	$0.036-1.56$
Yokogawa SLPC	I	0.0 or 1.0	0.0 or 1.0	10	0.1

the dynamics are essentially of the first order (level controls in single tanks, stirred tank reactors with perfect mixing, etc). It is fairly easy to find out if this is the case by measuring the step response or the frequency response of the process. If the step response looks like that of a first-order system or, more precisely, if the Nyquist curve lies in the first and the fourth quadrants only, then PI control is sufficient. Another reason is that the process has been designed so that its operation does not require tight control. Then, even if the process has higher-order dynamics, what it needs is an integral action to provide zero steady-state offset and an adequate transient response by proportional action.

When Is PID Control Sufficient?

Similarly, PID control is sufficient for processes where the dominant dynamics are of the second order. For such processes there are no benefits gained by using a more complex controller.

A typical case of derivative action improving the response is when the dynamics are characterized by time constants that differ in magnitude. Derivative action can then profitably be used to speed up the response. Temperature control is a typical case. Derivative control is also beneficial when tight control of a higher-order system is required. The higher-order dynamics would limit the amount of proportional gain for good control. With a derivative action, improved damping is provided, hence, a higher proportional gain can be used to speed up the transient response.

When Is More Sophisticated Control Needed?

The benefits of using a more sophisticated controller than the PID is demonstrated by some examples below.

Higher-Order Processes

When the system is of a higher order than two, the control can be improved by using a more complex controller than the PID controller. This is illustrated by the following example.

EXAMPLE 3.3 Control of a higher-order process

Consider a third-order process described by the following transfer function.

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

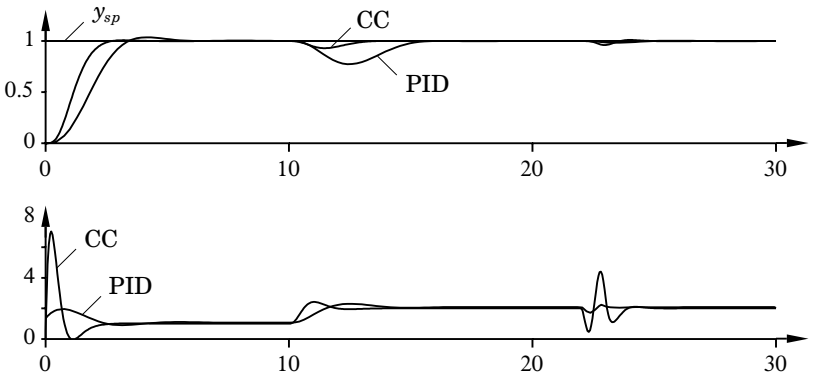

Figure 3.40 Control of the third-order system in Example 3.3 using a PID controller (PID) and a more complex controller (CC). The figure shows responses to a setpoint change, a load disturbance, and finally a measurement disturbance. The upper diagram shows setpoint y_{sp} and measurement signal y, and the lower diagram shows control signal u.

Figure 3.40 shows the control obtained using a PID controller and a more complex controller of higher order. The PID controller has the parameters $K = 3.4, T_i = 2.0$ and $T_d = 0.6$. The PID controller is compared with a controller of the form

$$R(s)u(t) = -S(s)y(t) + T(s)y_{sp}(t)$$

with the following controller polynomials

$$\begin{aligned} R(s) &= s(s^2 + 11.5s + 57.5) \\ S(s) &= 144s^3 + 575s^2 + 870s + 512 \\ T(s) &= 8s^3 + 77s^2 + 309s + 512 \end{aligned}$$

The benefits of using a more complex controller in the case of higher-order dynamics is clearly demonstrated in the figure. □

Systems with Long Dead Time

Control of systems with a dominant time delay are notoriously difficult. It is also a topic on which there are many different opinions concerning the merit of PID control. There seems to be general agreement that derivative action does not help much for processes with dominant time delays. For open-loop stable processes, the response to command signals can be improved substantially by introducing dead-time compensation. The load disturbance rejection can also be improved to some degree because a dead-time compensator allows a higher loop gain than a PID controller. Systems with dominant time delays are thus candidates for more sophisticated control.

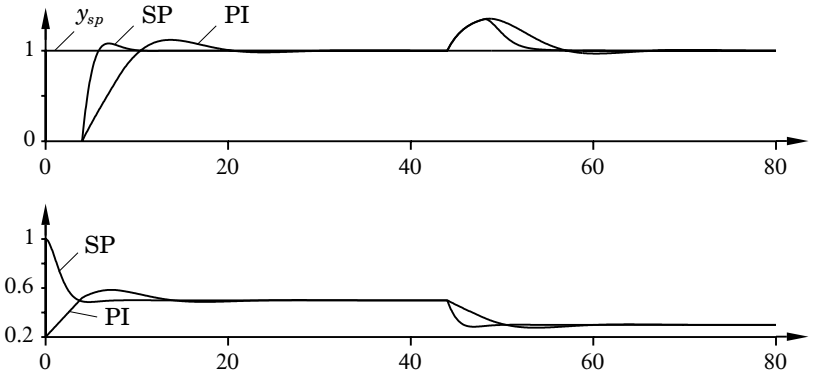

Figure 3.41 Control of the system in Example 3.4 with PI control (PI) and with a Smith predictor (SP). The upper diagram shows setpoint y_{sp} and measurement signal y, and the lower diagram shows control signal u.

EXAMPLE 3.4 Dead-time compensation

Consider a process described by the equation

$$\frac{dy(t)}{dt} = -0.5y(t) + 0.5u(t-4)$$

The process has a time constant of 2 and a time delay of 4 time units. This process was first controlled by a PI controller with a gain of 0.2 and an integral time of 2.5 (see Figure 3.41). The figure also shows the properties of the control obtained with a Smith predictor. The response to setpoint changes is much improved, while the difference is less for the load disturbance. When dead-time compensation was used, the gain in the PI controller was increased to $K = 1$, and the integral time was $T_i = 1$. □

Systems with Oscillatory Modes

Systems with oscillatory modes that occur when there are inertias and compliances is another case where PID control is not sufficient. There are several approaches to systems of that type. In the so-called notch filter approach, no attempt is made to damp the oscillatory modes, but an effort is made to reduce the signal transmission through the controller by a filter that drastically reduces signal transmission at the resonant frequency. A PID controller may be used when there is only one dominant oscillatory mode. Notch filter action can be achieved by judicious choice of the controller parameters. In this case, parameters T_i and T_d should be chosen so that the numerator has complex roots. The interacting form in Equation (3.15) does not work well in this case.

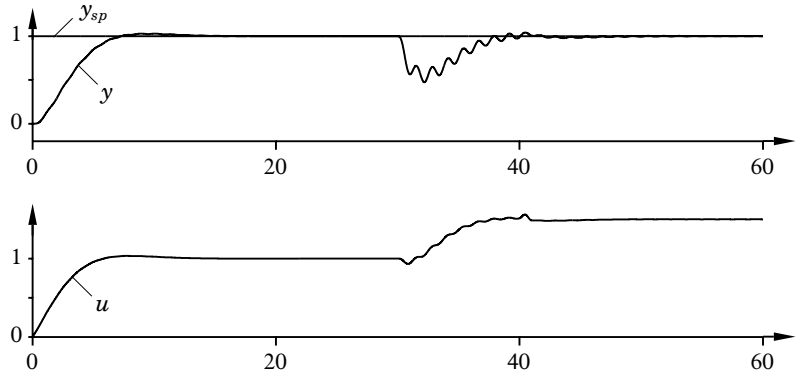

Figure 3.42 Response of the closed-loop system to setpoint and load disturbances. The graphs show setpoint y_{sp}, process output y, and control signal u. The controller parameters are $K = -0.25$, $T_i = -1$, and $b = 0$.

EXAMPLE 3.5 PI control of a system with oscillatory modes

Consider for example a process with the transfer function

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

where $a = 1$ and $b = 5$. The process has two undamped oscillatory poles. If these poles are neglected, the process is simply a first-order system that can conveniently be controlled by a PI controller. Attempting to control the process with a PI controller, we find that controller parameters K and T_i have to be negative. Reasonable values of the parameters are $K = -0.25$ and $T_i = -1$. Figure 3.42 shows the response of the closed-loop system to setpoint and load disturbances. Notice that the setpoint command does not excite the oscillatory poles so much. These modes are clearly visible, however, in the load disturbance response. With a nonzero b the setpoint changes will also excite the oscillatory modes, as is seen in Figure 3.43. □

The system in Example 3.5 gives only moderate damping of the oscillatory modes. For systems where the oscillatory modes are inside the servo bandwidth, it is necessary to have a controller with complex zeros. Such a controller can provide damping of the oscillatory modes because the poles will be attracted to the controller zeros. The controller zeros are the zeros of the function

$$1 + \frac{1}{sT_i} + sT_d = \frac{T_d}{s}\left(s^2 + \frac{1}{T_d}s + \frac{1}{T_iT_d}\right) \tag{3.42}$$

Assume that the zeros correspond to the polynomial

$$s^2 + 2\zeta\omega s + \omega^2$$

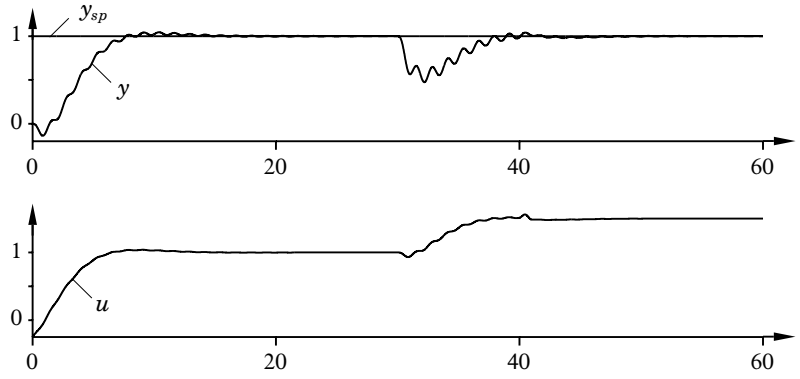

Figure 3.43 Response of the closed-loop system to setpoint and load disturbances. The graphs show setpoint y_{sp}, process output y, and control signal u. The controller parameters are $K = -0.25$, $T_i = -1$, and $b = 1$.

we find

$$\begin{aligned} \omega T_i &= 2\zeta \\ \omega T_d &= \frac{1}{2\zeta} \end{aligned} \tag{3.43}$$

Hence

$$\frac{T_i}{T_d} = 4\zeta^2 \tag{3.44}$$

The value of ζ used typically has to be quite small, say $\zeta = 0.2$, which gives $T_i/T_d = 0.16$. This ratio is significantly different from the commonly used value 4. Also, notice that a controller with $T_i < 4T_d$ can not be realized using the series form. To deal with oscillatory systems it is thus essential that the parallel form is used.

The above calculation is based on a simplified PID controller. For a controller where the derivative term has a limited high-frequency gain, Equations (3.42) and (3.43) are replaced by

$$1 + \frac{1}{sT_i} + \frac{sT_d}{1 + sT_d/N}$$

and

$$\begin{aligned} \omega T_i &= \pm\zeta + \sqrt{\zeta^2 - \frac{1}{N+1}} \\ \omega T_d &= \mp\left(\zeta - \sqrt{\zeta^2 - \frac{1}{N+1}}\right) \end{aligned} \tag{3.45}$$

It is desirable to have N as small as possible, this value is

$$N = \frac{1}{\zeta^2} - 1$$

which gives

$$\begin{aligned} \omega T_i &= \zeta \\ \omega T_d &= N\zeta \end{aligned} \tag{3.46}$$

Hence

$$\frac{T_i}{T_d} = \frac{1}{N} = \frac{\zeta^2}{1-\zeta^2} \tag{3.47}$$

For systems with oscillatory modes, the normal situation is that T_i is much smaller than T_d. Notice also that the choice of parameter N is critical for these applications.

Summary

When the dynamics of the process to be controlled are simple, a PID controller is sufficient. When the dynamics become more complicated, the control performance can be improved by using a more sophisticated controller structure than the PID. Examples of such processes have been given above. We end this section with some additional examples.

For some systems with large parameter variations it is possible to design linear controllers that allow operation over a wide parameter range. Such controllers are, however, often of high order.

The control of process variables that are closely related to important quality variables may be of a significant economic value. In such control loops it is frequently necessary to select the controller with respect to the disturbance characteristics. This often leads to strategies that are not of the PID type. These problems are often associated with time delays.

A general controller attempts to model the disturbances acting on the system. Since a PID controller has limited complexity, it cannot model complex disturbance behavior in general nor periodic disturbances in particular.

3.10 Conclusions

A detailed presentation of the PID algorithm has been given. Several modifications of the "textbook" version must be made to obtain a practical, useful controller. Problems that must be handled are, for example, integral wind-up and introduction of setpoint values. In a computer implementation, a discrete version of the PID algorithm is needed. Several methods to derive discrete PID algorithms have been described. Additional problems due to the sampling procedure must

be handled, such as the design of a prefilter to avoid aliasing. A discussion of the limitations of the PID algorithm and a characterization of processes where the PID controller manages to perform the control have also been given.

3.11 References

Proportional feedback in the form of a centrifugal governor was used to regulate the speed of windmills around 1750. In 1788 James Watt used a similar system for speed control of steam engines. The benefits of integral action was discovered a little later. Feedback control with proportional and integral action was rediscovered many times after that. In the early stages, the development of controllers was closely related to development of sensors and actuators. Sensing, actuation, and control were often combined in the same device.

The PID controller, in the form we know it today, emerged in the period from 1915 to 1940. It coincided with the development of legendary control companies such as Bristol, Fisher, Foxboro, Honeywell, Leeds & Northrup, and Taylor Instrument. Proportional and integral action had been used for a long time. Integral action was often called automatic reset, because it replaced a manual reset that was used in proportional controllers to obtain the correct steady state value. The potential of a controller that could anticipate future control errors was discussed in the 1920s. However, it took some time before the idea could be implemented. A controller with derivative action was introduced by Ralph Clarridge of the Taylor Instrument Company in 1935. At that time the function was called "pre-act." An interesting overview of the early history of PID controllers is given in (Stock, 1987 88). There is also much interesting material in publications from the instrument companies. The interview with Nichols, who is one of the pioneers in our field, in (Blickley, 1990) gives a perspective on the early development.

It is interesting to observe that feedback was crucial for the construction of the controller itself. The early pneumatic systems used the idea that an essentially linear controller can be obtained by a feedback loop composed of linear passive components and a nonlinear amplifier, the flapper valve. Similar ideas were used in electronic controllers with electric motors and relays.

Many of the practically useful modifications of the controller first appeared as special hardware functions. They were not expressed in mathematical form. An early mathematical analysis of a steam engine with a governor was made in (Maxwell, 1868). This analysis clearly showed the difference between proportional and integral control. The papers (Minorsky, 1922), (Küpfmüller, 1928), (Nyquist, 1932), and

(Hazen, 1934) were available at the time when the PID controller was developed. However, there is little evidence that the engineers in the process control field knew about them. Process control, therefore, developed independently. Two of the early papers were (Grebe *et al.*, 1933), written by engineers at the Dow Chemical Company, (Ivanoff, 1934), (Callender *et al.*, 1936), and (Hartree *et al.*, 1937).

The PID controller has gone through an interesting development because of the drastic technology changes that have happened since 1940. The pneumatic controller improved drastically by making systematic use of the force balance principle. Pneumatics was replaced by electronics when the operational amplifier appeared in the 1950s. A very significant development took place with the emergence of computer control in the 1960s. In the early computer control systems the computer commanded the setpoints of analog controllers. The next stage of the development was direct digital control (DDC), where the computer was controlling the actuator directly, see (Webb, 1967). A digital computer was then used to implement many PID controllers. This development led to a reconsideration of much of the fundamentals of PID control, see e.g. (Goff, 1966b), (L&N, 1968), (Moore *et al.*, 1970), and (Palmor and Shinnar, 1979). The appearance of microprocessors in the 1970s made it possible to use digital control for single loop controllers, see (Stojić and Petrović, 1986). It also led to the development of distributed control systems for process control, where the PID controller was a key element, see (Lukas, 1986). As the computing power of the microprocessors increased it was possible to introduce tuning and adaptation in the single loop controllers. This development started in the 1980s and has accelerated in the 1990s, see (Åström *et al.*, 1993).

It is interesting to observe that many facts about PID control were rediscovered in connection with the shifts in technology. One reason being that many practical aspects of PID control were considered as proprietory information that was not easily accessible in public literature. Much useful information was also scattered in the literature.

In spite of their wide spread use PID controllers are only treated superficially in many textbooks and at university courses. The book (Shinskey, 1988) gives a good coverage. Implementation issues are discussed in (Goff, 1966b), (Takahashi *et al.*, 1972), (Clarke, 1984), (Åström and Wittenmark, 1990). The paper (Åström and Steingrímsson, 1991) describes an implementation on a digital signal processor, which admits a very high sampling rate. The usefulness of a two-degree-of-freedom structure is discussed in (Horowitz, 1963). The application to PID control is treated in (Shigemasa *et al.*, 1987).

The phenomena of integral windup was well known in the early analog implementations. The controller structures used were often

such that windup was avoided. The anti-windup schemes were rediscovered in connection with the development of direct digital control. This is discussed in (Fertik and Ross, 1967). Much work on avoiding windup have been done since then, and windup has now made its way into some text books of control, see (Åström and Wittenmark, 1984). There are many papers written on the windup phenomena, see (Kramer and Jenkins, 1971), (Glattfelder and Schaufelberger, 1983), (Krikelis, 1984), (Gallun *et al.*, 1985), (Kapasouris and Athans, 1985), (Glattfelder and Schaufelberger, 1986), (Howes, 1986), (Åström, 1987b), (Hanus *et al.*, 1987), (Chen and Wang, 1988), (Glattfelder *et al.*, 1988), (Hanus, 1988), (Zhang and Evans, 1988), (Åström and Rundqwist, 1989), (Rundqwist, 1990), and (Walgama and Sternby, 1990). Mode switching is treated in the paper (Åström, 1987b).

The Smith predictor for control of systems with long time delays was presented in (Smith, 1957). The papers (Ross, 1977) and (Meyer *et al.*, 1976) compare the Smith predictor with the PID controller.

Controller Design

4.1 Introduction

This chapter describes some methods for determining the parameters of a PID controller. To obtain rational methods for designing controllers it is necessary to define the main purpose of the control system. This is done in Section 4.2.

The design methods differ with respect to the knowledge of the process dynamics they require. A PI controller is described by two parameters (K and T_i) and a PID controller by three or four parameters (K, T_i, T_d, and N). The classical Ziegler-Nichols methods are discussed in Section 4.3. In these methods process dynamics are characterized by two parameters. One parameter is related to the process gain and the other describes how fast the process is. In the step response method, the parameters are simple characteristics obtained from the step response. In the frequency response method, the parameters are the ultimate gain and the ultimate frequency.

An obvious extension of the frequency response method is to develop methods that are based on more knowledge of the open-loop transfer function, e.g., the slope of the transfer function or its values at two or more frequencies. In Section 4.4 we discuss various methods that are based on attempts to shape the loop transfer function. Section 4.5 treats analytical design methods, where the controller transfer function is obtained from the specifications and the process transfer function by a direct calculation.

One possibility for compromise between several different criteria is to use optimization methods. This is discussed in Section 4.6. Another way to characterize process dynamics with few parameters is to use low-order dynamic models with few parameters. Such methods are discussed in Section 4.7 where the design goal is to position all the poles of the closed-loop system. It is shown that methods based on dynamic models of first and second order lead to PI and PID controllers.

Instead of attempting to position all closed-loop poles, it can be attempted to assign only a few dominating poles. Such methods are dis-

cussed in Section 4.8. The approach leads to systematic design methods and a unification of many other techniques. New simple design methods based on the dominant pole design method are presented in Chapter 5.

In Section 4.9, design methods based on disturbance rejection are presented. Finally, conclusions and references are given in Sections 4.10 and 4.11.

4.2 Specifications

When solving a control problem it is necessary to understand what the primary goal of control is. Two common types of problems are to follow the setpoint and to reject disturbances. It is also important to have an assessment of the major restrictions, which can be

- System dynamics
- Nonlinearities
- Disturbances
- Process uncertainty

Typical specifications on a control system may include

- Attenuation of load disturbances
- Sensitivity to measurement noise
- Robustness to model uncertainty
- Setpoint following

Specifications can be expressed in many different ways. Features of time responses for typical inputs is one possibility. Features of frequency responses or transfer functions are other possibilities. Some of the specifications such as attenuation and sensitivity to measurement errors, are conflicting, and others such as setpoint following and load disturbance rejection are nonconflicting.

For process control applications setpoint following is often less important than load disturbance attenuation. Setpoint changes are often only made when the production rate is altered. Furthermore, the response to setpoint changes can be improved by feeding the setpoint through ramping functions or by adjusting the setpoint weightings described in Section 3.4.

Load Disturbances

Load disturbances are disturbances that drive the process variables away from their desired values. Attenuation of load disturbances is of primary concern for process control. This is particularly the case

for regulation problems where the processes are running in steady state with constant setpoint for a long time. Load disturbances are often of low frequencies. Step signals are often used as prototype disturbances. The disturbances may enter the system in many different ways. If nothing else is known, it is often assumed that the disturbances enter at the process input. Typical responses due to a unit step disturbance at the process input are shown in Figure 4.1. The characteristics of the graphs in Figure 4.1 are often used to specify the response to load disturbances. Let e be the error caused by a unit step disturbance at the process input. Typical quantities used to characterize the error are: maximum error $e_{\max}$, time to reach maximum $t_{\max}$, settling time t_s, decay ratio d, and the integrated absolute error, which is defined by

$$IAE = \int_0^\infty |e(t)|dt \tag{4.1}$$

The criterion IAE is in many cases a natural choice, at least for control of quality variables. A severe drawback is that its evaluation requires significant computations or a simulation of the process. The simulation must also be made with sufficient accuracy. Since the criterion is based on an infinite integral it is also necessary to simulate for a long time.

For processes that are nonoscillatory, IAE is the same as the integrated error

$$IE = \int_0^\infty e(t)dt \tag{4.2}$$

The quantity IE is a good approximation of IAE for systems that are oscillatory but well damped. The reason for using IE is that its value is directly related to the controller parameters. To see this assume that the control law is

$$u(t) = ke(t) + k_i \int_0^t e(t)dt - k_d \frac{dy}{dt}$$

and that this controller gives a stable closed-loop system. Furthermore assume that the error is initially zero and that a unit step disturbance is applied at the process input. Since the closed-loop system is stable and has integral action the control error will go to zero. We thus find

$$u(\infty) - u(0) = k_i \int_0^\infty e(t)dt$$

Since the disturbance is applied at the process input, the change in control signal is equal to the change of the disturbance. Hence,

$$IE = \int_0^\infty e(t)dt = \frac{1}{k_i} = \frac{T_i}{K} \tag{4.3}$$

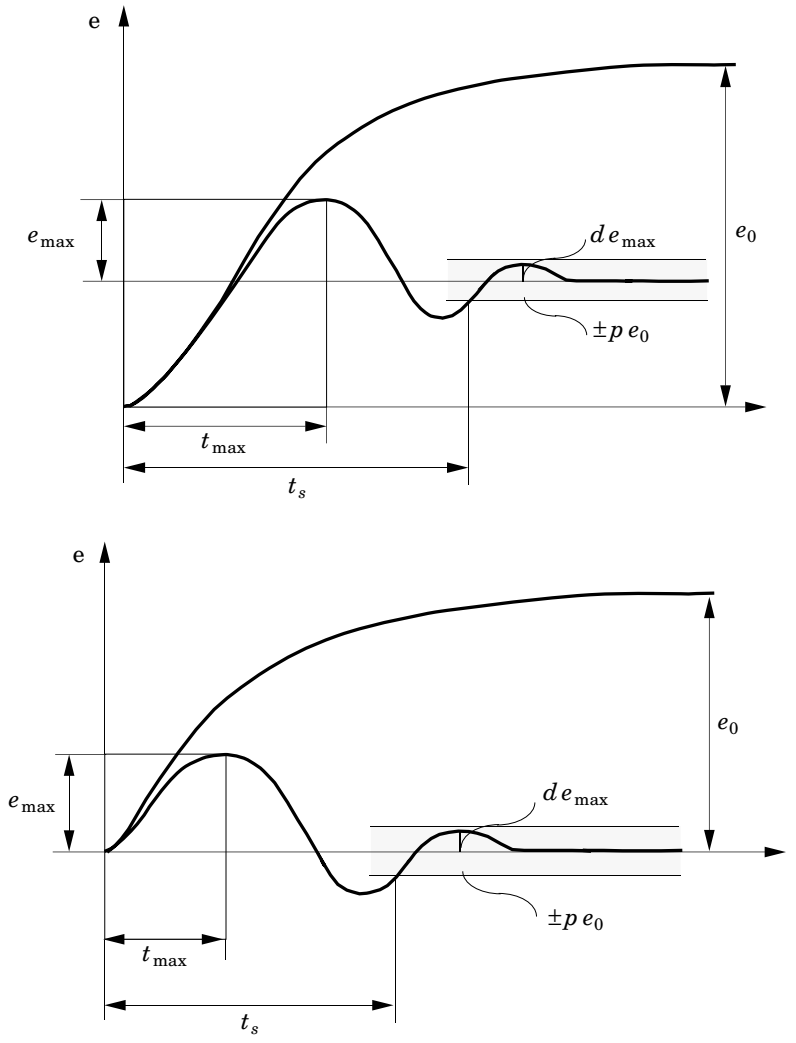

Figure 4.1 The error due to a unit step load disturbance at the process input and some features used to characterize attenuation of load disturbances. The curves show the open-loop error and the error obtained using a controller without integral action (upper) and with integral action (lower).

Integral gain k_i is thus inversely proportional to IE.

The criterion IE is a natural choice for control of quality variables for a process where the product is sent to a mixing tank. The criterion may be strongly misleading, however, in other situations. It will be zero for an oscillatory system with no damping. It will also be zero for a controller with a double integrator.

The quadratic criterion

$$ISE = \int_0^\infty e^2(t)dt \tag{4.4}$$

is also easy to compute. It has, however, the disadvantage that it gives a very high weight to large errors, which often leads to a poorly damped closed loop.

Sensitivity to Measurement Noise

Measurement noise is typically of high frequency. Care should always be taken to reduce noise by appropriate filtering. For sampled systems it is also important to choose the sampling rate properly. Measurement noise will be fed into the system through the feedback. It will generate control actions and control errors. The transmission of measurement noise to control actions can be described by the transfer function

$$G_{nu} = \frac{G_c}{1 + G_\ell} \tag{4.5}$$

where G_p is the process transfer function, G_c is the controller transfer function, and $G_\ell = G_p G_c$ is the loop transfer function. The transfer function from measurement noise to process output is

$$G_{ny} = \frac{1}{1 + G_\ell} = S \tag{4.6}$$

where S is called the sensitivity function. Since the magnitude of G_ℓ normally is small for high frequencies, we have approximately $G_{nu} = G_c$ for high frequencies.

The high-frequency gain of a PID controller is

$$K_{hf} = K(1 + N) \tag{4.7}$$

Notice that $N = 0$ corresponds to PI control. Multiplication of the measurement noise by K_{hf} gives the fluctuations in the control signal that are caused by the measurement noise. Also notice that there may be a significant difference in K_{hf} for PI and PID control. It is typically an order of magnitude larger for a PID controller, since the gain normally is higher for a PID controller than for a PI controller, and N is typically around 10.

Sensitivity to Process Characteristics

The controller parameters are typically matched to the process characteristics. Since the process may change it is important that the

controller parameters are chosen in such a way that the closed-loop system is not too sensitive to variations in process dynamics. There are many ways to specify the sensitivity. Many different criteria are conveniently expressed in terms of the Nyquist curve of the loop transfer function $G_\ell(s) = G_c(s)G_p(s)$ (see Figure 4.2). We choose to characterize sensitivity by

$$M_s = \max_{0\le\omega<\infty} \left|\frac{1}{1 + G_p(i\omega)G_c(i\omega)}\right| = \max_{0\le\omega<\infty} |S(i\omega)| \qquad (4.8)$$

Notice that the transfer function S, which is called the sensitivity function, also appeared in the expression for the sensitivity to measurement noise (compare with Equation (4.6)). The quantity M_s is simply the inverse of the shortest distance from the Nyquist curve to the critical point -1. Reasonable values of M_s are in the range from 1.3 to 2.

Sensitivity function S has many useful physical interpretations. One is the following. Assume that there is a sinusoidal disturbance with frequency ω that enters the system in an arbitrary way. Let the amplitude of the open-loop system be a_0. If the system is controlled with a controller that gives the sensitivity function S, the amplitude of the controlled system is then $a_0|S(i\omega)|$. Feedback thus reduces the

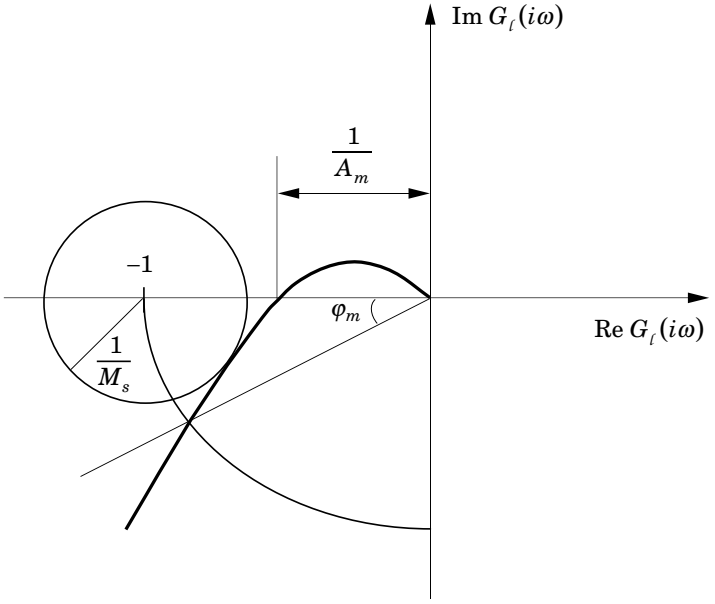

Figure 4.2 Definitions of sensitivity M_s, amplitude margin A_m, and phase margin φ_m. A sensitivity M_s guarantees that the distance from the critical point to the Nyquist curve is always greater than $1/M_s$.

effect of the disturbance if $|S(i\omega)| < 1$, and it amplifies a disturbance if $|S(i\omega)| > 1$.

Under very general assumptions it can be shown that the sensitivity can not be smaller than one for all frequencies. With a controller having integral action we have $|S(0)| = 0$. Low frequency disturbances thus can be reduced effectively with such a controller. When designing a controller it is important to be aware of the frequencies where disturbances are amplified. It is also important that the largest value of the sensitivity is limited. It is common to require that the maximum value of the sensitivity function, M_s be in the range of 1.3 to 2.

Amplitude margin (A_m) and phase margin (φ_m) are other common sensitivity measures. They are defined as

$$\begin{aligned} A_m &= \frac{1}{|G_\ell(i\omega_u)|} \\ \varphi_m &= \pi + \arg G_\ell(i\omega_g) \end{aligned} \tag{4.9}$$

where the ultimate frequency ω_u is the frequency where $\arg G_\ell(i\omega) = -\pi$ and the gain cross-over frequency ω_g is the frequency where $|G_\ell(i\omega)| = 1$. See Figure 4.2. The amplitude margin is also called gain margin. We have the following relations

$$\begin{aligned} A_m &> \frac{M_s}{M_s - 1} \\ \varphi_m &> 2 \arcsin \frac{1}{2M_s} \end{aligned} \tag{4.10}$$

Typical values of φ_m range from 30° to 60°. Amplitude margins could typically vary from 2 to 5. A geometrical interpretation of the criterion given by Equation (4.8) is that the Nyquist curve of the loop transfer function is always outside a circle around the critical point -1 with the radius $1/M_s$. An engineering interpretation is that the system remains stable even if the gain is increased by the factor $M_s/(M_s - 1)$ or decreased by the vector $M_s/(M_s + 1)$. The closed loop will remain stable even if a nonlinearity characterized by

$$xM_s/(M_s + 1) < f(x) < xM_s/(M_s - 1)$$

is inserted in the loop. A small value of M_s thus ensures that the system will remain stable in spite of nonlinear actuator characteristics.

Setpoint Following

Specifications on setpoint following may include requirements on rise time, settling time, decay ratio, overshoot, and steady-state offset for step changes in setpoint. See Figure 4.3. These quantities are defined in different ways and there are also different standards.

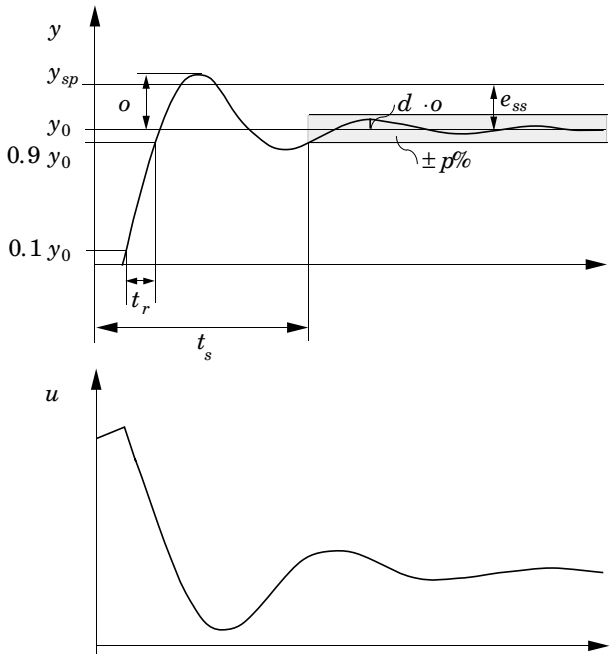

Figure 4.3 Specifications on setpoint following based on the time response to a unit step in the setpoint.

- The *rise time* t_r is either defined as the inverse of the largest slope of the step response or the time it takes the step to pass from 10% to 90% of its steady state value.
- The *settling time* t_s is the time it takes before the step response remains within $p\%$ of its steady state value. The value $p = 2\%$ is commonly used.
- The *decay ratio* d is the ratio between two consecutive maxima of the error for a step change in setpoint or load. The value $d = 1/4$, which is called quarter amplitude damping, has been used traditionally. This value is, however, too high as will be shown later.
- The *overshoot* o is the ratio between the difference between the first peak and the steady state value and the steady state value of the step response. In industrial control applications it is common to specify an overshoot of 8%–10%. In many situations it is desirable, however, to have an overdamped response with no overshoot.
- The *steady-state error* e_{ss} is the value of control error e in steady state. With integral action in the controller, the steady-state error is always zero.

Criteria like IAE, IE, and ISE can also be used to characterize setpoint responses if the error in Equations (4.1), (4.2), and (4.4) are interpreted as the error due to a unit step change of the setpoint. For step changes in the setpoint there will always be a large initial error. It is then useful to have criteria that put little weight on the initial error. It has been found that criteria of the type

$$ITAE = \int_0^\infty t\,|e(t)|\,dt$$

$$ITE = \int_0^\infty te(t)dt$$

$$ITSE = \int_0^\infty te^2(t)dt$$

$$ISTE = \int_0^\infty t^2e^2(t)dt$$

are more suitable to judge performance for setpoint following. These integrals are finite only if the steady-state error is zero. In some cases setpoint following may also contain requirements based on ramp signals.

For a system with pure error feedback the relation between process output and setpoint is given by

$$Y(s) = \frac{G_p(s)G_c(s)}{1 + G_p(s)G_c(s)}\,Y_{sp} = \frac{G_\ell(s)}{1 + G_\ell(s)}\,Y_{sp} \tag{4.11}$$

The setpoint response is thus uniquely given by G_ℓ. For systems with two degrees of freedom the corresponding relation is

$$Y(s) = \frac{G_p(s)G_{ff}(s)}{1 + G_p(s)G_c(s)}Y_{sp} = \frac{G_p(s)G_{ff}(s)}{1 + G_\ell(s)}Y_{sp} \tag{4.12}$$

where $G_{ff}(s)$ is the transfer function between the setpoint and the controller output. (Compare with Section 3.4.) Setpoint following and load disturbance rejection can be decoupled by using a two-degree-of-freedom structure. For PID controllers it is mostly the setpoint weighting that is used to modify the setpoint response.

To judge the properties of a control system we must consider both the process output and the control signal. The response of the control signal to a step change in the setpoint typically has an overshoot as is shown in Figure 4.3. The initial change of the control signal is $\Delta u(0) = Kb\Delta y_{sp}$, where K is the controller gain, b is the setpoint weighting, and Δy_{sp} is the setpoint change. The steady-state change of the control signal is $\Delta u_{ss} = \Delta y_{sp}/K_p$, where K_p is the static process gain. The quantity

$$M_u = \frac{\Delta u(0)}{\Delta u_{ss}} = KK_pb$$

is thus a normalized initial overshoot of the control signal. This expression is a correct value of the overshoot if the control signal has its largest value immediately after the change in the setpoint. For systems with time delays the maximum will occur later. The overshoot can then be approximated by the expression

$$M_u = K K_p \left(b + \frac{L}{T_i} \right)$$

where L is the apparent dead time and T_i is the integration time of the controller.

The dimensionless quantity M_u, which we call the control signal overshoot, is a quantity that is useful for evaluating the performance of a control system. For systems where an essential part of the dynamics is due to the sensors it is important that M_u is not too large.

Relations Between Specifications

Specifications express different properties of a system such as load disturbance attenuation and setpoint following. They are also expressed in different ways using frequency domain or time domain properties. To get some insight into the relations we will investigate a second-order system in detail.

Second-Order System

Consider a first-order system with the transfer function

$$G_p(s) = \frac{K_p}{1 + sT}$$

that is controlled by an I controller. For such a system the transfer function from setpoint to process output is

$$G(s) = \frac{G_p G_c}{1 + G_p G_c} = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{4.13}$$

The response to a unit step in the setpoint is

$$y(t) = 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_0 t} \sin\left(\omega_0 t \sqrt{1-\zeta^2} + \phi \right) \tag{4.14}$$

where $\phi = \arctan \sqrt{1-\zeta^2}/\zeta$.

The transfer function from a load disturbance at the process input to process output is given by

$$G_{ly}(s) = \frac{G_p}{1 + G_p G_c} = \frac{K_p}{T} \frac{s}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{4.15}$$

A unit step load disturbance at the process input gives the error

$$e(t) = \frac{K_p}{\omega_0 T\sqrt{1-\zeta^2}}\, e^{-\zeta\omega_0 t}\sin\omega_0 t\sqrt{1-\zeta^2} \tag{4.16}$$

If $0 < \zeta < 1$, the two closed-loop poles of the system are

$$p_i = -\zeta\omega_0 \pm i\omega_0\sqrt{1-\zeta^2}$$

where ζ is called the relative damping, and ω_0 is the undamped natural frequency. The time responses of the system are characterized by a damped oscillation with period

$$T_p = \frac{2\pi}{\omega_0\sqrt{1-\zeta^2}} \tag{4.17}$$

and decay ratio

$$d = e^{-2\pi\zeta/\sqrt{1-\zeta^2}} \tag{4.18}$$

From the step response (Equation 4.14) we can calculate the rise time, the settling time, and the overshoot. Defining the rise time as the inverse of the maximum slope of the step response we get

$$t_r = \frac{1}{\omega_0}\, e^{\phi/\tan\phi} \tag{4.19}$$

The settling time, i.e., the time required for the output to be within the fraction p of the steady state value, is a discontinuous function of the parameters due to the oscillatory nature of the step response. An approximative formula is obtained by considering the envelope of the step response. This gives

$$t_s \approx -\frac{\log\left(p\sqrt{1-\zeta^2}\right)}{\zeta\omega_0} \tag{4.20}$$

This formula is conservative because it overestimates the settling time. The slope has its maximum at

$$t = \frac{\phi}{\omega_0\cos\phi}$$

With $\zeta = 0.707$ and $p = 0.02$ we get $t_r = 2.2/\omega_0$ and $t_s = 6.0/\omega_0$. The overshoot is given by

$$o = e^{-\pi\zeta/\sqrt{1-\zeta^2}} = \sqrt{d}$$

where d is the decay ratio. It occurs at the time

$$t_{\max} = \frac{\pi}{\omega_0\sqrt{1-\zeta^2}}$$

The overshoot is 4% for $\zeta = 0.707$, 8% for $\zeta = 0.63$, 16% for $\zeta = 0.5$. It is 50% for quarter amplitude decay ratio.

Equation (4.16) gives the error signal after a step disturbance at the process input. This signal has the maximum

$$e_{\max} = \frac{K_p}{\omega_0 T\sqrt{1-\zeta^2}}\, e^{-\phi \tan \phi} \sin \phi$$

for

$$t_{\max} = \frac{1}{\omega_0\sqrt{1-\zeta^2}}\, \phi$$

The integrated error is

$$IE = \frac{K_p}{\omega_0^2 T}$$

This quantity is close to the IAE, if the overshoot is sufficiently small.

The high-frequency gain of the controller is

$$K_{hf} = \frac{2\zeta\omega_0 T - 1}{K_p} \approx \frac{2\zeta\omega_0 T}{K_p}$$

where the approximation holds when $\omega_0 T$ is large. The sensitivity is

$$M_s = \sqrt{\frac{1 + 8\zeta^2 + (1+4\zeta^2)\sqrt{1+8\zeta^2}}{1 + 8\zeta^2 + (-1+4\zeta^2)\sqrt{1+8\zeta^2}}}$$

The sensitivity function is infinite for $\zeta = 0$ and decreases with increasing ζ. Its values for $\zeta = 0.3$, 0.5, and 0.7 are 2.0, 1.5, and 1.3. To have a reasonable value of the sensitivity it must, therefore, be required that the relative damping is greater than 0.3. This implies that the decay ratio d must be smaller than 0.14.

The equations given can be used to understand how the properties of the closed-loop system are influenced by ω_0 and ζ. The integrated error is inversely proportional to ω_0^{-2}. The maximum error, the rise time, and the settling time are proportional to ω_0^{-1}. The high-frequency gain of the controller is proportional to ω_0. Both the load disturbance response and the setpoint response are improved by increasing ω_0. The control actions generated by noise do increase, however, with ω_0. The overshoot, the decay ratio, and the sensitivity will increase with decreasing ζ.

This general pattern also holds for more complex systems. Both load disturbance attenuation and response time to setpoint changes will generally increase with increasing bandwidth of the system. For more complex controllers, the load disturbance response and the setpoint response can be specified separately.

Averaging Control

There are several situations where the purpose of control is not to keep the process variables at constant values. Level control in buffer tanks is a typical example. The reason for using a buffer tank is to smooth flow variations. In such a case the tank level should fluctuate within some limits. It is often undesirable that the tank becomes empty or that it overflows. The specifications are thus that the tank level should be allowed to fluctuate between given limits. This is called averaging control. It is often solved with a controller with a small gain. Sometimes gain scheduling is introduced to have a larger gain when the level gets close to the limits. Another approach is to use error-squared control. This was discussed in Section 3.4.

Dominant Poles

The formulas derived above for a second-order system can often be used as approximations for more complex systems. The reason for this is that the dynamics of complex systems can often be characterized by a few poles. Many properties of the closed-loop system can be deduced from the poles and the zeros of

$$G(s) = \frac{G_\ell(s)}{1 + G_\ell(s)} \tag{4.21}$$

The closed-loop zeros are the same as the zeros of loop transfer function $G_e ll(s)$, and the closed-loop poles are the roots of the equation

$$1 + G_\ell(s) = 0 \tag{4.22}$$

The pole-zero configurations of closed-loop systems may vary considerably. Many simple feedback loops, however, will have a configuration of the type shown in Figure 4.4, where the principal characteristics

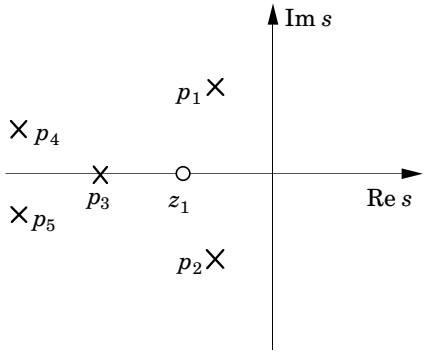

Figure 4.4 Pole-zero configuration of a simple feedback system.

of the response are given by a complex pair of poles, p_1 and p_2, called the *dominant poles*. The response is also influenced by real poles and zeros p_3 and z_1 close to the origin. The position of p_3 and z_1 may be reversed. There may also be more poles and zeros far from the origin, which typically are of less influence. Poles and zeros with real parts much smaller than the real part of the dominant poles have little influence on the transient response.

Complex poles are often characterized in terms of their frequency ω_0, which is the distance from the origin, and their relative damping ζ. If a pair of complex poles is dominating, the formulas derived above for a second-order system can be used as approximation. Classical control was very much concerned with closed-loop systems having the pole-zero configuration shown in Figure 4.4.

Even if many closed-loop systems have a pole-zero configuration similar to the one shown in Figure 4.4, there are, however, exceptions. For instance, systems with mechanical resonances, which may have poles and zeros close to the imaginary axis, are generic examples of systems that do not fit the pole-zero pattern of the figure. Another example is processes with a long dead time.

Determination of the Dominant Poles from the Frequency Response

A simple method for approximate determination of the dominant poles from knowledge of the Nyquist curve of the loop transfer function will now be given. Consider the loop transfer function $G_\ell(s)$ as a mapping from the s-plane to the G_ℓ-plane. The map of the imaginary axis in the s-plane is the Nyquist curve $G_\ell(i\omega)$, which is indicated in Figure 4.5.

The closed-loop poles are the roots of the characteristic equation

$$1 + G_\ell(s) = 0$$

The map of a straight vertical line through the dominant closed-loop poles in the s-plane is thus a curve through the critical point $G_\ell = -1$ in the G_ℓ-plane. This curve is shown by a dashed line in Figure 4.5. Since the map is conform, the straight line A′C′ is mapped on the curve AC, which intersects the Nyquist curve orthogonally. The triangle ABC is also mapped conformally to A′B′C′. If ABC can be approximated by a triangle, we have

$$\frac{G_\ell(i\omega_2) - G_\ell(i\omega_1)}{i\omega_2 - i\omega_1} \approx \frac{1 + G_\ell(i\omega_2)}{\sigma}$$

When ω_1 is close to ω_2 this becomes

$$\sigma = (1 + G_\ell(i\omega_2)) \frac{i\omega_2 - i\omega_1}{G_\ell(i\omega_2) - G_\ell(i\omega_1)} \approx \frac{1 + G_\ell(i\omega_2)}{G'_\ell(i\omega_2)}$$

where

$$G'_\ell(s) = \frac{dG_\ell(s)}{ds}$$

To determine the dominant poles we first determine the point A on the Nyquist curve that is closest to the ultimate point. Then we determine the derivative of the loop transfer function at that point or evaluate the transfer function at a neighboring point ω_1.

Design Parameters and Design Methods

In control designs it is often convenient to have a few parameters that can be changed to influence the performance of the system. The parameters should be chosen in such a way that their influence on the performance of the system is transparent. In the case of the second-order example discussed above, the parameters can be chosen as ω_0 and ζ. The relative damping can be replaced by the sensitivity M_s.

A good design method should take a number of different specifications into account in a balanced way. Most design methods, unfortunately, concentrate on one or a few of the specifications only.

4.3 Ziegler-Nichols' and Related Methods

Two classical methods for determining the parameters of PID controllers were presented by Ziegler and Nichols in 1942. These methods are still widely used, either in their original form or in some modification. They often form the basis for tuning procedures used by controller manufacturers and process industry. The methods are based on determination of some features of process dynamics. The

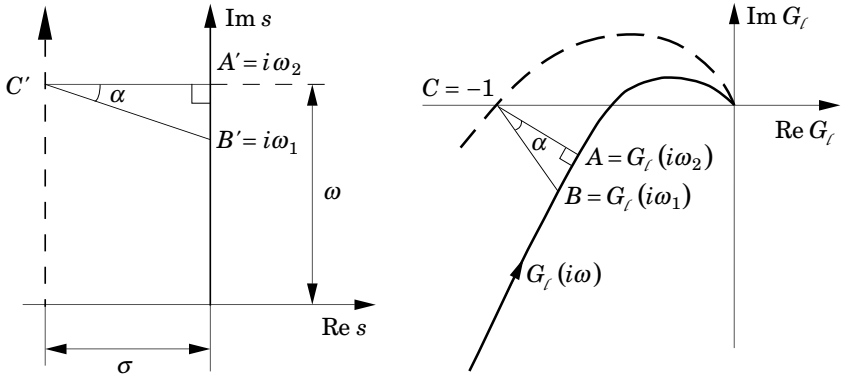

Figure 4.5 Representation of the loop transfer function as a map of complex planes.

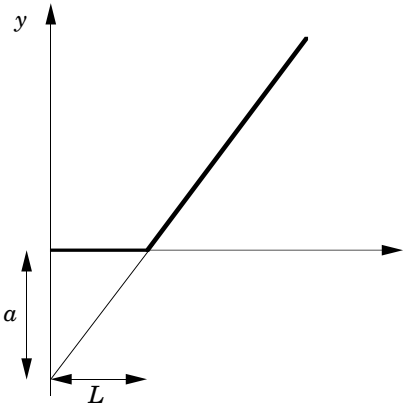

Figure 4.6 Characterization of a step response in the Ziegler-Nichols step response method.

controller parameters are then expressed in terms of the features by simple formulas.

The Step Response Method

The first design method presented by Ziegler and Nichols is based on a registration of the open-loop step response of the system, which is characterized by two parameters. The parameters are determined from a unit step response of the process, as shown in Figure 4.6.

The point where the slope of the step response has its maximum is first determined, and the tangent at this point is drawn. The intersections between the tangent and the coordinate axes give the parameters a and L. In Chapter 2, a model of the process to be controlled was derived from these parameters. This corresponds to modeling a process by an integrator and a time delay. Ziegler and Nichols have given PID parameters directly as functions of a and L. These are given in Table 4.1. An estimate of the period T_p of the closed-loop system is also given in the table.

EXAMPLE 4.1 Ziegler-Nichols step response method

Ziegler-Nichols method will be applied to a process with the transfer function

$$G(s) = \frac{1}{(s+1)^3} \tag{4.23}$$

Measurements on the step response give the parameters $a = 0.218$ and $L = 0.806$. The controller parameters can now be determined from Table 4.1. The parameters of a PID controller are $K = 5.50$,

Table 4.1 PID controller parameters obtained from the Ziegler-Nichols step response method.

Controller	K	T_i	T_d	T_p
P	$1/a$			$4L$
PI	$0.9/a$	$3L$		$5.7L$
PID	$1.2/a$	$2L$	$L/2$	$3.4L$

$T_i = 1.61$, and $T_d = 0.403$. Figure 4.7 shows the response of the closed-loop systems to a step change in setpoint followed by a step change in the load. The behaviour of the controller is as can be expected. The decay ratio for the step response is close to one quarter. It is smaller for the load disturbance. The overshoot in the setpoint response is too large. This can be improved by reducing parameter b. Compare with Section 3.4. □

The Frequency Response Method

This method is also based on a simple characterization of the process dynamics. The design is based on knowledge of the point on the Nyquist curve of the process transfer function $G(s)$ where the Nyquist curve intersects the negative real axis. For historical reasons this point is characterized by the parameters K_u and T_u, which are called the *ultimate gain* and the *ultimate period*. These parameters can be

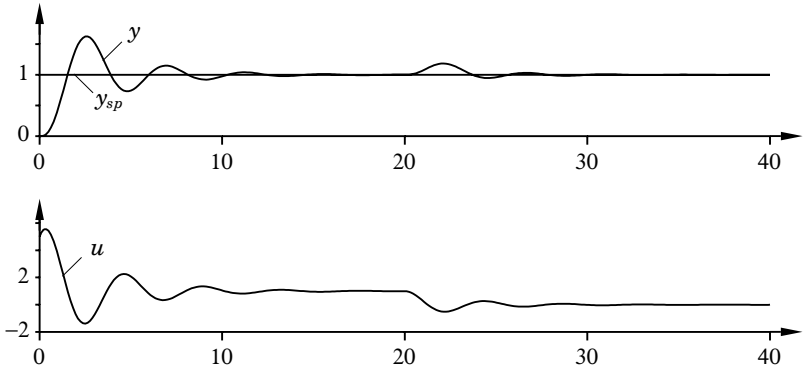

Figure 4.7 Setpoint and load disturbance response of a process with transfer function $1/(s+1)^3$ controlled by a PID controller tuned with the Ziegler-Nichols step response method. The diagrams show setpoint y_{sp}, process output y, and control signal u.

Table 4.2 PID controller parameters obtained from the Ziegler-Nichols frequency response method.

Controller	K	T_i	T_d	T_p
P	$0.5K_u$			T_u
PI	$0.4K_u$	$0.8T_u$		$1.4T_u$
PID	$0.6K_u$	$0.5T_u$	$0.125T_u$	$0.85T_u$

determined in the following way. Connect a controller to the process, set the parameters so that control action is proportional, i.e., $T_i = \infty$ and $T_d = 0$. Increase the gain slowly until the process starts to oscillate. The gain when this occurs is K_u and the period of the oscillation is T_u. The parameters can also be determined approximately by relay feedback as is discussed in Section 2.6.

Ziegler-Nichols have given simple formulas for the parameters of the controller in terms of the ultimate gain and the ultimate period (see Table 4.2). An estimate of the period T_p of the dominant dynamics of the closed-loop system is also given in the table.

We illustrate the design procedure with an example.

EXAMPLE 4.2 The Ziegler-Nichols frequency response method

Consider the same process as in Example 4.1. The process given by Equation (4.23) has the ultimate gain $K_u = 8$ and the ultimate period $T_u = 2\pi/\sqrt{3} \approx 3.63$. Table 4.2 gives the parameters $K = 4.8$, $T_i = 1.81$, and $T_d = 0.44$ for a PID controller. Figure 4.8 shows the closed-loop setpoint and load disturbance responses when the controller is applied to the process given by Equation (4.23). The parameters and the performance of the controllers obtained with the frequency response method are close to those obtained by the step response method. The responses are slightly better damped. □

The Ziegler-Nichols tuning rules were originally designed to give systems with good responses to load disturbances. They were obtained by extensive simulations of many different systems. The design criterion was quarter amplitude decay ratio. Equation (4.18) gives a relation between decay ratio d and relative damping ζ. Using this relation we find that $d = 1/4$, gives $\zeta = 0.22$, which is often too small, as is seen in the examples. For this reason the Ziegler-Nichols method often requires modification or retuning. Since the primary design objective was to reduce load disturbances, it is often necessary to choose setpoint weighting carefully in order to obtain a satisfactory setpoint response.

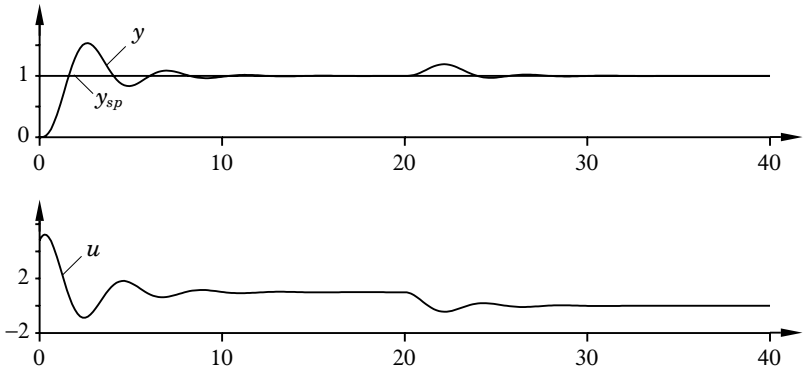

Figure 4.8 Setpoint and load disturbance response of a process with the transfer function $1/(s+1)^3$ controlled by a PID controller that is tuned with the Ziegler-Nichols frequency response method. The diagrams show setpoint y_{sp}, process output y, and control signal u.

Relations Between the Ziegler-Nichols Tuning Methods

Insight into the relations between the Ziegler-Nichols methods can be obtained by calculating the controller parameters for different systems. Consider a process with the transfer function

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

which is the model originally used by Ziegler and Nichols to derive their tuning rules for the step response method. For this process we have $a = bL$. The ultimate frequency is $\omega_u = \pi/2L$, which gives the ultimate period $T_u = 4L$, and the ultimate gain is $K_u = \pi/2bL$.

The step response method gives the following parameters for a PI controller

$$K = \frac{0.9}{bL}, \quad T_i = 3L$$

This can be compared with the parameters

$$K = \frac{0.63}{bL}, \quad T_i = 3.2L$$

obtained for the frequency response method. Notice that the integral times are within 10% but that the step response method gives a gain that is about 40% higher.

The PID parameters obtained from the step response method are

$$K = \frac{1.2}{bL}, \quad T_i = 2L \quad \text{and} \quad T_d = \frac{L}{2}$$

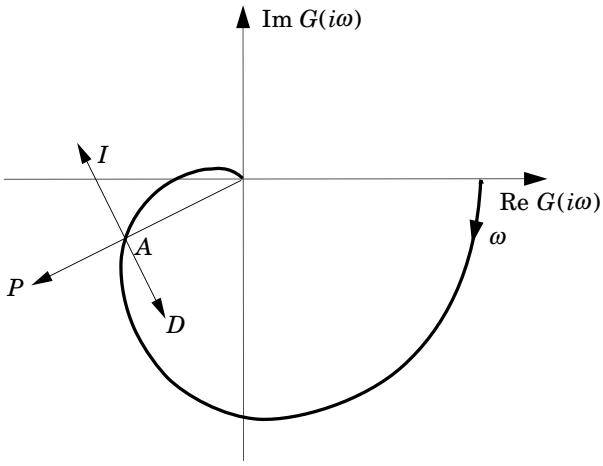

Figure 4.9 A given point on the Nyquist curve may be moved to another position in the G-plane by PI, PD, or PID control. Point A may be moved in the directions $G(i\omega)$, $-iG(i\omega)$, and $iG(i\omega)$ by changing the proportional, integral, and derivative gain, respectively.

and those given by the frequency response methods are,

$$K = \frac{0.94}{bL}, \quad T_i = 2L \quad \text{and} \quad T_d = \frac{L}{2}$$

In this particular case both methods give the same values of integral and derivative times but the step response method gives a gain that is about 25% higher than the frequency response method. The results of this example are quite typical. The step response method often gives higher values of the gain.

An Interpretation of the Frequency Domain Method

The frequency domain method can be interpreted as a method where one point of the Nyquist curve is positioned. With PI or PID control, it is possible to move a given point on the Nyquist curve to an arbitrary position in the complex plane, as indicated in Figure 4.9. By changing the gain, a point on the Nyquist curve is moved radially from the origin. The point can be moved in the orthogonal direction by changing integral or derivative gain. Notice that with positive controller parameters the point can be moved to a quarter plane with PI or PD control and to a half plane with PID control. The frequency response method starts with determination of the point $(-1/K_u, 0)$ where the Nyquist curve of the open-loop transfer function intersects the negative real axis. Let us now investigate how the ultimate point is changed by the

controller. For a PI controller with Ziegler-Nichols tuning we have $K = 0.4K_u$ and $\omega_u T_i = (2\pi/T_u)0.8T_u = 5.02$. Thererfore, the transfer function of the PI controller at the ultimate frequency is

$$G_c(i\omega_u) = K\left(1 + \frac{1}{i\omega_u T_i}\right) = 0.4K_u(1 - i/5.02) = K_u(0.4 - 0.08i)$$

The ultimate point is thus moved to $-0.4 + 0.08i$. This means that a lag of 11.2° is introduced at the ultimate frequency.

For a PID controller we have $K = 0.6K_u$, $\omega_u T_i = \pi$ and $\omega_u T_d = \pi/4$. The frequency response of the controller at frequency ω_u is

$$\begin{aligned} G_c(i\omega_u) &= K\left(1 + i\Big(\omega_u T_d - \frac{1}{\omega_u T_i}\Big)\right) = 0.6K_u\left(1 + i\Big(\frac{\pi}{4} - \frac{1}{\pi}\Big)\right) \\ &\approx 0.6K_u(1 + 0.467i) \end{aligned}$$

This controller gives a phase advance of 25° at the ultimate frequency. The loop transfer function is

$$G_\ell(i\omega_u) = G_p(i\omega_u)G_c(i\omega_u) = -0.6(1 + 0.467i) = -0.6 - 0.28i$$

The Ziegler-Nichols frequency response method thus moves the ultimate point $(-1/K_u, 0)$ to the point $-0.6 - 0.28i$. The distance from this point to the critical point is 0.5. This means that the method gives a sensitivity that is always greater than 2.

Modified Ziegler-Nichols Method

With the given interpretation of the frequency domain method, it is straightforward to generalize it in the following way. Choose an arbitrary point on the Nyquist curve of the open-loop system. Determine a controller that moves this point to a suitable location. Let the chosen point be

$$A = G_p(i\omega_0) = r_a e^{i(\pi + \phi_a)}$$

Determine a controller that moves this point to

$$B = G_\ell(i\omega_0) = r_b e^{i(\pi + \phi_b)}$$

Writing the frequency response of the controller as $G_c(i\omega_0) = r_c e^{i\phi_c}$ we get

$$r_b e^{i(\pi + \phi_b)} = r_a r_c e^{i(\pi + \phi_a + \phi_c)}$$

The controller should thus be chosen so that

$$\begin{aligned} r_c &= \frac{r_b}{r_a} \\ \phi_c &= \phi_b - \phi_a \end{aligned}$$

For a PI controller this implies

$$\begin{aligned} K &= \frac{r_b \cos(\phi_b - \phi_a)}{r_a} \\ T_i &= \frac{1}{\omega_0 \tan(\phi_a - \phi_b)} \end{aligned} \tag{4.24}$$

This means that we must require $\phi_a > \phi_b$ in order to have positive T_i. For a PID controller we get similarly

$$\begin{aligned} K &= \frac{r_b \cos(\phi_b - \phi_a)}{r_a} \\ \omega_0 T_d - \frac{1}{\omega_0 T_i} &= \tan(\phi_b - \phi_a) \end{aligned} \tag{4.25}$$

The gain K is uniquely given. There is, however, only one equation to determine parameters T_i and T_d. An additional condition must thus be introduced to determine these parameters uniquely. A common method is to specify that the ratio of these parameters is constant, i.e.,

$$T_d = \alpha T_i$$

as in the Ziegler-Nichols rules, where $\alpha = 0.25$. Straightforward calculations then give

$$\begin{aligned} T_i &= \frac{1}{2\alpha\omega_0}\left(\tan(\phi_b - \phi_a) + \sqrt{4\alpha + \tan^2(\phi_b - \phi_a)}\right) \\ T_d &= \alpha T_i \end{aligned} \tag{4.26}$$

Assuming that a Ziegler-Nichols experiment is used to determine a suitable point, we have $r_a = 1/K_u$ and $\phi_a = 0$. The PI controller parameters then become

$$\begin{aligned} K &= K_u r_b \cos\phi_b \\ T_i &= -\frac{T_u}{2\pi \tan\phi_b} \end{aligned} \tag{4.27}$$

Notice that ϕ_b must be negative in order to have positive controller parameters. Choosing $\alpha = 0.25$, the PID controller parameters are given by

$$\begin{aligned} K &= K_u r_b \cos\phi_b \\ T_i &= \frac{T_u}{\pi}\left(\frac{1 + \sin\phi_b}{\cos\phi_b}\right) \\ T_d &= \frac{T_u}{4\pi}\left(\frac{1 + \sin\phi_b}{\cos\phi_b}\right) \end{aligned} \tag{4.28}$$

Notice that the tuning rules are of the same form as for the frequency response method but with different values of the numerical parameters. Systems with better damping than the Ziegler-Nichols rules can

be obtained by proper choices of r_b and ϕ_b. A reasonable choice is $r_b = 0.5$ and $\phi_b = 20°$.

It has been suggested by Pessen to move the ultimate point to $-0.2 - 0.36i$ or $-0.2 - 0.21i$. This corresponds to $r_b = 0.41$ and $\phi_b = 61°$, and $r_b = 0.29$ and $\phi_b = 46°$ respectively.

There are limitations with a design method where only one point on the Nyquist curve is positioned. The properties of the closed-loop system can then change significantly depending on the slope of the curve. This is illustrated in Figure 4.10, which shows the Nyquist curves of three systems having the same amplitude margin, $A_m = 2$, which means that the Nyquist curves of all systems pass through the point $(-0.5, 0)$. The figure also shows the closed-loop responses to a step change in setpoint.

Assessment of Ziegler-Nichols Tuning

The Ziegler-Nichols tuning procedures are simple and intuitive. They require little process knowledge and they can be applied with modest effort. These are some of the reasons why they are so widely used. The methods have, however, some limitations as we have already seen. A fundamental drawback is that the basic design criterion is to obtain a closed-loop system with quarter amplitude decay ratio ($d = 0.25$). This gives good rejection of load disturbances, but also creates a closed-loop system that is very poorly damped and that has poor stability margins. The closed-loop gain is typically 2 to 3 times too high. The frequency response method is more reliable than the step response method. One reason for this is that the ultimate gain is uniquely defined, but that there are many ways to define the apparent dead time. The step response method typically also gives somewhat higher gains.

The methods generally will work better for PID control than for PI control. The reason for this will be discussed in Chapter 5. Let it suffice here to give an example.

EXAMPLE 4.3 PI control

Consider the same process as in Examples 4.1 and 4.2, where the transfer function has three equal lags. See Equation (4.23). Measurements on the step response give the parameters $a = 0.218$ and $L = 0.806$. The step response method gives a PI controller with parameters $K = 4.13$ and $T_i = 2.42$. The ultimate gain is $K_u = 8$ and the ultimate period is $T_u = 2\pi/\sqrt{3} \approx 3.63$. The frequency domain method gives a PI controller with parameters $K = 3.2$ and $T_i = 2.90$. Notice that the gains obtained with the frequency response method are lower than those obtained with the step response method. Figure

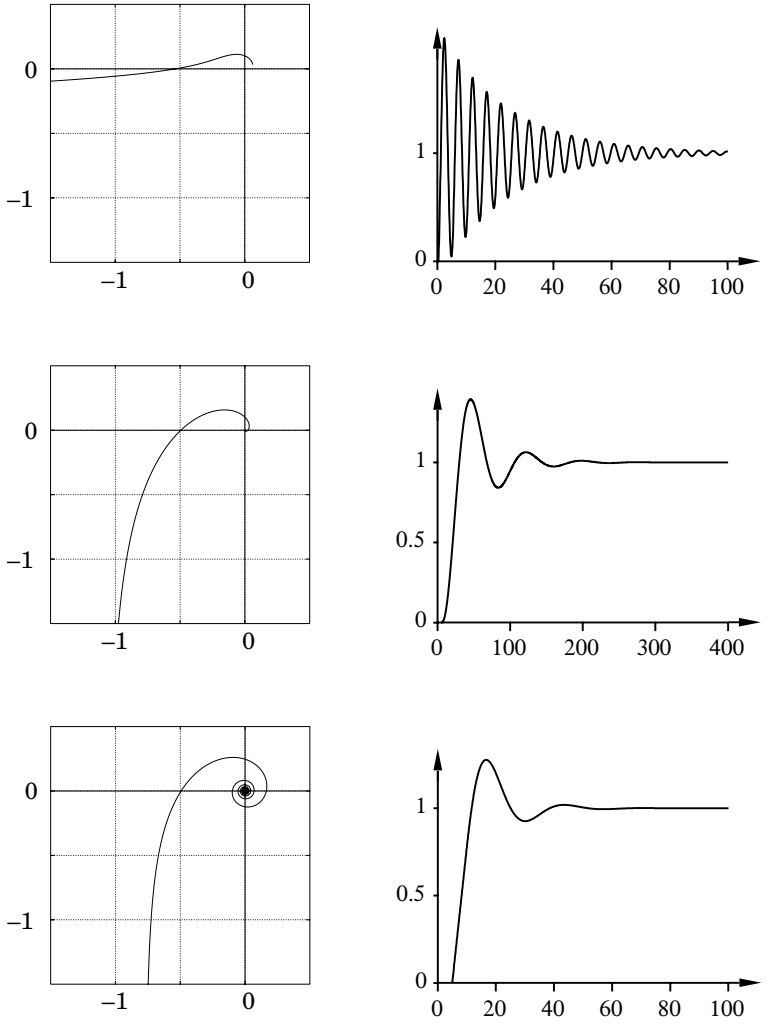

Figure 4.10 Nyquist curves of three systems with amplitude margin $A_m = 2$, and their corresponding closed-loop step responses.

4.11 shows the response of the closed-loop system to step changes in setpoint and load when the PI controller is tuned with the frequency response method. The figure shows clearly that the decay ratio is much larger than the design value $d = 1/4$. The performance is even worse if the step response method is used. Compare with Figure 4.8 which shows the results obtained with a PID controller tuned by the Ziegler-Nichols frequency response method. □

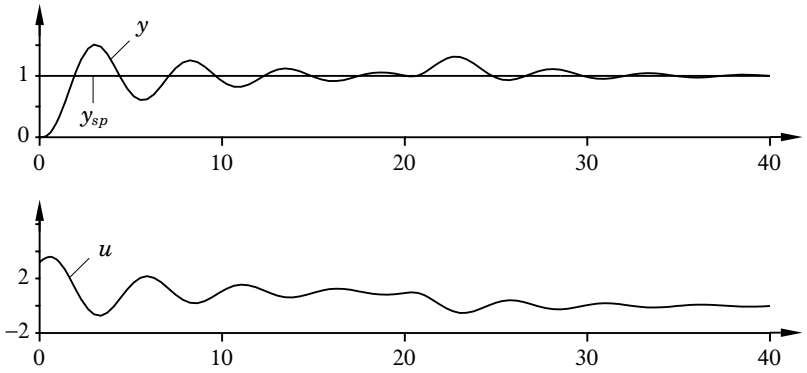

Figure 4.11 Setpoint and load disturbance response of a process with transfer function $G(s) = (s+1)^{-3}$ controlled by a PI controller tuned with the Ziegler-Nichols frequency response method. The diagrams show setpoint y_{sp}, process output y, and control signal u.

Although the Ziegler-Nichols methods have many attractive properties they are far from perfect. Hence, there is a need to characterize those situations where reasonable tuning is obtained with the Ziegler-Nichols method and also to estimate the achievable performance. For this purpose the process will be characterized by the quantities normalized dead time τ and gain ratio κ introduced in Chapter 2. Recall that τ is the ratio of apparent dead time and average residence time and that κ is the ratio of the process gains at frequencies ω_u and 0. Also remember that both quantities normally vary from 0 to 1 and that they are approximately linearly related. Furthermore processes with small κ or τ are easy to control. The difficulty increases as the parameters approach 1.

The following empirical rules have been developed based on simulation of a large number of systems. There is no precise definition of the region of validity. Roughly speaking they apply to processes with essentially monotone step responses.

Case 1: Small κ and τ. Processes with small κ or τ are easy to control. A small value of τ means that the dynamics is lag dominated. In this case there are factors other than process dynamics that limits performance, e.g., measurement noise. If specifications on response time are not severe, satisfactory performance can often be obtained with a PI controller. The tuning obtained by the Ziegler-Nichols methods can often be improved significantly by using other methods. Derivative action or even more complicated control laws are often useful for obtaining systems with high performance in those cases where the disturbances are small. Notice that the Ziegler-Nichols rules do not give guidance for finding parameters in PD controllers.

Case 2: Intermediate κ and τ. This is the primary range for using the Ziegler-Nichols method for PID control. Derivative action often gives significant improvement of performance. The overshoot for setpoint changes can often be too large. It can be reduced by proper choice of setpoint weighting.

Case 3: κ and τ close to 1. This case corresponds to processes that are dead time dominated. The Ziegler-Nichols tuning rules do not perform well in those cases. PI or PID control can still be used, but the tuning rules must be improved. It is also possible to get drastically improved setpoint responses by using other control algorithms like the Smith predictor. (Compare with Example 3.4 in Section 3.9.)

The boundaries between the different cases are approximately 0.07 and 0.4 for κ, or 0.15 and 0.4 for τ. The following example illustrates that the Ziegler-Nichols rules give poor tuning in Case 3.

EXAMPLE 4.4 Ziegler-Nichols tuning for κ and τ close to 1

Consider a process with the transfer function

$$G(s) = \frac{e^{-5s}}{(s+1)^3}$$

Applying the frequency response method we find that $K_u = 1.25$ and $T_u = 15.7$. The controller parameters then become $K = 0.75$, $T_i = 7.9$ and $T_d = 2.0$. The normalized dead time varies between 0.6 and to 0.7 depending on the method used to compute it. Compare with Section 2.4. The gain ratio is, however, uniquely defined and becomes $\kappa = 0.8$. This case thus belongs to Case 3 above. Figure 4.12 shows a simulation of the setpoint and load responses of the closed-loop system. The responses are oscillatory as can be expected. Notice also that the recovery from load disturbances is slow because the integral action is too small. □

Achievable Performance

It is also of interest to characterize the performance that can be achieved with Ziegler-Nichols tuned PID controllers. A first indication is already given in Table 4.1 and Table 4.2, which give the period of the closed-loop systems. Several empirical observations have been made from experimental investigations of tuned loops.

The rise time obtained is approximately equal to the apparent dead time for processes without integration and L/2 for processes with integration.

The error due to a step disturbance at the process input has a maximum at a time that is approximately equal to $0.25T_u$ or L. The size of the peak is approximately $1.4K_p\kappa$, where K_p is the static process gain. Notice that the error is proportional to κ.

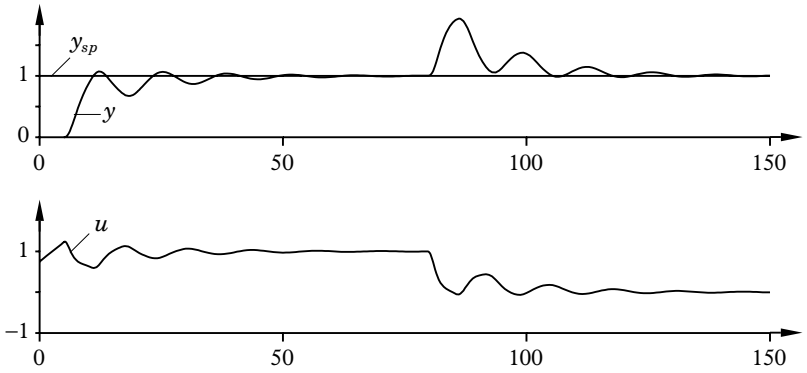

Figure 4.12 Step responses of a process with the transfer function $G(s) = e^{-5s}/(s+1)^3$ controlled by PID controllers tuned with the Ziegler-Nichols frequency response method. The diagrams show setpoint y_{sp}, process output y, and control signal u.

With Ziegler-Nichols tuning the sensitivity is always larger than 2. In Section 4.2 it was shown that a quarter amplitude decay ratio corresponds to a sensitivity $M_s = 2.6$.

Tuning Maps

Since the Ziegler-Nichols methods only give "ball-park" values, it is necessary to make manual tuning to obtain the desired performance. A device called tuning maps have been developed to guide manual tuning. The purpose of these maps is to provide intuition about how changes in controller parameters influence the behaviour of the closed-loop system. The tuning maps are simply two-dimensional arrays of transient responses or frequency responses organized in a systematic way.

An example of a tuning map is given in Figure 4.13. The figure illustrates how the load disturbance response is influenced by changes in gain and integral time. The process model

$$G(s) = \frac{1}{(s+1)^8}$$

has been used in the example. The Ziegler-Nichols frequency response method gives the controller parameters $K = 1.13$, $T_i = 7.58$, and $T_d = 1.9$. The figure shows clearly the benefits of having a smaller value of T_i. Judging from the figure, the values $K = 1$ and $T_i = 5.0$ appear reasonable. The figure also shows that the choice of T_i is fairly critical. Also notice that controllers with $T_i < 7.6$ cannot be implemented on series form (compare with Section 3.4).

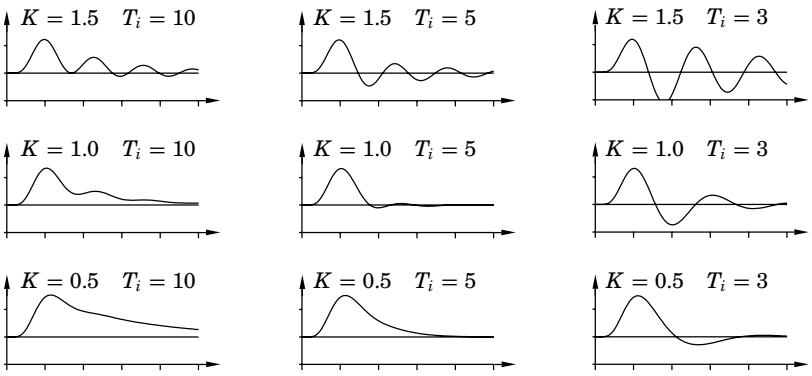

Figure 4.13 Tuning map for PID control of a process with the transfer function $G(s) = (s+1)^{-8}$. The figure shows the responses to a unit step disturbance at the process input. Parameter T_d has the value 1.9.

Another example of a tuning map is given in Figure 4.14, which shows the Nyquist curves of the loop transfer functions that correspond to Figure 4.13. The figure shows that with Ziegler-Nichols tuning there is too much phase lead. This is reduced by reducing parameter T_i. A comparative study of curves like Figure 4.13 and Figure 4.14 is a good way to develop intuition for the relations between the time domain and the frequency domain.

It is useful to have a simple way to judge if the integral action of a controller is too weak, as in the three left and the lower middle examples in Figures 4.13 and 4.14. Such a criterion can be based on a calculation of the asymptotic behaviour of the loop transfer function for low frequencies. For a process with transfer function G_p and a PI controller with transfer function G_c we have

$$\begin{aligned} G_\ell(s) &= G_p(s)G_c(s) \\ &\approx \left(G_p(0) + sG_p'(0)\right) K \left(1 + \frac{1}{sT_i}\right) \\ &\approx \frac{KG_p(0)}{sT_i} + KG_p(0) + \frac{KG_p'(0)}{T_i} \end{aligned}$$

Thus, for low frequencies the asymptote of the Nyquist curve is parallel to the imaginary axis with the real part equal to

$$KG_p(0) + \frac{KG_p'(0)}{T_i} = KK_p\left(1 - \frac{T_{ar}}{T_i}\right)$$

where $K_p = G(0)$ is the static process gain, and T_{ar} is the average residence time. It is reasonable to require that the real part of the

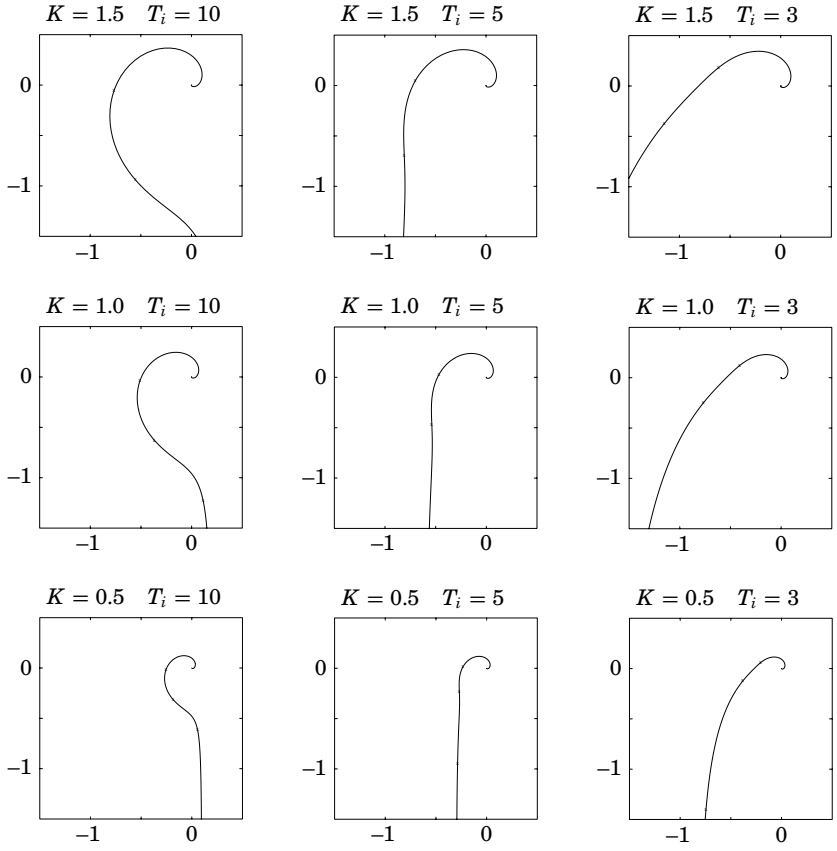

Figure 4.14 Tuning map for PID control of a process with the transfer function $G(s) = (s + 1)^{-8}$. The figure shows the Nyquist curves of the loop transfer function. Parameter T_d has the value 1.9.

asymptote is less than -0.5. This gives

$$T_i < T_{ar} \frac{2KK_p}{1 + 2KK_p} \tag{4.29}$$

For the system in Figures 4.13 and 4.14, we get the requirement $T_i < 6.0$ for the systems in the upper row, $T_i < 5.3$ for the systems in the middle row, and $T_i < 4.0$ for the systems in the lower row. This means that condition (4.29) excludes the three left and the lower middle examples in Figures 4.13 and 4.14.

Assuming that the process dynamics is governed by

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

we find that $T_{ar} = L + T$. With Ziegler-Nichols tuning of a PI controller, we then find that condition (4.29) is satisfied if

$$3L < (L + T)\frac{1.8T}{L + 1.8T}$$

This means that we must require that the normalized dead time satisfies

$$\tau = \frac{L}{L + T} < 0.28$$

A similar calculation for a process described by

$$G_p(s) = K_p \frac{e^{-sL}}{(1 + sT)^2}$$

which has

$$\tau = \frac{L + (3 - e)T}{L + 2T}$$

shows that condition (4.29) holds for a PI controller tuned according to the Ziegler-Nichols method if

$$\tau < 0.38$$

We can thus conclude that the Ziegler-Nichols tuning rules for PI controllers can be applied only for small values of τ. The upper bound is approximately $\tau = 0.3$. For larger normalized dead times the integral action is too weak.

The Chien, Hrones and Reswick Method

There has been many suggestions of modifications of the Ziegler-Nichols methods. Chien, Hrones and Reswick (CHR) changed the step response method to give better damped closed-loop systems. They proposed to use "quickest response without overshoot" or "quickest response with 20% overshoot" as design criteria. They also made the important observation that tuning for setpoint response or load disturbance response are different.

To tune the controller according to the CHR method, the parameters a and L of the process model are first determined in the same way as for the Ziegler-Nichols step response method. The controller parameters for the load disturbance response method are then given as functions of these two parameters. They are summarized in Table 4.3.

The tuning rules based on the 20% overshoot design criteria in Table 4.3 are quite similar to the Ziegler-Nichols step response method presented in Table 4.1. However, when the 0% overshoot design criteria is used, the gain and the derivative time are smaller and the

Table 4.3 Controller parameters obtained from the Chien, Hrones and Reswick load disturbance response method.

Overshoot		0%			20%	
Controller	K	T_i	T_d	K	T_i	T_d
P	$0.3/a$			$0.7/a$		
PI	$0.6/a$	$4L$		$0.7/a$	$2.3L$	
PID	$0.95/a$	$2.4L$	$0.42L$	$1.2/a$	$2L$	$0.42L$

integral time is larger. This means that the proportional action, the integral action, as well as the derivative action, are smaller.

In the setpoint response method, the controller parameters are not only based on a and L, but also on the time constant T. Methods to obtain these parameters were presented in Section 2.4. The tuning rules for setpoint response are summarized in Table 4.4.

Discussion

The Ziegler-Nichols tuning rules were developed empirically based on simulation of a large number of cases. The cases considered were typically such that process dynamics is the main factor that limits performance. When developing the rules it was also attempted to choose numerical values that give simple rules. The methods are simple and easy to use. The process is characterized by two parameters that can be determined by simple experiments. The frequency response method has the advantage that parameters K_u and T_u are easier to determine accurately than the parameters a and L, which are used by the step response method.

The main design criterion was to obtain good rejection of load disturbances specified as quarter amplitude decay ratio. Little emphasis was given to measurement noise, sensitivity to process variations, and setpoint response. The quarter amplitude decay ratio gives systems with very poor damping. The step response method often gives higher loop gains than the frequency domain method. Both methods give better parameters for PID control than for PI control, but in spite of their widely spread use they give poor tuning.

Table 4.4 Controller parameters obtained from the Chien, Hrones and Reswick setpoint response method.

Overshoot		0%			20%	
Controller	K	T_i	T_d	K	T_i	T_d
P	$0.3/a$			$0.7/a$		
PI	$0.35/a$	$1.2T$		$0.6/a$	T	
PID	$0.6/a$	T	$0.5L$	$0.95/a$	$1.4T$	$0.47L$

4.4 Loop Shaping

The Ziegler Nichols frequency response method tries to position one point on the loop transfer function appropriately. Even though the point is chosen cleverly it is surprising that such a method works so well. Some consequences of positioning one point only were discussed in Section 4.3. There are many other design methods that try to obtain a loop transfer function with a good shape. Some of these methods are discussed in this section.

Slope Adjustments

Only two parameters are needed to change the value of the loop transfer function at one frequency. This is the reason why the Ziegler-Nichols method gives unique parameters for a PI controller. For a PID controller, which has three parameters, the condition $T_i = 4T_d$ was introduced in order to obtain unique parameter values. Thus, for PID controllers we have one degree of freedom that can be used to shape the loop transfer function. One possibility is to position one point and to adjust the slope of the Nyquist curve at the chosen point. A natural requirement is that the slope at the chosen frequency ω_0 should be orthogonal to the line $1 + G_\ell(i\omega_0)$ (see Figure 4.15). This ensures that the sensitivity is minimized locally. To see how this can be accomplished, consider a system with Ziegler-Nichols tuning, i.e.,

$$G_p(i\omega_u) = -\frac{1}{K_u}$$

where the controller

$$G_c(i\omega) = K\left(1 + i\big(\omega T_d - \frac{1}{\omega T_i}\big)\right)$$

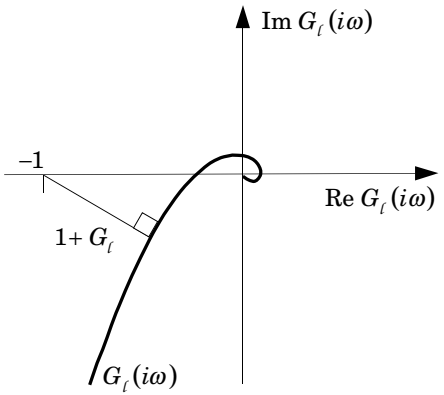

Figure 4.15 Adjustment of the slope of the Nyquist curve.

is chosen so that the loop transfer function has the value $r_b e^{i(\pi+\phi_b)}$ at ω_u as was discussed in Section 4.3. This means that

$$\begin{aligned} K &= K_u r_b \cos\phi_b \\ T_d - \frac{1}{\omega_u^2 T_i} &= \frac{1}{\omega_u}\tan\phi_b = a \end{aligned} \tag{4.30}$$

Notice that we still have freedom to choose the ratio T_i/T_d. To make this choice so that the Nyquist curve has a given slope at ω_u consider the loop transfer function $G_\ell(i\omega) = G_p(i\omega)G_c(i\omega)$. Differentiating the loop transfer function with respect to ω gives.

$$\frac{dG_\ell(i\omega)}{d\omega} = G_p(i\omega)\frac{dG_c(i\omega)}{d\omega} + \frac{dG_p(i\omega)}{d\omega}G_c(i\omega)$$

Furthermore, we have

$$\frac{dG_c(i\omega)}{d\omega} = iK\left(T_d + \frac{1}{\omega^2 T_i}\right) = iK(2T_d - a)$$

where the last equality follows from Equation (4.30). If transfer function G_p is parameterized as

$$G_p(i\omega) = r(\omega)e^{i(\phi(\omega)-\pi)}$$

straightforward but tedious calculations give

$$G'_\ell(i\omega_u) = -\frac{K}{K_u}\left(\frac{r'}{r} - a\phi'\omega_u + i\left(\phi' + a\omega_u\frac{r'}{r} + 2T_d - a\right)\right)$$

where $'$ denotes derivative with respect to ω. Hence

$$\arg\frac{dG_\ell(i\omega)}{dt} = \arctan\frac{\phi' + a\omega_u r'/r + 2T_d - a}{r'/r - a\omega_u\phi'}$$

This gives the following equation for T_d.

$$T_d = \frac{1}{2}\left(a - \phi' - a\omega_u \frac{r'}{r} + \left(\frac{r'}{r} - a\omega_u \phi'\right)\tan\psi\right) \tag{4.31}$$

where ψ is the desired slope of the Nyquist curve at ω_u. Parameter T_i is then given by

$$T_i = \frac{1}{\omega_u^2(T_d - a)} \tag{4.32}$$

We illustrate the procedure with an example

EXAMPLE 4.5

Consider the same process model as in Examples 4.1 and 4.2, i.e.

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

This process has ultimate gain $K_u = 8$ and ultimate frequency $\omega_u = \sqrt{3}$. The amplitude r and its derivative, and the phase ϕ and its derivative are

$$r = \frac{1}{(1+\omega^2)^{3/2}} \qquad r' = -\frac{3\omega}{(1+\omega^2)^{5/2}}$$

$$\phi = \pi - 3\arctan\omega \qquad \phi' = -\frac{3}{(1+\omega^2)}$$

Their values at ω_u become

$$r(\omega_u) = \frac{1}{8} \qquad r'(\omega_u) = -\frac{3\sqrt{3}}{32} \qquad \phi(\omega_u) = 0 \qquad \phi'(\omega_u) = -\frac{3}{4}$$

Suppose that we want to move the ultimate point to the new position

$$r_b = \frac{1}{\sqrt{2}} \qquad \phi_b = 45^\circ$$

The slope of Nyquist curve is orthogonal to the line $1 + G_\ell(i\omega_u)$ if it is chosen to $\psi = 45^\circ$. This choice gives an M_s value equal to $M_s = \sqrt{2} \approx 1.4$.

The controller parameters can now be obtained from Equations (4.30), (4.31) and (4.32). They become $K = 4$, $T_i = 1.9$, and $T_d = 0.75$. Figure 4.16 shows the response of the closed-loop systems to a step change in setpoint followed by a step change in the load. □

Frequency Domain Design of a PID Controller

The Ziegler-Nichols method was based on knowledge of the process transfer function at the ultimate point. The design method we have

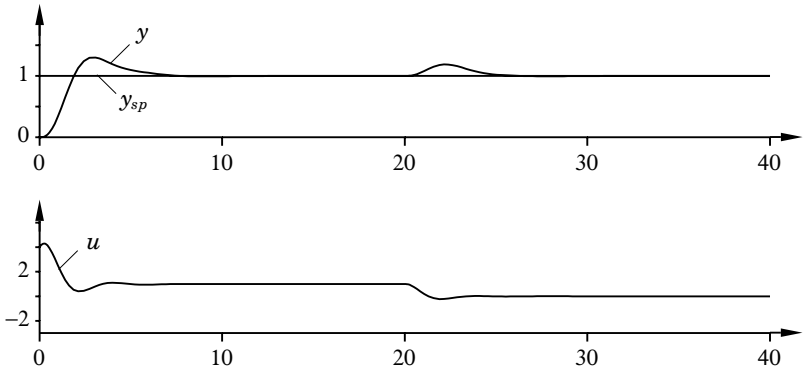

Figure 4.16 Setpoint and load disturbance response of a process with transfer function $1/(s+1)^3$ controlled by a PID controller tuned with the loop-shaping method. The diagrams show setpoint y_{sp}, process output y, and control signal u.

just discussed requires, in addition, knowledge of the slope of the plant transfer function at the ultimate point. More effective methods can be used if the whole plant transfer function is known. Such a method will now be discussed.

The design criterion is to obtain a specified sensitivity M_s with good rejection of load disturbances. Let $r(\omega)$ and $\phi(\omega)$ denote magnitude and phase of the frequency response of the process i.e.

$$G_p(i\omega) = r(\omega)e^{i(\phi(\omega)-\pi)}$$

and let the controller transfer function be

$$G_c(s) = k + \frac{k_i}{s} + k_d s$$

To have a given sensitivity M_s, the Nyquist curve of the loop transfer function must avoid a circle around the critical point with radius $r_0 = 1/M_s$ (see Figure 4.17). Assume that the curve meets the circle tangentially at the point A. The condition that the loop transfer function goes through A is

$$r(\omega)e^{i(\phi(\omega)-\pi)}\left(k + i\left(k_d\omega - \frac{k_i}{\omega}\right)\right) = -1 + r_0\cos\theta - ir_0\sin\theta$$

where r_0 and θ are defined in Figure 4.17. Separating real and imaginary parts we get

$$kr(\omega)\cos\phi(\omega) + k_i\frac{r(\omega)}{\omega}\sin\phi(\omega) - k_d\omega r(\omega)\sin\phi(\omega) = -1 + r_0\cos\theta$$

$$kr(\omega)\sin\phi(\omega) - k_i\frac{r(\omega)}{\omega}\cos\phi(\omega) + k_d\omega r(\omega)\cos\phi(\omega) = -r_0\sin\theta$$

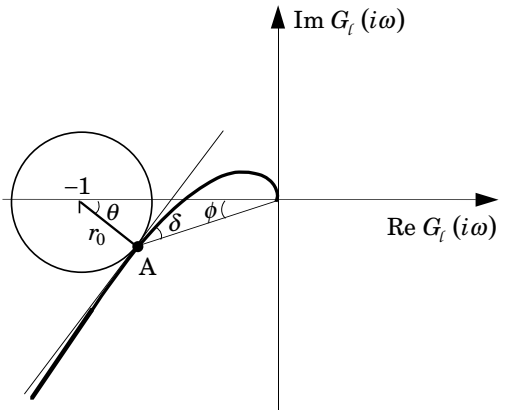

Figure 4.17 Geometrical illustration of the loop-shaping design method.

Calculating the derivative of the loop transfer function we find, after simplifications,

$$\frac{dG_\ell(i\omega)}{d\omega} = re^{i(\phi-\pi)}\left(k\frac{r'}{r} - \left(\omega k_d - \frac{k_i}{\omega}\right)\phi' + i\left(k\phi' + \left(\omega k_d - \frac{k_i}{\omega}\right)\frac{r'}{r} + k_d + \frac{k_i}{\omega^2}\right)\right)$$

The condition for tangency can be written as

$$\arg\frac{dG_\ell(i\omega)}{d\omega} = \frac{\pi}{2} - \theta$$

This can be simplified to

$$ak + bk_i + ck_d = 0$$

where

$$\begin{aligned} a &= \phi'(\omega) - \frac{r'(\omega)}{r(\omega)}\tan\delta \\ b &= \frac{1}{\omega^2} - \frac{r'(\omega)}{\omega r(\omega)} - \frac{\phi'(\omega)\tan\delta}{\omega} \\ c &= 1 + \frac{\omega r'(\omega)}{r(\omega)} + \omega\phi'(\omega)\tan\delta \end{aligned} \tag{4.33}$$

and the angle δ is defined in Figure 4.17. We thus obtain three equations to determine the controller parameters k, k_i, and k_d. Notice, however, that both ω and θ can be considered as unknowns. To determine these parameters, we can introduce the condition that k_i should be as large as possible, i.e., to minimize IE. Another possibility is

to make ω as large as possible. The design of the controller then is reduced to an optimization problem. Notice that parameter θ varies in the range $(0, \pi/2)$. The value ω_u can be used as an initial value for the optimization. Instead of minimizing IE we could also consider minimization of IAE. However, this will increase the computational effort considerably. Notice that when performing the optimization we also obtain the argument ω_0 for which the optimum is achieved. This indicates the frequency range where model precision is needed.

4.5 Analytical Tuning Methods

There are several analytical tuning methods where the controller transfer function is obtained from the specifications by a direct calculation. Let G_p and G_c be the transfer functions of the process and the controller. The closed-loop transfer function obtained with error feedback is then

$$G_0 = \frac{G_p G_c}{1 + G_p G_c}$$

Solving this equation for G_c we get

$$G_c = \frac{1}{G_p} \cdot \frac{G_0}{1 - G_0} \tag{4.34}$$

If the closed-loop transfer function G_0 is specified and G_p is known, it is thus easy to compute G_c. The key problem is to find reasonable ways to determine G_0 based on engineering specifications of the system.

It follows from Equation (4.34) that all process poles and zeros are canceled by the controller. This means that the method cannot be applied when the process has poorly damped poles and zeros. The method will also give a poor load disturbance response when slow process poles are canceled.

λ-Tuning

The method called λ-tuning was developed for processes with long dead time L. Consider a process with the transfer function

$$G_p = \frac{K_p}{1 + sT} e^{-sL}$$

Assume that the desired closed-loop transfer function is specified as

$$G_0 = \frac{e^{-sL}}{1 + s\lambda T}$$

where λ is a tuning parameter. The time constants of the open- and closed-loop systems are the same when $\lambda = 1$. The closed-loop system responds faster than the open-loop system if $\lambda < 1$. It is slower when $\lambda > 1$.

It follows from Equation (4.34) that the controller transfer function becomes

$$G_c = \frac{1 + sT}{K_p(1 + \lambda sT - e^{-sL})} \tag{4.35}$$

The controller has integral action, because $G_c(0) = \infty$. The input-output relation of the controller is

$$\left(1 + s\lambda T - e^{-sL}\right) U(s) = \frac{1}{K_p}(1 + sT)E(s) \tag{4.36}$$

This can be written as

$$U(s) = \frac{1}{\lambda K_p}\left(1 + \frac{1}{sT}\right)\left(E(s) - \frac{K_p}{1 + sT}\left(1 - e^{-sL}\right) U(s)\right) \tag{4.37}$$

When $L = 0$, this becomes a PI controller with gain $K = 1/(\lambda K_p)$ and integral time $T_i = T$. The term

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

can be interpreted as a prediction of the process output at time t based on the values of the control signal in the time interval $(t-T, t)$. The controller given by Equation (4.37) can thus be interpreted as a predictive PI controller where the prediction is formed by correcting for the effects of the control actions that have been taken, but have not yet appeared in the output because of the delay in the process. The controller is, therefore, called a predictive PI controller (PPI). For processes with long dead times, the prediction given in Equations (4.36) and (4.37) is much better than the prediction obtained by derivative action.

The PPI controller can be written as

$$U(s) = \frac{1}{\lambda K_p}\left(1 + \frac{1}{sT}\right) E(s) - \frac{1}{s\lambda T}\left(1 - e^{-sL}\right) U(s) \tag{4.38}$$

In Section 4.2, it was shown that the integral error for a PID controller is

$$IE_{\text{PID}} = \frac{T_i}{K}$$

The integrated errors obtained with a PID controller and a controller with λ-tuning are compared. With a PID controller based on the Ziegler-Nichols step response method, we obtain

$$IE_{\text{PID}} = \frac{K_p L^2}{0.6T}$$

To compute the integral error for the PPI controller it will be assumed that the system is initially at rest and that a load disturbance in the form of a unit step is applied to the process input. Since the controller has integral action, we have $u(\infty) = 1$. To integrate Equation (4.38), first note that

$$\frac{1}{\lambda T}\int_0^\infty \big(u(t) - u(t-L)\big)\,dt = \frac{L}{\lambda T}$$

Integration of Equation (4.38) from 0 to ∞ now gives

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

The integral error thus becomes

$$IE_{\text{PPI}} = K_p(L + \lambda T)$$

The integrated error is smaller with PPI control than with PID control when L is large. For $\lambda = 1$ the criteria are equal when $L/T = 1.1$. The improvements with PPI control increases with decreasing values of λ.

The sensitivity function obtained with λ-tuning is given by

$$S(s) = 1 - \frac{e^{-sL}}{1 + s\lambda T} = \frac{1 + s\lambda T - e^{-sL}}{1 + s\lambda T}$$

It can be shown that

$$M_s = \max_{\omega} |S(i\omega)|$$

is always less than 2. An approximate expression for M_s is given by

$$M_s = 2 - \lambda\frac{T}{L}$$

Thus, to have a value of M_s smaller than 2 it is important that λ is sufficiently large.

We note that in order to make the integrated error small it is advantageous to have a small value of λ. A small value of λ, however, will increase the sensitivity.

In practice it is common to choose λ between 0.5 and 5. The PPI controller is particularly simple if $\lambda = 1$, i.e., if the desired closed-loop time constant is equal to the open-loop time constant. Equation (4.38) then becomes

$$U(s) = K\left(1 + \frac{1}{sT_i}\right)E(s) - \frac{1}{sT_i}\left(1 - e^{-sL}\right)U(s)$$

where $K = 1/K_p$ and $T_i = T$. This equation can also be written as

$$U(s) = KE(s) + \frac{e^{-sL}}{1 + sT_i}\,U(s) \tag{4.39}$$

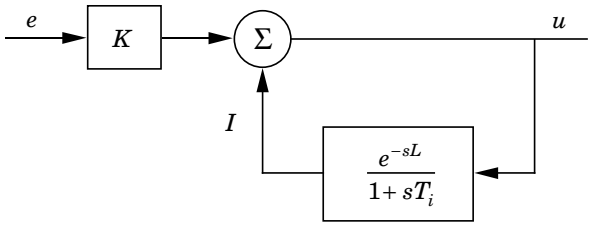

Figure 4.18 Block diagram of the PPI controller with $\lambda = 1$.

A block diagram describing this equation is given in Figure 4.18. Notice the strong similarity with the PI controller shown in Figure 3.8.

The Haalman Method

Another approach is to determine an ideal loop transfer function G_ℓ that gives the desired performance and to choose the controller transfer function as

$$G_c = \frac{G_\ell}{G_p} \tag{4.40}$$

where G_p is the process transfer function. Such an approach can give PI and PID controllers provided that G_ℓ and G_p are sufficiently simple. There are many ways to obtain a suitable G_ℓ.

For systems with a time delay L, Haalman has suggested choosing

$$G_\ell(s) = \frac{2}{3Ls}\, e^{-sL} \tag{4.41}$$

The value $2/3$ was found by minimizing the mean square error for a step change in the setpoint. This choice gives a sensitivity $M_s = 1.9$, which is a reasonable value. Notice that it is only the dead time of the process that influences the loop transfer function. All other process poles and zeros are canceled, which may lead to difficulties.

Applying Haalman's method to a processes with the transfer function

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

gives the controller

$$G_c(s) = \frac{2(1+sT)}{3Ls} = \frac{2T}{3L}\left(1 + \frac{1}{sT}\right)$$

which is a PI controller with $K = 2T/3L$ and $T_i = T$. These parameters can be compared with the values $K = 0.9T/L$ and $T_i = 3L$ obtained by the Ziegler-Nichols step response method. A PID controller

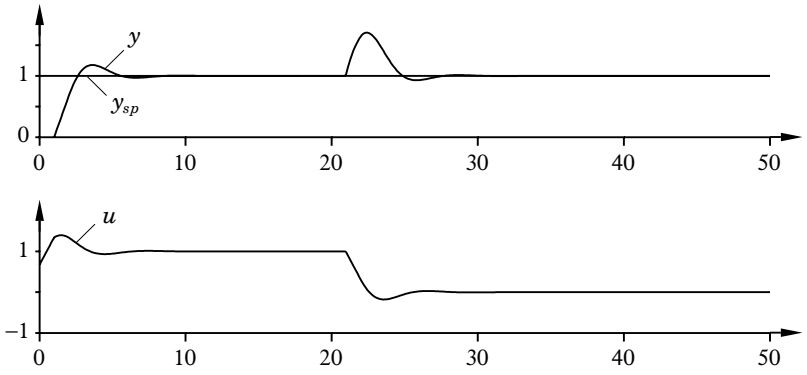

Figure 4.19 Simulation of a closed-loop system obtained by Haalman's method. The plant transfer function is $G(s) = e^{-s}/(s+1)$. The diagrams show setpoint y_{sp}, process output y, and control signal u.

is obtained if the method is applied to a process with the transfer function

$$G_p(s) = \frac{1}{(1+sT_1)(1+sT_2)}\, e^{-sL}$$

The parameters of the controller are $K = 2(T_1+T_2)/3L$, $T_i = T_1+T_2$, and $T_d = T_1T_2/(T_1+T_2)$.

For more complex processes it is necessary to approximate the processes to obtain a transfer function of the desired form as was discussed in Section 2.9. Figure 4.19 shows a simulation of Haalman's method for a system whose dynamics is dominated by dead time. The normalized dead time is 0.5 for this system. The figure shows that the responses are excellent.

Drawbacks of Pole-Zero Cancellations

A key feature of Haalman's method is that process poles and zeros are canceled by poles and zeros in the controller. When poles and zeros are canceled, there will be uncontrollable modes in the closed-loop system. This may lead to poor performance if the modes are excited. The problem is particular severe if the canceled modes are slow or unstable. We use an example to illustrate what may happen.

EXAMPLE 4.6 Loss of controllability due to cancellation

Consider a closed-loop system where a process with the transfer function

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

is controlled with a PI controller whose parameters are chosen so that the process pole is canceled. The transfer function of the controller is then

$$G_c(s) = K\left(1 + \frac{1}{sT}\right) = K\,\frac{1+sT}{sT}$$

The process can be represented by the equation

$$\frac{dy(t)}{dt} = \frac{1}{T}\,(u(t-L) - y(t)) \tag{4.42}$$

and the controller can be described by

$$\frac{du(t)}{dt} = -K\left(\frac{dy(t)}{dt} + \frac{y(t)}{T}\right) \tag{4.43}$$

Consider the behaviour of the closed-loop system when the initial conditions are chosen as $y(0) = 1$ and $u(t) = 0$ for $-L < t < 0$. Without feedback the output is given by

$$y_{ol}(t) = e^{-t/T}$$

To compute the output for the closed-loop system we first eliminate $y(t)$ between Equations (4.42) and (4.43). This gives

$$\frac{du(t)}{dt} = -\frac{K}{T}\,u(t-L)$$

It thus follows that $u(t) = 0$, and Equation (4.42) then implies that

$$y_{cl}(t) = e^{-t/T} = y_{ol}(t)$$

The trajectories of the closed-loop system and the open-loop system thus are the same. The control signal is zero, which means that the controller does not attempt to reduce the control error. □

The example clearly indicates that there are drawbacks with cancellation of process poles. Another illustration of the phenomenon is given in Figure 4.20, which is a simulation of a closed-loop system where the controller is designed by Haalman's method. This simulation is identical to the simulation in Figure 4.19 but the process time constant is now 10 instead of 1 for the simulation in Figure 4.19. In this case we find that the setpoint response is excellent but that the response to load disturbances is very poor. The reason for this is that the controller cancels the pole $s = -0.1$, by having a controller zero at $s = -0.1$. Notice that the process output after a load disturbance decays with the time constant $T = 10$, but that the control signal is practically constant due to the cancellation. The attenuation of load disturbances is improved considerably by reducing the integral time of the controller as shown in Figure 4.20.

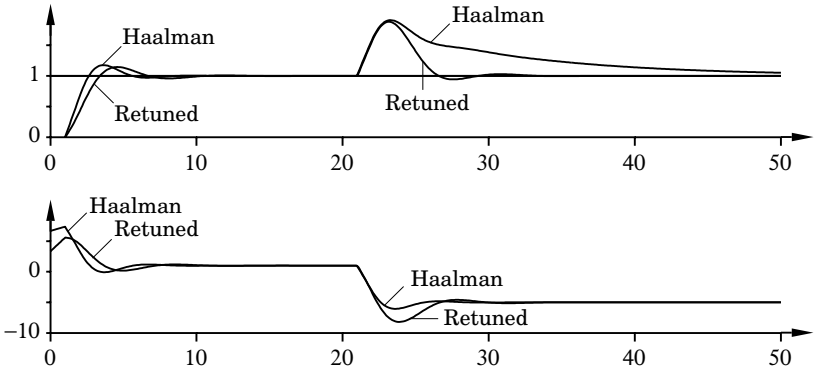

Figure 4.20 Simulation of a closed-loop system obtained by Haalman's method. The process transfer function is $G(s) = e^{-s}/(10s+1)$, and the controller parameters are $K = 6.67$ and $T_i = 10$. The upper diagram shows setpoint $y_{sp} = 1$ and process output y, and the lower diagram shows control signal u. The figure also shows the responses to a retuned controller with $K = 6.67$, $T_i = 3$ and $b = 0.5$.

We have thus shown that cancellation of process poles may give systems with poor rejection of load disturbances. Notice that this does not show up in simulations unless the process is excited. For example, it will not be noticed in a simulation of a step change in the setpoint. We may also ask why there is such a big difference in the simulation in Figure 4.18 and Figure 4.20. The reason is that the canceled pole in Figure 4.20 is slow in comparison with the closed-loop poles, but it is of the same magnitude as the closed-loop poles in Figure 4.18.

We can thus conclude that pole cancellation can be done for systems that are dead time dominated but not for systems that are lag dominated.

The Internal Model Controller (IMC)

The internal model principle is a general method for design of control systems that can be applied to PID control. A block diagram of such a system is shown in Figure 4.21. In the diagram it is assumed that all disturbances acting on the process are reduced to an equivalent disturbance d at the process output. In the figure G_m denotes a model of the process, $G_m^\dagger$ is an approximate inverse of G_m, and G_f is a low-pass filter. The name internal model controller derives from the fact that the controller contains a model of the process internally. This model is connected in parallel with the process.

If the model matches the process, i.e., $G_m = G_p$, the signal e is equal to the disturbance d for all control signals u. If $G_f = 1$ and $G_m^\dagger$ is

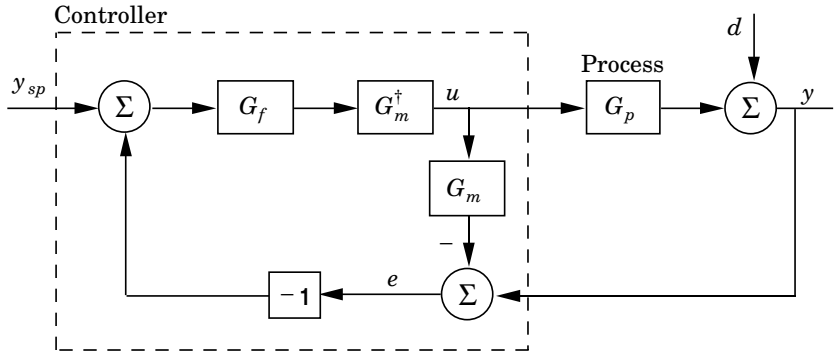

Figure 4.21 Block diagram of a closed-loop system with a controller based on the internal model principle.

an exact inverse of the process, then the disturbance d will be canceled perfectly. The filter G_f is introduced to obtain a system that is less sensitive to modeling errors. A common choice is $G_f(s) = 1/(1+sT_f)$, where T_f is a design parameter.

The controller obtained by the internal model principle can be represented as an ordinary series controller with the transfer function

$$G_c = \frac{G_f G_m^\dagger}{1 - G_f G_m^\dagger G_m} \tag{4.44}$$

From this expression it follows that controllers of this type cancel process poles and zeros.

The internal model principle will typically give controllers of high order. By making special assumptions it is, however, possible to obtain PI or PID controllers from the principle. To see this consider a process with the transfer function

$$G_p(s) = \frac{K_p}{1+sT} e^{-sL} \tag{4.45}$$

An approximate inverse is given by

$$G_m^\dagger(s) = \frac{1+sT}{K_p}$$

Notice that it is not attempted to find an inverse of the time delay. Choose the filter

$$G_f(s) = \frac{1}{1+sT_f}$$

Approximating the time delay by

$$e^{-sL} \approx 1 - sL$$

Equation (4.44) now gives

$$G_c(s) = \frac{1 + sT}{K_p s(L + T_f)}$$

which is a PI controller. If the time delay is approximated instead by a first-order Padé approximation

$$e^{-sL} \approx \frac{1 - sL/2}{1 + sL/2}$$

Equation (4.44) gives instead the PID controller

$$G_c(s) = \frac{(1 + sL/2)(1 + sT)}{K_p s(L + T_f + sT_f L/2)} \approx \frac{(1 + sL/2)(1 + sT)}{K_p s(L + T_f)}$$

For processes described by Equation (4.45), we thus find that the internal model principle will give PI or PID controllers. Approximations like the ones discussed in Section 2.9 can be used in the usual manner to obtain PI and PID controllers for more complex processes.

An interesting feature of the internal model controller is that robustness is considered explicitly in the design. Robustness can be adjusted by selecting the filter G_f properly. A trade-off between performance and robustness can be made by using the filter constant as a design parameter. The IMC can be designed to give excellent response to setpoint changes. Since the design method inherently implies that poles and zeros of the plant are canceled, the response to load disturbances may be poor if the canceled poles are slow in comparison with the dominant poles. Compare with the responses in Figure 4.20. The IMC controller can also be viewed as an extension of the Smith predictor.

4.6 Optimization Methods

Optimization is a powerful tool for design of controllers. The method is conceptually simple. A controller structure with a few parameters is specified. Specifications are expressed as inequalities of functions of the parameters. The specification that is most important is chosen as the function to optimize. The method is well suited for PID controllers where the controller structure and the parameterization are given. There are several pitfalls when using optimization. Care must be exercised when formulating criteria and constraints; otherwise, a criterion will indeed be optimal, but the controller may still be unsuitable because of a neglected constraint. Another difficulty is that the loss function may have many local minima. A third is that the computations required may easily be excessive. Numerical problems may also arise. Nevertheless, optimization is a good tool that

Table 4.5 Controller parameters obtained from minimization of integrated absolute error, IAE.

L	IAE	M_s	K_{hf}	K	T_i
0.0	0		∞	∞	0
0.2	0.14	3.3	4.7	4.7	0.62
0.5	0.60	3.0	2.0	2.0	1.1
1.0	1.5	2.4	1.0	1.0	1.4
2.0	3.2	2.1	0.60	0.60	1.8
5.0	7.7	2.0	0.42	0.42	3.1
10.0	15	1.9	0.37	0.37	5.3

has successfully been used to design PID controllers. In this section we discuss some of these methods.

EXAMPLE 4.7 A PI controller optimized for IAE

Consider a process with the transfer function

$$G_p(s) = \frac{1}{s+1}\, e^{-sL} \tag{4.46}$$

Table 4.5 gives controller parameters obtained when minimizing IAE for load disturbances. Some of the other criteria are also given in the table. Notice that the integrated absolute error increases with L, as can be expected. Notice also that, although the criterion IAE is minimized, several other design criteria such as the M_s value and the high-frequency controller gain K_{hf} have undesirable values. Notice in particular that the values of M_s are quite high. The example illustrates the necessity of considering many performance criteria when using optimization methods. Unfortunately, this was not observed in much of the early work on controller tuning. □

Tuning Formulas Based on Optimization

Many studies have been devoted to development of tuning rules based on optimization. Very often a process described by

$$G_p = \frac{K_p}{1+sT}\, e^{-sL}$$

has been considered. The loss functions obtained for unit step changes in setpoint and process input have been computed and formulas of the

type

$$p = a\left(\frac{L}{T}\right)^b$$

where p is a controller parameter and a and b are constants, have been fitted to the numerical values obtained. In many cases the criterion is IAE for load disturbances, which often gives systems with low damping and poor sensitivity. The formulas given often only hold for a small range of normalized dead times, e.g., $0.2 < \tau < 0.6$. It should also be observed that criteria based on setpoint changes can often be misleading because it is often not observed that the setpoint changes are drastically influenced by different setpoint weightings.

Modulus and Symmetrical Optimum

Modulus Optimum (BO) and Symmetrical Optimum (SO) are two methods for selecting and tuning controllers that are similar in spirit to Haalman's method. The acronyms BO and SO are derived from the German words Betrags Optimum and Symmetrische Optimum. These methods are based on the idea of finding a controller that makes the frequency response from setpoint to plant output as close to one as possible for low frequencies. If $G(s)$ is the transfer function from the setpoint to the output, the controller is determined in such a way that $G(0) = 1$ and that $d^n|G(i\omega)|/d\omega^n = 0$ at $\omega = 0$ for as many n as possible. We illustrate the idea with a few examples.

EXAMPLE 4.8 Second-order system

Consider the transfer function

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

which has been chosen so that $G(0) = 1$. Let us first consider how the parameters should be chosen in order to get a maximally flat frequency response. We have

$$|G(i\omega)|^2 = \frac{a_2^2}{a_1^2\omega^2 + (a_2 - \omega^2)^2} = \frac{a_2^2}{a_2^2 + \omega^2(a_1^2 - 2a_2) + \omega^4}$$

By choosing $a_1 = \sqrt{2a_2}$ we find

$$|G(i\omega)|^2 = \frac{a_2^2}{a_2^2 + \omega^4}$$

The first three derivatives of $|G(i\omega)|$ will vanish at the origin. The transfer function then has the form

$$G(s) = \frac{\omega_0^2}{s^2 + \sqrt{2}\omega_0 s + \omega_0^2}$$

The step response of a system with this transfer function has an overshoot $o = 4\%$. The settling time to 2% of the steady state value is $t_s = 6/\omega_0$. □

If the transfer function G in the example is obtained by error feedback of a system with the loop transfer function G_{BO}, the loop transfer function is

$$G_{\mathrm{BO}}(s) = \frac{G(s)}{1 - G(s)} = \frac{\omega_0^2}{s(s + \sqrt{2}\omega_0)} \tag{4.47}$$

which is the desired loop transfer function for the method called modulus optimum.

The calculation in Example 4.8 can be performed for higher-order systems with more effort. We illustrate by another example.

EXAMPLE 4.9 Third-order system

Consider the transfer function

$$G(s) = \frac{a_3}{s^3 + a_1 s^2 + a_2 s + a_3}$$

After some calculations we get

$$|G(i\omega)| = \frac{a_3}{\sqrt{a_3^2 + (a_2^2 - 2a_1 a_3)\omega^2 + (a_1^2 - 2a_2)\omega^4 + \omega^6}}$$

Five derivatives of $|G(i\omega)|$ will vanish at $\omega = 0$, if the parameters are such that $a_1^2 = 2a_2$ and $a_2^2 = 2a_1 a_3$. The transfer function then becomes

$$G(s) = \frac{\omega_0^3}{s^3 + 2\omega_0 s^2 + 2\omega_0^2 s + \omega_0^3} = \frac{\omega_0^3}{(s + \omega_0)(s^2 + \omega_0 s + \omega_0^2)} \tag{4.48}$$

The step response of a system with this transfer function has an overshoot $o = 8.1\%$. The settling time to 2% of the steady state value is $9.4/\omega_0$. A system with this closed-loop transfer function can be obtained with a system having error feedback and the loop transfer function

$$G_\ell(s) = \frac{\omega_0^3}{s(s^2 + 2\omega_0 s + 2\omega_0^2)}$$

The closed-loop transfer function (4.48) can also be obtained from other loop transfer functions if a two-degree of freedom controller is used. For example, if a process with the transfer function

$$G_p(s) = \frac{\omega_0^2}{s(s + 2\omega_0)}$$

is controlled by a PI controller having parameters $K = 2$, $T_i = 2/\omega_0$ and $b = 0$, the loop transfer function becomes

$$G_{SO} = \frac{\omega_0^2(2s + \omega_0)}{s^2(s + 2\omega_0)} \tag{4.49}$$

The symmetric optimum aims at obtaining the loop transfer function given by Equation (4.49). Notice that the Bode diagram of this transfer function is symmetrical around the frequency $\omega = \omega_0$. This is the motivation for the name symmetrical optimum.

If a PI controller with $b = 1$ is used, the transfer function from setpoint to process output becomes

$$G(s) = \frac{G_{SO}(s)}{1 + G_{SO}(s)} = \frac{(2s + \omega_0)\omega_0^2}{(s + \omega_0)(s^2 + \omega_0 s + \omega_0^2)}$$

This transfer function is not maximally flat because of the zero in the numerator. This zero will also give a setpoint response with a large overshoot, about 43%. □

The methods BO and SO can be called loop-shaping methods since both methods try to obtain a specific loop transfer function. The design methods can be described as follows. It is first established which of the transfer functions, G_{BO} or G_{SO}, is most appropriate. The transfer function of the controller $G_c(s)$ is then chosen so that $G_\ell(s) = G_c(s)G_p(s)$, where G_ℓ is the chosen loop transfer function. We illustrate the methods with the following examples.

EXAMPLE 4.10 BO control

Consider a process with the transfer function

$$G_p(s) = \frac{K_p}{s(1 + sT)} \tag{4.50}$$

With a proportional controller the loop transfer function becomes

$$G_\ell(s) = \frac{KK_p}{s(1 + sT)}$$

To make this transfer function equal to G_{BO} given by Equation (4.47) it must be required that

$$\omega_0 = \frac{\sqrt{2}}{2T}$$

The controller gain should be chosen as

$$K = \frac{\omega_0\sqrt{2}}{2K_p} = \frac{1}{2K_pT}$$

□

EXAMPLE 4.11 SO control

Consider a process with the same transfer function as in the previous example (Equation (4.50)). With a PI controller having the transfer function

$$G_c(s) = \frac{K(1 + sT_i)}{sT_i}$$

we obtain the loop transfer function

$$G_\ell(s) = \frac{K_p K(1 + sT_i)}{s^2 T_i(1 + sT)}$$

This is identical to G_{SO} if we choose

$$K = \frac{1}{2K_p T}$$
$$T_i = 4T$$

To obtain the transfer function given by Equation (4.48) between setpoint and process output, the controller should have the input-output relation

$$u(t) = K\left(-y(t) + \frac{1}{T_i}\int^t (y_{sp}(s) - y(s))\, ds\right)$$

The coefficient b in the standard controller thus should be set to zero. □

A Design Procedure

A systematic design procedure can be based on the methods BO and SO. The design method consists of two steps. In the first step the process transfer function is simplified to one of the following forms

$$G_1(s) = \frac{K_p}{1 + sT} \tag{4.51}$$

$$G_2(s) = \frac{K_p}{(1 + sT_1)(1 + sT_2)}, \quad T_1 > T_2 \tag{4.52}$$

$$G_3(s) = \frac{K_p}{(1 + sT_1)(1 + sT_2)(1 + sT_3)}, \quad T_1 > T_2 > T_3 \tag{4.53}$$

$$G_4(s) = \frac{K_p}{s(1 + sT)} \tag{4.54}$$

$$G_5(s) = \frac{K_p}{s(1 + sT_1)(1 + sT_2)}, \quad T_1 > T_2 \tag{4.55}$$

Process poles may be canceled by controller zeros to obtain the desired loop transfer function. A slow pole may be approximated by an integrator; fast poles may be lumped together as discussed in Section 2.9.

The rule of thumb given in the original papers on the method is that time constants such that $\omega_0 T < 0.25$ can be regarded as integrators.

The controller is derived in the same way as in Examples 4.10 and 4.11 by choosing parameters so that the loop transfer function matches either G_{BO} or G_{SO}. By doing this we obtain the results summarized in Table 4.6. Notice, for example, that Example 4.10 and 4.11 correspond to the entries Process G_4 in the table. It is natural to view the smallest time constant as an approximation of neglected dynamics in the process. It is interesting to observe that it is this time constant that determines the bandwidth of the closed-loop system.

The setpoint response for the BO method is excellent. Notice that it is necessary to use a controller with a two-degree-of-freedom structure or a prefilter to avoid a high overshoot for the SO method. Notice that process poles are canceled in the cases marked C1 or C2 in Table 4.6. The response to load disturbances will be poor if the canceled pole is slow compared to the closed-loop dynamics, which is characterized by ω_0 in Table 4.6.

These design principles can be extended to processes other than those listed in the table.

EXAMPLE 4.12 Application of BO and SO

Consider a process with the transfer function

$$G(s) = \frac{1}{(1+s)(1+0.2s)(1+0.05s)(1+0.01s)} \tag{4.56}$$

Since this transfer function is of fourth order, the design procedure cannot be applied directly. We show how different controllers are obtained depending on the approximations made. The performance of the closed-loop system depends on the approximation. We use parameter ω_0 as a crude measure of performance.

If a controller with low performance is acceptable, the process (4.56) can be approximated with

$$G(s) = \frac{1}{1+1.26s} \tag{4.57}$$

The approximation has a phase error less than 10° for $\omega \leq 1.12$. It follows from Table 4.6 that the system (4.57) can be controlled with an integrating controller with

$$k_i = \frac{K}{T_i} = \frac{0.5}{1.26} = 0.4$$

This gives a closed-loop system with $\omega_0 = 0.55$.

A closed-loop system with better performance is obtained if the transfer function (4.56) is approximated with

$$G(s) = \frac{1}{(1+s)(1+0.26s)} \tag{4.58}$$

Table 4.6 Controller parameters obtained with the BO and SO methods. Entry P gives the process transfer function, entry C gives the controller structure, and entry M tells whether the BO or SO method is used. In the entry Remark, A1 means that $1 + sT_1$ is approximated by sT_1 and Ci means that the time constant T_i is canceled.

P	C	M	Remark	KK_p	T_i	T_d	ω_0	b	c
G_1	I	BO		0.5	T		$\frac{0.7}{T}$		
G_2	P	BO	A1	$\frac{T_1}{2T_2}$			$\frac{0.7}{T_2}$	1	
G_2	PI	BO	C1	$\frac{T_1}{2T_2}$	T_1		$\frac{0.7}{T_2}$	1	
G_2	PI	SO	A1	$\frac{T_1}{2T_2}$	$4T_2$		$\frac{0.5}{T_2}$	0	
G_3	PD	BO	A1, C2	$\frac{T_1}{2T_3}$		T_2	$\frac{0.7}{T_3}$	1	1
G_3	PID	BO	C1, C2	$\frac{T_1+T_2}{2T_3}$	T_1+T_2	$\frac{T_1T_2}{T_1+T_2}$	$\frac{0.7}{T_3}$	1	1
G_3	PID	SO	A1, C2	$\frac{T_1(T_2+4T_3)}{8T_3^2}$	T_2+4T_3	$\frac{4T_2T_3}{T_2+4T_3}$	$\frac{0.5}{T_3}$	$\frac{T_2}{T_2+4T_3}$	0
G_4	P	BO		$\frac{1}{2T}$			$\frac{0.7}{T}$	1	
G_4	PI	SO		$\frac{1}{2T}$	$4T$		$\frac{0.5}{T}$	0	
G_5	PD	BO	C1	$\frac{1}{2T_2}$		T_1	$\frac{0.7}{T_2}$	1	1
G_5	PD	SO	A1	$\frac{T_1}{8T_2^2}$		$4T_2$	$\frac{0.5}{T_2}$	1	0
G_5	PID	SO	C1	$\frac{T_1+4T_2}{8T_2^2}$	T_1+4T_2	$\frac{4T_1T_2}{T_1+4T_2}$	$\frac{0.5}{T_2}$	$\frac{T_1}{T_1+4T_2}$	0

The slowest time constant is thus kept and the remaining time constants are approximated by lumping their time constants. The approximation has a phase error less than 10° for $\omega \leq 5.15$. A PI controller can be designed using the BO method. The parameters $K = 1.92$ and $T_i = 1$ are obtained from Table 4.6. The closed-loop system has $\omega_0 = 2.7$.

If the transfer function is approximated as

$$G(s) = \frac{1}{(1+s)(1+0.2s)(1+0.06s)} \tag{4.59}$$

the approximation has a phase error less than 10° for $\omega \leq 26.6$. The

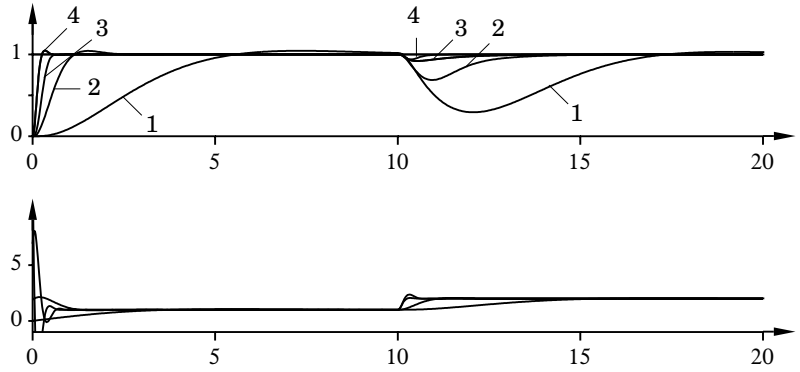

Figure 4.22 Simulation of the closed-loop system obtained with different controllers designed by the BO and SO methods given in Table 4.7. The upper diagram shows setpoint y_{sp} and process output y, and the lower diagram shows control signal u.

BO method can be used also in this case. Table 4.6 gives the controller parameters $K = 10$, $T_i = 1.2$, and $T_d = 0.17$. The controller structure is defined by the parameters $b = 1$ and $c = 1$. This controller gives a closed-loop system with $\omega_0 = 11.7$.

The method SO can also be applied to the system (4.59). Table 4.6 gives the controller parameters $K = 15.3$, $T_i = 0.44$, $T_d = 0.11$, and $b = 0.45$. For these parameters we get $\omega_0 = 8.3$.

Thus, we note that controllers with different properties can be obtained by approximating the transfer function in different ways. A summary of the properties of the closed-loop systems obtained is given in Table 4.7, where IAE refers to the load disturbance response. Notice that Controller 2 cancels a process pole with time constant 1 s, and that Controller 3 cancels process poles with time constants 1 s and 0.25 s. This explains why the IAE drops drastically for Controller 4, which does not cancel any process poles. Controller 4 actually has a lower bandwidth ω_0 than Controller 3.

A simulation of the different controllers is shown in Figure 4.22.

Summary

In this section we discussed using optimization methods to arrive at desirable loop transfer functions. The method by Haalman is designed for systems having dead time. The methods BO and SO apply to systems without dead time. Small dead times can be dealt with by approximation. An interesting feature of both BO and SO is that approximations are used to obtain simple low order transfer functions. There are possibilities to combine the approaches. A drawback with

Table 4.7 Results obtained with different controllers designed by the BO and SO methods in Example 4.12. The frequency ω_m defines the upper limit when the phase error is less than 10%.

Controller	K	T_i	T_d	k_i	b	c	ω_0	ω_m	IAE
1				0.4			0.55	1.12	2.7
2	1.92	1		0.52	1		2.7	5.15	0.52
3	10	1.2	0.17	8.3	1	1	11.7	26.6	0.12
4	15.3	0.44	0.11	35	0.45	0	8.3	26.6	0.029

all design methods of this type is that process poles are canceled. This may lead to poor attenuation of load disturbances if the canceled poles are excited by disturbances and if they are slow compared to the dominant closed-loop poles.

4.7 Pole Placement

This section presents design methods that are based on knowledge of the process transfer function. The pole placement design method simply attempts to find a controller that gives desired closed-loop poles. We illustrate the method by two simple examples.

EXAMPLE 4.13 PI control of a first-order system

Suppose that the process can be described by the following first-order model:

$$G_p(s) = \frac{K_p}{1 + sT} \tag{4.60}$$

which has only two parameters, the process gain (K_p) and the time constant (T). By controlling this process with the PI controller,

$$G_c(s) = K\left(1 + \frac{1}{sT_i}\right)$$

a second-order closed-loop system is obtained:

$$G(s) = \frac{G_p G_c}{1 + G_c G_p}$$

The two closed-loop poles can be chosen arbitrarily by a suitable choice of the gain (K) and the integral time (T_i) of the controller. This is seen as follows. The poles are given by the characteristic equation,

$$1 + G_c G_p = 0$$

The characteristic equation becomes

$$s^2 + s\,\frac{1 + K_p K}{T} + \frac{K_p K}{T T_i} = 0$$

Now suppose that the desired closed-loop poles are characterized by their relative damping (ζ) and their frequency (ω_0). The desired characteristic equation then becomes

$$s^2 + 2\zeta\omega_0 s + \omega_0^2 = 0$$

Making the coefficients of these two characteristic equations equal gives two equations for determining K and T_i:

$$\omega_0^2 = \frac{K_p K}{T T_i}$$
$$2\zeta\omega_0 = \frac{1 + K_p K}{T}$$

Solving these for the controller parameters, we get

$$K = \frac{2\zeta\omega_0 T - 1}{K_p}$$
$$T_i = \frac{2\zeta\omega_0 T - 1}{\omega_0^2 T}$$

Notice that the transfer function from setpoint to process output has a zero at $s = -1/(bT_i)$. To avoid excessive overshoot in the setpoint response, parameter b should be chosen so that the zero is to the left of the dominant closed-loop poles. A reasonable value is $b = 1/(\omega_0 T_i)$, which places the zero at $s = -\omega_0$. Notice also that in order to have positive controller gains it is necessary that the chosen frequency (ω_0) is larger than $1/(2\zeta T)$. It also follows that if ω_0 is large, the integral time T_i is given by

$$T_i \approx \frac{2\zeta}{\omega_0}$$

and is, thus, independent of the process dynamics for large ω_0. There is no formal upper bound to the bandwidth. However, a simplified model like Equation (4.60) will not hold for large frequencies. The upper bound on the bandwidth is determined, therefore, by the validity of the model. □

EXAMPLE 4.14 System with two real poles

Suppose that the process is characterized by the second-order model

$$G_p = \frac{K_p}{(1 + sT_1)(1 + sT_2)} \tag{4.61}$$

This model has three parameters. By using a PID controller, which also has three parameters, it is possible to arbitrarily place the three poles of the closed-loop system. The transfer function of the PID controller can be written as

$$G_c(s) = \frac{K(1 + sT_i + s^2T_iT_d)}{sT_i}$$

The characteristic equation of the closed-loop system becomes

$$s^3 + s^2\left(\frac{1}{T_i} + \frac{1}{T_2} + \frac{K_pKT_d}{T_1T_2}\right) + s\left(\frac{1}{T_1T_2} + \frac{K_pK}{T_1T_2}\right) + \frac{K_pK}{T_1T_2T_i} = 0 \quad (4.62)$$

A suitable closed-loop characteristic equation of a third-order system is

$$(s + \alpha\omega_0)(s^2 + 2\zeta\omega_0 s + \omega_0^2) = 0 \quad (4.63)$$

which contains two dominant poles with relative damping (ζ) and frequency (ω_0), and a real pole located in $-\alpha\omega_0$. Identifying the coefficients of equal powers of s in the Equations (4.62) and (4.63) gives

$$\begin{aligned}
\frac{1}{T_i} + \frac{1}{T_2} + \frac{K_pKT_d}{T_1T_2} &= \omega_0(\alpha + 2\zeta) \\
\frac{1}{T_1T_2} + \frac{K_pK}{T_1T_2} &= \omega_0^2(1 + 2\zeta\omega_0) \\
\frac{K_pK}{T_1T_2T_i} &= \alpha\omega_0^3
\end{aligned}$$

Solving these equations gives the following controller parameters

$$\begin{aligned}
K &= \frac{T_1T_2\omega_0^2(1 + 2\alpha\zeta) - 1}{K_p} \\
T_i &= \frac{T_1T_2\omega_0^2(1 + 2\alpha\zeta) - 1}{T_1T_2\alpha\omega_0^3} \\
T_d &= \frac{T_1T_2\omega_0(\alpha + 2\zeta) - T_1 - T_2}{T_1T_2\omega_0^2(1 + 2\alpha\zeta) - 1}
\end{aligned}$$

Provided that $c = 0$, the transfer function from setpoint to process output has one zero at $s = -1/(bT_i)$. To avoid excessive overshoot in the setpoint response, parameter b can be chosen so that this zero cancels the pole at $s = -\alpha\omega_0$. This gives

$$b = \frac{1}{\alpha\omega_0T_i} = \frac{\omega_0^2T_1T_2}{\omega_0^2T_1T_2(1 + 2\alpha\zeta) - 1}$$

Also, notice that pure PI control is obtained for

$$\omega_0 = \omega_c = \frac{T_1 + T_2}{(\alpha + 2\zeta)T_1T_2}$$

The choice of ω_0 may be critical. The derivative time is negative for $\omega_0 < \omega_c$. Thus, the frequency (ω_c) gives a lower bound to the bandwidth. The gain increases rapidly with ω_0. The upper bound to the bandwidth is given by the validity of the simplified model (Equation 4.61). □

The methods BO and SO, discussed in the previous section, can clearly be interpreted as pole placement methods. The desired closed-loop characteristic polynomial is

$$A_{\mathrm{BO}}(s) = s^2 + \sqrt{2}\omega_0 s + \omega_0^2$$

for the modulus optimum and

$$A_{\mathrm{SO}}(s) = (s + \omega_0)(s^2 + \omega_0 s + \omega_0^2)$$

for the symmetrical optimum.

The calculations in Example 4.14 can be done for any linear system. The algebraic formulas obtained, however, may be quite complicated. Another useful example follows.

EXAMPLE 4.15 Second-order systems with a zero

Suppose that the process is characterized by the second-order model

$$G_p = \frac{b_1 s + b_2}{s^2 + a_1 s + a_2} \tag{4.64}$$

This model has four parameters. It has two poles that may be real or complex, and it has one zero. The model given by Equation (4.64) captures many processes, oscillatory systems, and systems with right half-plane zeros. The right half-plane zero can also be used as an approximation of a time delay. We assume that the process is controlled by a PID controller parameterized as

$$G_c(s) = k + \frac{k_i}{s} + k_d s \tag{4.65}$$

The closed-loop system is of third order and has the characteristic equation

$$s(s^2 + a_1 s + a_2) + (b_1 s + b_2)(k_d s^2 + k s + k_i) = 0 \tag{4.66}$$

A suitable closed-loop characteristic equation of a third-order system is

$$(s + \alpha\omega_0)(s^2 + 2\zeta\omega_0 s + \omega_0^2) = 0 \tag{4.67}$$

Equating coefficients of equal power in s in Equations (4.66) and (4.67) gives the following equations:

$$a_1 + b_2 k_d + b_1 k = (\alpha\omega_0 + 2\zeta\omega_0)(1 + b_1 k_d)$$

$$a_2 + b_2 k + b_1 k_i = (1 + 2\alpha\zeta)\omega_0^2(1 + b_1 k_d)$$

$$b_2 k_i = \alpha\omega_0^3(1 + b_1 k_d)$$

This is a set of linear equations in the controller parameters. The solution is straightforward but tedious and is given by

$$k = \frac{a_2 b_2^2 - a_2 b_1 b_2(\alpha + 2\zeta)\omega_0 - (b_2 - a_1 b_1)(b_2(1 + 2\alpha\zeta)\omega_0^2 + \alpha b_1 \omega_0^3)}{b_2^3 - b_1 b_2^2(\alpha + 2\zeta)\omega_0 + b_1^2 b_2(1 + 2\alpha\zeta)\omega_0^2 - \alpha b_1^3 \omega_0^3}$$

$$k_i = \frac{(-a_1 b_1 b_2 + a_2 b_1^2 + b_2^2)\alpha\omega_0^3}{b_2^3 - b_1 b_2^2(\alpha + 2\zeta)\omega_0 + b_1^2 b_2(1 + 2\alpha\zeta)\omega_0^2 - \alpha b_1^3 \omega_0^3}$$

$$k_d = \frac{-a_1 b_2^2 + a_2 b_1 b_2 + b_2^2(\alpha + 2\zeta)\omega_0 - b_1 b_2 \omega_0^2(1 + 2\alpha\zeta) + b_1^2 \alpha\omega_0^3}{b_2^3 - b_1 b_2^2(\alpha + 2\zeta)\omega_0 + b_1^2 b_2(1 + 2\alpha\zeta)\omega_0^2 - \alpha b_1^3 \omega_0^3}$$

These formulas are quite useful because many processes can be approximately described by the transfer function given by Equation (4.64). □

The formulas given in Example 4.15 are particularly useful in cases when we are "stretching" the PID controller to extreme situations. The standard tuning rules will typically not work in these cases. Typical examples are systems with zeros in the right half-plane and systems with poorly damped oscillatory modes. To illustrate this we will consider an example.

EXAMPLE 4.16 A difficult process

Consider a system with the transfer function

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

This system has one right half-plane zero and two undamped complex poles. None of the standard methods for tuning PID controllers work well for this system. To apply the pole-placement method we require that the closed-loop system has the characteristic equation

$$s^3 + 2s^2 + 2s + 1 = 0$$

The formulas in Example 4.15 give a controller with the parameters $k = 0$, $k_i = 1/3$, and $k_d = 2/3$. This can also be verified with a simple calculation. Notice that the proportional gain is zero and that the controller has two complex zeros at $\pm i\sqrt{2}$. Such a controller can only be implemented with a PID controller having the parallel form. Compare with section 3.4. □

The General Case

The calculations in the examples can be extended to general linear systems. They are, however, more complicated. It is necessary to spec-

ify more closed-loop poles. Some pole patterns that are used are Butterworth configurations, where the roots of the characteristic polynomials are placed symmetrically in a circle, and the Bessel configurations, which correspond to filters that attempt to preserve the shape of the wave form. The order of the controllers also increases with the complexity of the model. To obtain PID controllers it is necessary to restrict the models to first- or second-order systems. For more complex processes, it is, therefore, necessary to make approximations so that a process model in the form of a rational function of first or second order is obtained. Several ways to perform these approximations aree given in Chapter 2. We illustrate the procedure with an example.

EXAMPLE 4.17 Pole placement with an approximate model

Consider a process described by the transfer function

$$G_p(s) = \frac{1}{(1+s)(1+0.2s)(1+0.05s)(1+0.01s)} \tag{4.68}$$

This process has four lags with time constants 1, 0.2, 0.05, and 0.01. The approximations can be done in several different ways. If the control requirements are not too severe, we can attempt to approximate the transfer function by

$$G_p(s) = \frac{1}{1+1.26s}$$

where the time constant is the average residence time of the system. As discussed in Section 2.9, this approximation is good at low frequencies. The phase error is less than 10° for frequencies below 1.1 rad/s. Designing a PI controller with the pole placement method with $\zeta = 0.5$, the following controller parameters are obtained

$$K = 1.26\omega_0 - 1$$

$$T_i = \frac{1.26\omega_0 - 1}{1.26\omega_0^2}$$

$$b = \frac{1.26\omega_0}{1.26\omega_0 - 1}$$

where b is chosen so that the zero becomes $s = -\omega_0$. If the process model would be correct, the phase margin with $\zeta = 0.5$ would be 50°. Because of the approximations made, the phase margin will be less. It will decrease with ω_0. For $\omega_0 = 1$ the phase margin is $\varphi_m = 42°$. □

Another way of applying pole placement design is given in the next example.

EXAMPLE 4.18 Application to an approximate model

Consider the same process model as in the previous example (Equation 4.68). Approximate the transfer function by

$$G_p(s) = \frac{1}{(1+s)(1+0.26s)}$$

It is obtained by keeping the longest time constant and approximating the three shorter time constants with their sum. The phase error is less than 10° for frequencies below 5.1 rad/s. By making an approximation of the process model that is valid for higher frequencies than in the previous example, we can thus design a faster controller. If $\zeta = 0.5$ and $\alpha = 1$ are chosen in Equation (4.67), the design calculations in Example 4.14 gives the following PID parameters:

$$K = 0.52\omega_0^2 - 1$$

$$T_i = \frac{0.52\omega_0^2 - 1}{0.26\omega_0^3}$$

$$T_d = \frac{0.52\omega_0 - 1.26}{0.52\omega_0^2 - 1}$$

$$b = \frac{0.26\omega_0^2}{0.52\omega_0^2 - 1}$$

In this case, pure PI control is obtained for $\omega_0 = 2.4$. The derivative gain becomes negative for lower bandwidths. The approximation neglects the time constant 0.05. If the neglected dynamics are required to give a phase error of, at most, 0.3 rad (17 deg) at the bandwidth, $\omega_0 < 6$ rad/s can be obtained. In Figure 4.23, the behavior of the control is demonstrated for $\omega_0 = 4, 5$, and 6.

The specification of the desired closed-loop bandwidth is crucial, since the controller gain increases rapidly with the specified bandwidth. It is also crucial to know the frequency range where the model is valid. Alternatively, an upper bound to the controller gain can be used to limit the bandwidth. Notice the effect of changing the design frequency (ω_0). The system with $\omega_0 = 6$ responds faster and has a smaller error when subjected to load disturbances. The design will not work well when ω_0 is increased above 8. □

4.8 Dominant Pole Design

In pole placement design it is attempted to assign all closed-loop poles. One difficulty with the method is that complex models lead to complex controllers. In this section we will introduce a related method

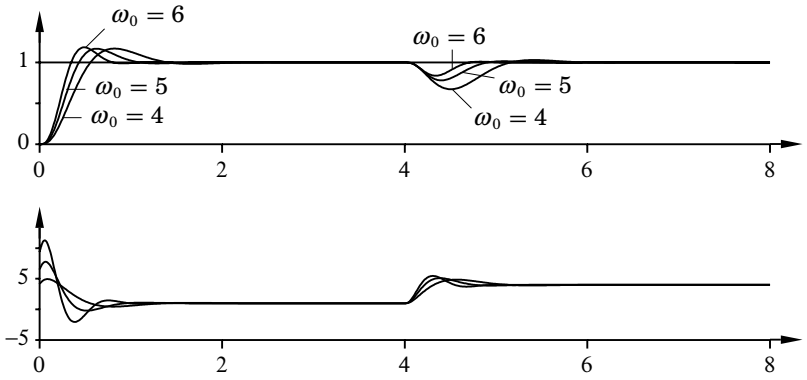

Figure 4.23 Setpoint and load disturbance responses of the process (Equation 4.68) controlled by a PID controller tuned according to Example 4.18. The responses for $\omega_0 = 4, 5$, and 6 are shown. The upper diagram shows setpoint $y_{sp} = 1$ and process output y, and the lower diagram shows control signal u.

where it is attempted to assign only a few poles. With this method it is possible to design simple controllers for complex processes. The method is based on the assumption that the transfer function of the process is known. The idea of positioning a few closed-loop poles was used in several of the early papers on PID control. A complete design methodology based on this idea is developed in this section. The method makes it possible to consider many different specifications. It is also possible to design controllers of many different types and to compare their performances.

The Cohen-Coon Method

The Cohen-Coon method is based on the process model

$$G_p = \frac{K_p}{1 + sT} e^{-sL}$$

The main design criterion is rejection of load disturbances. It attempts to position dominant poles that give a quarter amplitude decay ratio. For P and PD controllers the poles are adjusted to give maximum gain, subject to the constraint on the decay ratio. This minimizes the steady state error due to load disturbances. For PI and PID control the integral gain $k_i = K/T_i$ is maximized. This corresponds to minimization of IE, the integral error due to a unit step load disturbance. For PID controllers three closed-loop poles are assigned; two poles are complex, and the third real pole is positioned at the same distance from the origin as the other poles. The pole pattern is

Table 4.8 Controller parameters from the Cohen-Coon method.

Controller	K	T_i	T_d
P	$\frac{1}{a}\left(1+\frac{0.35\tau}{1-\tau}\right)$		
PI	$\frac{0.9}{a}\left(1+\frac{0.92\tau}{1-\tau}\right)$	$\frac{3.3-3.0\tau}{1+1.2\tau}L$	
PD	$\frac{1.24}{a}\left(1+\frac{0.13\tau}{1-\tau}\right)$		$\frac{0.27-0.36\tau}{1-0.87\tau}L$
PID	$\frac{1.35}{a}\left(1+\frac{0.18\tau}{1-\tau}\right)$	$\frac{2.5-2.0\tau}{1-0.39\tau}L$	$\frac{0.37-0.37\tau}{1-0.81\tau}L$

adjusted to give quarter amplitude decay ratio, and the distance of the poles to the origin are adjusted to minimize IE.

Since the process is characterized by three parameters (K_p, L, and T), it is possible to give tuning formulas where controller parameters are expressed in terms of these parameters. Such formulas were derived by Cohen and Coon based on analytical and numerical computations. The formulas are given in Table 4.8. The parameters $a = K_pL/T$ and $\tau = L/(L+T)$ are used in the table. A comparison with Table 4.1 shows that the controller parameters are close to those obtained by the Ziegler-Nichols step response method for small τ. Also notice that the integral time decreases for increasing τ which is desirable as was found in Section 4.3. The method does suffer, however, from the decay ratio being too small, which means that the closed-loop systems obtained have low damping and high sensitivity.

Integrating Control

Consider a process with transfer function $G_p(s)$ controlled by an integrating controller. Such a controller has the transfer function

$$G_c(s) = \frac{k_i}{s}$$

The closed-loop poles are given by

$$1 + k_i \frac{G_p(s)}{s} = 0$$

Since the controller has one adjustable parameter, it is possible to assign one pole. To obtain a pole at $s = -a$ the controller parameter should be chosen as

$$k_i = \frac{a}{G_p(-a)} \tag{4.69}$$

So far parameter a is a design parameter. To find suitable ways of choosing a it is observed that gain k_i is small when a is small. This means that the closed-loop poles are the design pole at $s = -a$ and a number of poles that are close to the open-loop poles. If the open-loop system is stable, the design pole is thus the slowest pole. Increasing a gives a faster closed-loop system. Notice that k_i becomes zero when a is equal to a real process pole. Since $IE = 1/k_i$, the integrated error will also decrease with increasing a. One possible way to choose a is to use a value that maximizes k_i.

PI Control

A PI controller has two parameters. Consequently, it is necessary to assign two poles. Consider a process with transfer function $G_p(s)$ and let the controller be parameterized as

$$G_c(s) = k + \frac{k_i}{s}$$

The closed-loop characteristic equation is

$$1 + \left(k + \frac{k_i}{s}\right) G_p(s) = 0 \tag{4.70}$$

Require that this equation have roots at

$$p_{1,2} = \omega_0 \left(-\zeta_0 \pm i\sqrt{1-\zeta_0^2}\right) = \omega_0 e^{i(\pi \pm \gamma)} = \omega_0(-\cos\gamma \pm i \sin\gamma) \tag{4.71}$$

where $\gamma = \arccos \zeta_0$. This gives

$$1 + \left(k + \frac{k_i}{p_1}\right) G_p(p_1) = 0$$

Introduce $a(\omega_0)$ and $\phi(\omega_0)$ defined as

$$G_p\left(\omega_0 e^{i(\pi-\gamma)}\right) = a(\omega_0) e^{i\phi(\omega_0)}$$

Notice that $G_p\left(\omega_0 e^{i(\pi-\gamma)}\right)$ represents the values of the transfer function on the ray $e^{i(\pi-\gamma)}$. When $\gamma = \pi/2$, then $G_p\left(\omega_0 e^{i(\pi-\gamma)}\right) = G_p(i\omega_0)$, which is the normal frequency response.

Equation (4.70) can be written as

$$1 + \left(k + \frac{k_i}{\omega_0 e^{i(\pi-\gamma)}}\right) a(\omega_0) e^{i\phi(\omega_0)} = 0$$

This equation, which is linear in k and k_i, has the solution

$$k = -\frac{\sin(\phi(\omega_0) + \gamma)}{a(\omega_0)\sin\gamma} \tag{4.72}$$

$$k_i = -\frac{\omega_0 \sin\phi(\omega_0)}{a(\omega_0)\sin\gamma} \tag{4.73}$$

Notice that $\phi(\omega_0)$ is zero for $\omega_0 = 0$ and typically negative as ω_0 increases. This implies that the proportional gain is negative and the integral gain positive but small for small ω_0. When ω_0 increases both k and k_i will increase initially. For larger values of ω_0 both parameters will decrease. Requiring that both parameters are positive, we find that ω_0 must be selected so that

$$\gamma < -\phi(\omega_0) < \pi$$

The integral time of the controller is

$$T_i = \frac{k}{k_i} = \frac{\sin(\phi(\omega_0) + \gamma)}{\omega_0 \sin \phi(\omega_0)} \tag{4.74}$$

Notice that T_i is independent of $a(\omega_0)$.

PD Control

A PD controller has two parameters. To obtain these it is necessary to specify two closed-loop poles. The controller is assumed to be parameterized as

$$G_c(s) = k + k_d s$$

Specifying the desired poles as

$$p_{1,2} = \omega_0 \left(-\zeta_0 \pm i\sqrt{1 - \zeta_0^2} \right)$$

and proceeding as in the derivation of the PI controller we find

$$k = \frac{\sin(\phi(\omega_0) - \gamma)}{a(\omega_0) \sin \gamma} \tag{4.75}$$

$$k_d = \frac{\sin \phi(\omega_0)}{\omega_0 a(\omega_0) \sin \gamma} \tag{4.76}$$

Note that the expressions of k and k_d for PD controllers are similar to those of PI controllers.

PID Control

Now we consider PID control. For simplicity, it is assumed that the controller is parameterized as

$$G_c(s) = k + \frac{k_i}{s} + k_d s \tag{4.77}$$

Modifications in the setpoint weighting and limitation of the derivative gain are taken care of later. Since the controller has three parameters, it is necessary to specify three poles of the closed-loop system.

We choose them as

$$p_{1,2} = \omega_0 \left(-\zeta_0 \pm i\sqrt{1-\zeta_0^2} \right) \tag{4.78}$$

$$p_3 = -\alpha_0 \omega_0. \tag{4.79}$$

Introduce the quantities $a(\omega_0)$, $b(\omega_0)$, and $\phi(\omega_0)$ defined by

$$G_p\left(\omega_0 e^{i(\pi-\gamma)}\right) = a(\omega_0)e^{i\phi(\omega_0)}$$

$$G_p(-\alpha\omega_0) = -b(\omega_0)$$

The condition that p_1, p_2, and p_3 are roots of

$$1 + G_p(s)G_c(s) = 0 \tag{4.80}$$

gives the conditions

$$k = -\frac{\alpha_0^2 b(\omega_0)\sin(\gamma+\phi) + b(\omega_0)\sin(\gamma-\phi) + \alpha_0 a(\omega_0)\sin 2\gamma}{a(\omega_0)b(\omega_0)(\alpha_0^2 - 2\alpha_0\cos\gamma + 1)\sin\gamma}$$

$$k_i = -\alpha_0\omega_0 \frac{a(\omega_0)\sin\gamma + b(\omega_0)(\sin(\gamma-\phi) + \alpha_0\sin\phi)}{a(\omega_0)b(\omega_0)(\alpha_0^2 - 2\alpha_0\cos\gamma + 1)\sin\gamma}$$

$$k_d = -\frac{\alpha_0 a(\omega_0)\sin\gamma + b(\omega_0)(\alpha_0\sin(\gamma+\phi) - \sin\phi)}{\omega_0 a(\omega_0)b(\omega_0)(\alpha_0^2 - 2\alpha_0\cos\gamma + 1)\sin\gamma}$$

PID Controller Based on PI Controller

Another way to obtain a PID controller is to start with a PI controller and to add derivative action. This can be done as follows. Assume that the controller

$$G_c(s) = k + \frac{k_i}{s} + k_d s \tag{4.81}$$

is used and that it is desired to have two closed-loop poles in

$$p_{1,2} = \omega_0 \left(-\zeta_0 \pm i\sqrt{1-\zeta_0^2} \right)$$

The value of the controller transfer function at these poles can be written as

$$\begin{aligned} G_c(p_1) &= k + \frac{k_i}{\omega_0} e^{-i(\pi-\gamma)} + k_d\omega_0 e^{i(\pi-\gamma)} \\ &= k - \left(k_d\omega_0 + \frac{k_i}{\omega_0}\right)\cos\gamma + i\left(k_d\omega_0 - \frac{k_i}{\omega_0}\right)\sin\gamma \end{aligned}$$

This implies that

$$k = k' + 2k_d\zeta_0\omega_0$$

$$k_i = k_i' + k_d\omega_0^2$$

where k' and k_i' are the controller parameters for a pure PI controller given by Equations (4.72) and (4.73). Using this parameterization the PID controller can be written

$$G_c(s) = G_c'(s) + \frac{k_d}{s}\left(s^2 + 2\zeta_0\omega_0 s + \omega_0^2\right) \tag{4.82}$$

where $G_c'(s)$ is a pure PI controller, i.e., $G_c(s)$ with $k_d = 0$. The task is now to choose ζ_0, ω_0, and k_d such that the system behaves well.

The characteristic equation of the closed-loop system becomes

$$1 + G_p(s)G_c(s) = 1 + G_p(s)\left(G_c'(s) + \frac{k_d}{s}\left(s^2 + 2\zeta_0\omega_0 s + \omega_0^2\right)\right) \tag{4.83}$$

For a system controlled by a PI controller we have

$$1 + G_p(s)G_c'(s) = \left(s^2 + 2\zeta_0\omega_0 s + \omega_0^2\right)R(s)$$

The zeros of $R(s)$ are the free poles of the system controlled by the PI controller $G_c'(s)$. Thus,

$$1 + G_p(s)G_c(s) = \left(s^2 + 2\zeta_0\omega_0 s + \omega_0^2\right)\left(R(s) + G_p(s)\frac{k_d}{s}\right) \tag{4.84}$$

The root locus of $1 + G_p(s)G_c(s)$ with respect to k_d will start in the zeros of $R(s)$ and end in the zeros of $G_p(s)$ or in infinity.

This parameterization offers a natural way to tune a PID controller: start with a well tuned PI controller and add derivative action. As k_d is increased the parameter ω_0 may have to be modified, e.g., in such a way that IE is maximized.

PID Controller Based on PD controller

Another way to obtain a PID controller is to start with a PD controller and to add integral action. Proceeding in the same way as previously but starting with a PD controller we get

$$k = k'' + \frac{2k_i\zeta_0}{\omega_0}$$

$$k_d = k_d'' + \frac{k_i}{\omega_0^2}$$

where k'' and k_d'' are the controller parameters for a pure PD controller given by Equations (4.75) and (4.76).

A Design Procedure

We have shown that it is possible to find controllers that assign as many closed-loop poles as there are free parameters in the controller.

The calculations required are simply a solution of a set of linear equations. For a PI controller it is possible to assign a pair of complex poles with given frequency ω_0 and relative damping ζ_0. The assigned poles will be dominating if the frequency is sufficiently small. We can use this idea to develop systematic design procedures. To do so we start with a set of specifications and design parameters. The specifications considered are to express load disturbance rejection, sensitivity to measurement noise and process variations and setpoint following as discussed in Section 4.2.

Method DPD1: Frequency and Damping as Design Parameters

One possibility is to use frequency ω_0 and relative damping ζ_0 as design parameters. Other specifications then have to be translated to conditions on frequency and damping using the relations in Section 4.2.

Method DPD2: Relative Damping as Design Parameter

It is comparatively easy to give reasonable values of relative damping ζ_0. Good values are in the range of 0.4 to 1.0. It is much more difficult to find reasonable values of frequency ω_0. This parameter may change by many orders of magnitude depending on the process. It would be useful, therefore, to determine parameter ω_0 automatically. This can be done by selecting a value that gives good rejection of load disturbances. Notice that the integrated error IE is related to parameter k_i through

$$IE = \frac{1}{k_i}$$

Hence, k_i should be maximized in order to minimize IE. The design procedure then becomes a bit more complicated because the controller parameters have to be computed for different values of ω_0, and an optimization has to be performed. Example 4.7. illustrates this. Notice that if relative damping is specified in a reasonable way, the criterion IE is a good measure of the rejection of load disturbances. An alternative is to consider the criterion IAE instead. The computational burden then increases significantly and the improvement in performance is marginal. When doing the optimization it must also be checked that measurement noise does not generate too much control action. This can be expressed by the constraint

$$K_{hf} = K(1 + N) < K_{\max} \tag{4.85}$$

It is straightforward to consider this constraint in the optimization. The optimization then also tells if performance is limited by process dynamics or measurement noise. This information is useful in order

to direct redesign of the system. Also notice that the design gives the frequency ω_0 as a result. This indicates the frequency range where the process model has to be reasonably accurate.

Method DPD3: Sensitivity as Design Parameter

One drawback of the design methods given is that sensitivity is only considered indirectly in the design through the specification of relative damping ζ_0. Another drawback is that for some systems it is possible to obtain significantly better attenuation of load disturbances by decreasing damping. One design method that takes this into account uses sensitivity M_s as a design parameter. The design is then carried out in the following way. We first fix ζ_0 and perform the design as in Method DPD2, but we also compute the sensitivity M_s. The parameter ζ_0 is then changed in order to maximize k_i subject to the constraints on sensitivity and measurement noise, see Equation (4.85). This method requires more computations than the previous methods, but has been shown to give very good results. One particular feature of this method is that the behaviors of the closed-loop system are very similar for many different processes. Another feature is that it gives values of both frequency and damping as intermediate results. The frequency gives an indication of the frequency ranges where model accuracy is needed. By comparing the values of k_i, K_{hf}, and ω_0, it is also possible to make an assessment of the performances of controllers having different structures, e.g., to compare PI and PID controllers.

Also notice that, if k_i can be maximized without violating the constraint on measurement noise, the method is equivalent to the loop shaping procedure discussed in Section 4.4, where IE was minimized subject to constraints on sensitivity M_s.

Examples

We illustrate the design method with a few examples.

EXAMPLE 4.19 A pure dead-time process

Consider a process with the transfer function

$$G_p(s) = e^{-sL}$$

Using pure integral control, it follows from Equation (4.69) that

$$k_i = ae^{-aL}$$

Table 4.9 Controller parameters obtained in Example 4.19.

ζ_0	k	k_iL	T_i/L	$\omega_0 L$	M_s	IAE/L
0.1	0.388	1.50	0.258	1.97	6.34	4.03
0.2	0.343	1.27	0.270	1.93	3.60	2.42
0.3	0.305	1.09	0.279	1.89	2.70	1.89
0.4	0.273	0.956	0.285	1.87	2.25	1.67
0.5	0.244	0.847	0.288	1.86	1.99	1.56
0.6	0.218	0.759	0.288	1.86	1.81	1.52
0.707	0.195	0.688	0.284	1.88	1.69	1.54
0.8	0.174	0.629	0.276	1.90	1.61	1.61
0.9	0.154	0.581	0.265	1.94	1.54	1.72
1.0	0.135	0.541	0.250	2.00	1.49	1.85

The gain has its largest value $k_i = e^{-1}/L$ for $a = 1/L$. The loop transfer function for the system is then

$$G_\ell(s) = \frac{1}{sL}\, e^{-(sL+1)}$$

The sensitivity of the system is $M_s = 1.39$, which is a good value.

Let us now consider PI control of the process. To do this we first must evaluate the transfer function on the ray $\omega_0 e^{\pi-\gamma}$. We have

$$G_p(\omega_0 e^{\pi-\gamma}) = G_p(-\omega_0 \cos\gamma + i\omega_0 \sin\gamma) = e^{\omega_0 L \cos\gamma} e^{-i\omega_0 L \sin\gamma}$$

Hence,

$$a(\omega_0) = e^{\omega_0 L \cos\gamma}$$
$$\phi(\omega_0) = -\omega_0 L \sin\gamma$$

It follows from Equations (4.72) and (4.73) that

$$k = \frac{\sin(\omega_0 L \sin\gamma - \gamma)}{\sin\gamma}\, e^{-\omega_0 L \cos\gamma}$$
$$k_i = \omega_0 \frac{\sin(\omega_0 L \sin\gamma)}{\sin\gamma}\, e^{-\omega_0 L \cos\gamma}$$

To minimize IE, we determine the value of ω_0 that maximizes k_i. Setting the derivative of k_i with respect to ω_0 equal to zero we get

$$\sin(\omega_0 L \sin\gamma) = \omega_0 L\,(\sin(\omega_0 L \sin\gamma)\zeta_0 - \cos(\omega_0 L \sin\gamma)\sin\gamma)$$

Solving this equation with respect to $\omega_0 L$ for different values of γ, we find the controller parameters given in Table 4.9. Table 4.9 also

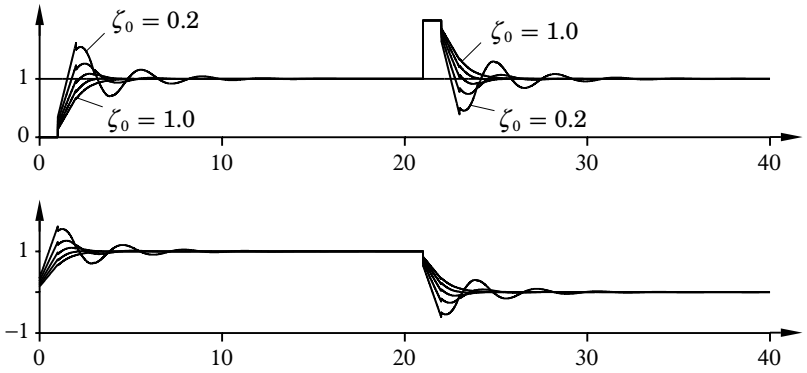

Figure 4.24 Simulation of different controllers for a pure delay process. The relative dampings are ζ_0=0.2, 0.4, 0.6, 0.8, and 1.0, respectively. The upper diagram shows setpoint $y_{sp} = 1$, process output y, and the lower diagram shows control signal u.

gives the M_s values and the IAE. The M_s value is reasonable for $\zeta_0 \geq 0.5$. The IEA has its minimum for $\zeta_0 = 0.6$. In particular we notice that for $\zeta_0 = 1$ we get $k = e^{-2}$ and $k_i = 4e^{-2}/L$. This can be compared with $k_i = e^{-1}L$ for pure I control. With PI control the integral gain can thus be increased by a factor of 1.5 compared with an I controller. Notice that for a well-damped system ($\zeta_0 = 0.707$) the gain is about 0.2 and the integral time is $T_i = 0.28L$. This can be compared with the values 0.9 and $3.3L$ given by the Ziegler-Nichols step response method, 0.45 and 2L by the Ziegler-Nichols frequency response method, and 0.083 and $0.14L$ for the Cohen-Coon method. Figure 4.24 shows a simulation of controllers with different parameters. In summary, we find that a process with a pure delay dynamics can be controlled quite well with a PI controller. Notice, however, that the tuning cannot be done by the Ziegler-Nichols rules. □

In the next example, the dominant pole design method is applied to the same process as the Ziegler-Nichols methods in Section 4.3.

EXAMPLE 4.20 Three equal lags

Consider a process with the transfer function

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

Table 4.10 gives the controller parameters obtained with the dominant pole design method that uses M_s as a tuning parameter for a PI controller. Figure 4.25 shows simulations with controllers obtained

Table 4.10 PI controller parameters obtained in Example 4.20.

M_s	ω_0	ζ_0	K	k_i	T_i	IAE
1.2	0.39	1.16	0.18	0.14	1.24	6.81
1.4	0.59	0.67	0.48	0.31	1.53	3.29
1.6	0.67	0.49	0.71	0.45	1.58	2.54
1.8	0.73	0.39	0.92	0.57	1.60	2.20
2.0	0.78	0.33	1.09	0.68	1.60	2.06
2.2	0.82	0.28	1.24	0.77	1.60	1.97
2.4	0.85	0.25	1.37	0.86	1.60	1.90
2.6	0.88	0.22	1.49	0.93	1.60	1.86
2.8	0.90	0.20	1.59	1.00	1.59	1.85
3.0	0.92	0.18	1.67	1.06	1.59	1.87

with different values of M_s. The behavior is good for values of M_s in the interval $1.4 \leq M_s \leq 2.0$. For higher values of M_s, the responses become oscillatory. For $M_s = 1.2$, the value of ζ_0 is greater than one. This means that the two dominant poles are real. The weighting factor $b = 1$ is used in the simulation. A smaller value of b would give responses to setpoint changes with a smaller overshoot.

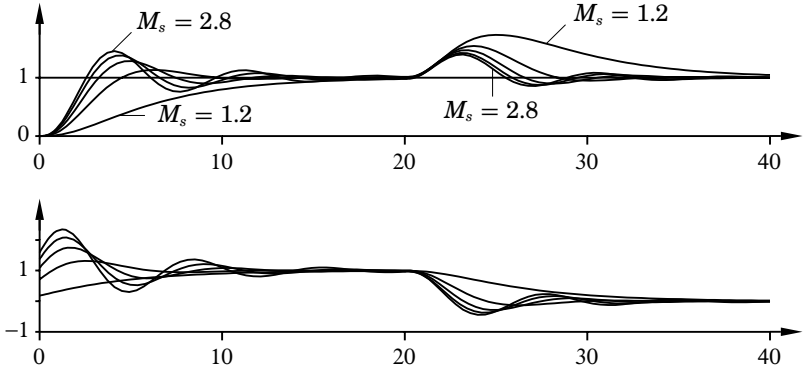

Figure 4.25 Simulation of different PI controllers for a process with transfer function $1/(s+1)^3$. The M_s values are M_s=1.2, 1.6, 2.0, 2.4, and 2.8. The upper diagram shows setpoint $y_{sp} = 1$ and process output y, and the lower diagram shows control signal u.

Table 4.11 PID controller parameters obtained in Example 4.20.

M_s	ω_0	ζ_0	KK_p	k_iK_p	T_i	T_d	IAE/K_p
1.2	0.83	0.85	0.99	0.43	2.27	0.57	2.93
1.4	1.07	0.55	2.05	0.93	2.21	0.54	1.08
1.6	1.22	0.42	2.89	1.37	2.11	0.50	0.74
1.8	1.33	0.34	3.61	1.77	2.04	0.47	0.60
2.0	1.42	0.29	4.24	2.14	1.98	0.44	0.52
2.2	1.49	0.26	4.80	2.47	1.94	0.42	0.48
2.4	1.54	0.23	5.29	2.77	1.91	0.41	0.44
2.6	1.59	0.21	5.74	3.04	1.89	0.40	0.42
2.8	1.64	0.19	6.14	3.29	1.87	0.39	0.41
3.0	1.67	0.17	6.51	3.52	1.85	0.38	0.40

In Example 4.3, Ziegler-Nichols methods were used to tune the same process. The step response method gave the controller parameters $K = 4.13$ and $T_i = 2.42$, whereas the frequency response method gave $K = 3.2$ and $T_i = 2.90$. Comparing these values with the ones in Table 4.10 shows that the Ziegler-Nichols method gives a PI controller with a far too high gain and a too long integral time.

Table 4.11 gives the controller parameters obtained with the dominant pole design method that uses M_s as a tuning parameter for a PID controller, and Figure 4.26 shows the results of the simulation. The load disturbance rejection is good for M_s values greater than 1.2. The IAE is significantly smaller than for corresponding PI controllers. The setpoint responses give too large overshoots, since the setpoint weighting is chosen to $b = 1$. □

EXAMPLE 4.21 Multiple lag process

Consider a process with the transfer function

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

To design an integrating controller, it follows from Equation (4.69) that

$$k_i = a(1-a)^n$$

Taking derivatives with respect to a gives

$$\frac{dk_i}{da} = (1-a)^n - na(1-a)^{n-1}$$

The derivative is zero for

$$a = \frac{1}{n+1}$$

The gain has its largest value

$$k_i = \frac{1}{n+1}\left(\frac{n}{n+1}\right)^n$$

The closed-loop characteristic equation is

$$s(s+1)^n + \frac{1}{n+1}\left(\frac{n}{n+1}\right)^n = 0$$

This equation has double roots at $s = -1/(n+1)$.

For $n = 1$ Equations (4.72) and (4.73) give the PI controller parameters

$$k = 2\zeta\omega_0 - 1$$
$$k_i = \omega_0^2$$

Since the closed-loop system is of second order the parameters are the same as those obtained by the pole placement method. Compare with Section 4.7. □

Choosing the Setpoint Weighting

It was shown in Chapter 3 that setpoint weighting is very useful in order to shape the response to setpoint changes. To do this properly, we also need a procedure to determine parameter b. For the dominant pole design method, it is easy to find such a method. With this method,

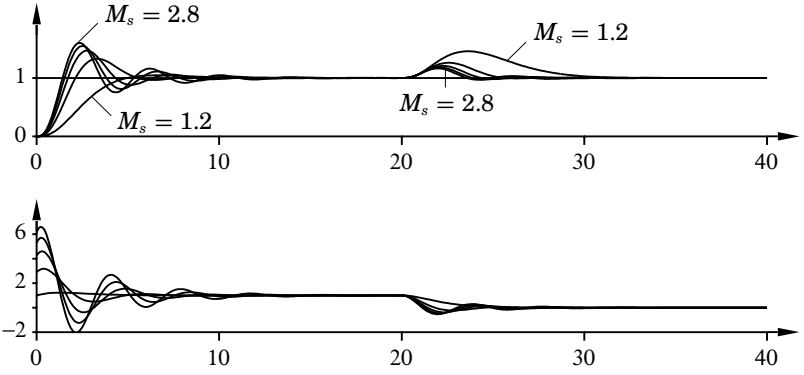

Figure 4.26 Simulation of different PID controllers for a process with transfer function $1/(s+1)^3$. The M_s values are M_s=1.2, 1.6, 2.0, 2.4, and 2.8. The upper diagram shows setpoint $y_{sp} = 1$ and process output y, and the lower diagram shows control signal u.

the closed-loop system will have two complex poles and one pole $-p_0$ on the real axis. This pole may be slower than the dominant poles.

With setpoint weighting, the closed-loop system has a zero at

$$s = -z_0 = -\frac{1}{bT_i}$$

By choosing b so that $z_0 = p_0$, we make sure that the setpoint does not excite the mode corresponding to the pole in $-p_0$. This works well and gives good transient responses for systems where the dominant poles are well damped, ($\zeta_0 > 0.7$). For systems where the poles are not so well damped, the choice $z_0 = 2p_0$ gives systems with less overshoot.

A suitable choice of parameter b is thus

$$b = \begin{cases} \dfrac{0.5}{p_0 T_i} & \text{if} \quad \zeta < 0.5 \\ \dfrac{0.5 + 2.5(\zeta - 0.5)}{p_0 T_i} & \text{if} \quad 0.5 \leq \zeta \leq 0.7 \\ \dfrac{1}{p_0 T_i} & \text{if} \quad \zeta > 0.7 \end{cases}$$

4.9 Design for Disturbance Rejection

The design methods discussed so far have been based on a characterization of process dynamics. The properties of the disturbances have only influenced the design indirectly. A load disturbance in the form of a step was used and in some cases a loss function based on the error due to a load disturbance was minimized. Measurement noise was also incorporated by limiting the high-frequency gain of the controller.

In this section, we briefly discuss design methods that directly attempt to make a trade-off between attenuation of load disturbances and amplification of measurement noise due to feedback.

Consider the system shown in Figure 4.27. Notice that the measurement signal is filtered before it is fed to the controller. Let V and E be the Laplace transforms of the load disturbance and the measurement error, respectively. The process output and the control signal are then given by

$$\begin{aligned} X &= \frac{G_p}{1 + G_\ell} V - \frac{G_\ell}{1 + G_\ell} E \\ U &= -\frac{G_\ell}{1 + G_\ell} V - \frac{G_c G_f}{1 + G_\ell} E \end{aligned} \tag{4.86}$$

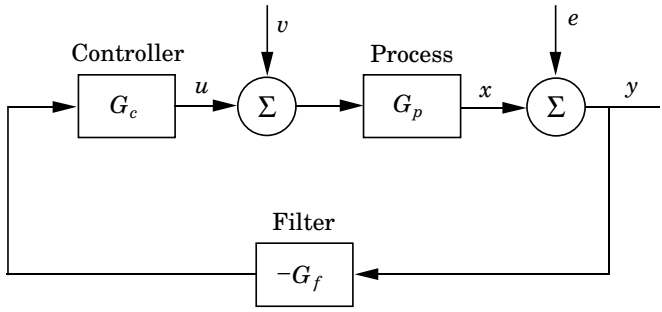

Figure 4.27 Block diagram of a closed-loop system.

where G_ℓ is the loop transfer function given by

$$G_\ell = G_p G_c G_f$$

Different assumptions about the disturbances and different design criteria can now be given. We illustrate by an example.

EXAMPLE 4.22

Assume that the transfer functions in Figure 4.27 are given by

$$G_p = \frac{1}{s} \qquad G_f = 1 \qquad G_c = k + \frac{k_i}{s}$$

Furthermore, assume that e is stationary noise with spectral density ϕ_e and that v is obtained by sending stationary noise with the spectrum ϕ_v through an integrator. This is one way to model the situation that the load disturbance is drifting and the measurement noise has high frequency.

With the given assumptions, Equation (4.86) is simplified to

$$\begin{aligned} X &= \frac{s}{s^2 + ks + k_i} \frac{1}{s} V_1 - \frac{sk + k_i}{s^2 + ks + k_i} E \\ U &= -\frac{sk + k_i}{s^2 + ks + k_i} \frac{1}{s} V_1 - \frac{s^2 k + k_i s}{s^2 + ks + k_i} E \end{aligned} \tag{4.87}$$

where we have assumed

$$V(s) = \frac{1}{s} V_1(s)$$

If e and v_1 are white noises, it follows that the variance of x is given by

$$J = Ex^2 = \frac{1}{2kk_i} \phi_v + \frac{1}{2} \left(k + \frac{k_i}{k} \right) \phi_e$$

This equation clearly indicates the compromise in designing the controller. The first term of the right-hand side is the contribution to the variance due to the load disturbance. The second term represents the contribution due to the measurement noise. Notice that the attenuation of the load disturbances increases with increasing k and k_i, but that large values of k and k_i also increase the contribution of measurement noise.

We can attempt to find values of k and k_i that minimize J. A straightforward calculation gives

$$k = \sqrt{2}\left(\frac{\phi_v}{\phi_e}\right)^{1/4}$$

$$k_i = \sqrt{\frac{\phi_v}{\phi_e}}$$

This means that the controller parameters are uniquely given by the ratio of the intensities of the process noise and the measurement noise. Also notice that with these parameters the closed-loop characteristic polynomial becomes

$$s^2 + \sqrt{2}\omega_0 s + \omega_0^2$$

with $\omega_0 = \sqrt{\phi_v/\phi_e}$. The optimal system thus has a relative damping $\zeta = 0.707$ and a bandwidth that is given by the ratio of the intensities of load disturbance and measurement noise. □

Notice that in Example 4.22 we have a controller that minimizes the variance of the process output. With white measurement noise it follows, however, from Equation (4.87) that the variance of the control signal is infinite. To obtain a control signal with finite variance, we can introduce a filter as shown in Figure 4.27. The effect of that is illustrated with another example.

EXAMPLE 4.23

Consider the same system as in the previous example, but assume that

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

We get

$$\begin{aligned} X &= \frac{s(s+a)}{s^3 + as^2 + aks + ak_i}\,\frac{1}{s}\,V_1 - \frac{aks + ak_i}{s^3 + as^2 + aks + ak_i}\,E \\ U &= -\frac{aks + ak_i}{s^3 + as^2 + aks + ak_i}\,\frac{1}{s}\,V_1 - \frac{aks^2 + ak_i s}{s^3 + as^2 + aks + ak_i}\,E \end{aligned} \tag{4.88}$$

Assuming that v_1 and e are white noise, the variance of the process output becomes

$$J = Ex^2 = \frac{1}{2}\left(\frac{k_i + a^2}{ak_i(ak - k_i)}\phi_v + \frac{ak_i(k^2 + k_i)}{k_i(ak - k_i)}\phi_e\right) \qquad \square$$

4.10 Conclusions

The PID controller is by far the most commonly used control strategy. There are many different methods to find suitable parameters of the controllers. The methods differ in complexity, flexibility, and in the amount of process knowledge used.

There is clearly a need to have several types of tuning methods. We need simple, easy-to-use, intuitive methods that require little information and that give moderate performance. There is also a need for sophisticated methods that give the best possible performance even if they require more information and more computations.

To discuss the methods we must realize that there are many different applications. There are cases where it is desirable to have tight control of the process variable. There are other cases where significant variations in the process variable is permitted. A typical case is surge tanks where the tank level is allowed to fluctuate considerably, as long as the vessel is neither flooded nor empty.

A good tuning method should be based on a rational design method that considers trade-offs between

- Load disturbance attenuation
- Effects of measurement noise
- Robustness to process variations
- Response to setpoint changes
- Model requirements
- Computational requirements

A tuning method should also be widely applicable. It should contain design parameters that influence the performance of the closed-loop system, and it should admit assessment of the differences in performance between PI and PID controllers. The method should also make it possible to judge whether controllers other than PID are more appropriate, and it should be applicable to different types of *a priori* data. Finally, it is desirable that the method be easy to use. Since these requirements are conflicting, it is clear that we need several methods.

The Ziegler-Nichols method is insufficient in spite of being simple and widely used. For the more complex models, it is necessary to have

more information about the process. This can be obtained by fitting rational functions to frequency responses or by applying system identification techniques. Methods based on cancellation of process poles like the IMC give simple calculations, but they are not uniformly applicable. Methods like the dominant pole design or the frequency response methods are better in this respect, but they are also more demanding computationally. Since the available computational capacity is expected to increase over the next ten year period, we do not think that this is a major disadvantage.

4.11 References

There is a very large literature on tuning of PID controllers. Good general sources are the books (Smith, 1972), (Deshpande and Ash, 1981), (Shinskey, 1988), (McMillan, 1983), (Corripio, 1990), and (Suda *et al.*, 1992). The books clearly show the need for a variety of techniques, simple tuning rules, as well as more elaborate procedures that are based on process modeling, formulation of specifications, and control design. Even if simple heuristic rules are used, it is important to realize that they are not a substitute for insight and understanding. Successful controller tuning can not be done without knowledge about process modeling and control theory. It is also necessary to be aware that there are many different types of control problems and consequently many different design methods. To only use one method is as dangerous as to only believe in empirical tuning rules.

Control problems can be specified in many different ways. A good review of different ways to specify requirements on a control system is given in (Truxal, 1955), (Maciejowski, 1989), and (Boyd and Barratt, 1991). To formulate specifications it is necessary to be aware of the factors that fundamentally limit the performance of a control system. Simple ways to asses the achievable performance of controllers are given in (Shinskey, 1990), (Shinskey, 1991a), and (Åström, 1991). There are many papers on comparisons of control algorithms and tuning methods. The paper (McMillan, 1986) gives much sound advice; other useful papers are (Miller *et al.*, 1967) and (Gerry, 1987).

The paper (Ziegler and Nichols, 1942) is the classic work on controller tuning. An interesting perspective on this paper is given in an interview with Ziegler, see (Blickley, 1990). The CHR-method, described in (Chien *et al.*, 1952), is a modification of the Ziegler-Nichols method. This is one of the first papers where it is mentioned that different tuning methods are required for setpoint response and for load disturbance response. Good response to load disturbances is often the relevant criteria in process control applications. In spite of this, most papers concentrate on the setpoint response. Notice also

that the responses can be tuned independently by having a controller that admits a two-degree-of-freedom structure. The usefulness of a design parameter is also mentioned in the CHR-paper. In spite of its shortcomings the Ziegler-Nichols method has been the foundation for many tuning methods, see (Tan and Weber, 1985), (Mantz and Tacconi, 1989), and (Hang *et al.*, 1991). Tuning charts were presented in (Wills, 1962b), (Wills, 1962a), and (Fox, 1979).

The loop-shaping methods were inspired by classical control design methods based on frequency response, see (Truxal, 1955). Applications to PID control are found in (Pessen, 1954), (Habel, 1980), (Chen, 1989), (Yuwana and Seborg, 1982).

The analytical tuning method was originally proposed in (Newton *et al.*, 1957); a more recent presentation is found in (Boyd and Barratt, 1991). The original papers on the λ-tuning method are (Dahlin, 1968) and (Higham, 1968). The method is sometimes called the Dahlin method, see (Deshpande and Ash, 1981). This method is closely related to the Smith predictor and the internal model controller, see (Smith, 1957), (Chien, 1988), (Chien and Fruehauf, 1990), and (Rivera *et al.*, 1986). The PPI controller, which is described in (Hägglund, 1992), and the method given in (Haalman, 1965) are special cases. The tuning technique developed in (Smith and Murrill, 1966), (Pemberton, 1972a), (Pemberton, 1972b), (Smith *et al.*, 1975), (Hwang and Chang, 1987) are also based on the analytical approach.

The analytical tuning method gives controllers that cancel poles and zeros in the transfer function of the process. This leads to lack of observability or controllability. There are severe drawbacks in this as has been pointed out many times, e.g., in (Shinskey, 1991b) and (Morari and Lee, 1991). The response to load disturbances will be very sluggish for processes with dominating, long time constants.

Many methods for control design are based on optimization techniques. This approach has the advantage that it captures many different aspects of the design problem. There is also powerful software that can be used. A general discussion of the use of optimization for control design is found in (Boyd and Barratt, 1991). The papers (Rovira *et al.*, 1969) and (Lopez *et al.*, 1969) give controllers optimized with respect to the criteria ISE, IAE and ITAE. Other applications to PID control are given in (Hazebroek and van der Waerden, 1950), (Wolfe, 1951), (Oldenburg and Sartorius, 1954), (van der Grinten, 1963a), (Lopez *et al.*, 1967), (Marsili-Libelli, 1981), (Yuwana and Seborg, 1982), (Patwardhan *et al.*, 1987), (Wong and Seborg, 1988), (Polonoyi, 1989), and (Zhuang and Atherton, 1991). The methods BO and SO were introduced in (Kessler, 1958a) and (Kessler, 1958b). A discussion of these methods with many examples are found in (Fröhr, 1967) and (Fröhr and Orttenburger, 1982).

Pole placement is a straightforward design method much used

in control engineering, see (Truxal, 1955). It has the advantage that the closed-loop poles are specified directly. Many other design methods can also be interpreted as pole placement. The papers (Elgerd and Stephens, 1959) and (Graham and Lathrop, 1953) show how many properties of the closed-loop system can be deduced from the closed-loop poles. This gives good guidance for choosing the suitable closed-loop poles. An early example of pole placement is (Cohen and Coon, 1953), (Coon, 1956a), and (Coon, 1956b). It may be difficult to choose desired closed-loop poles for high-order systems. This is avoided by specifying only a few poles, as in the dominant pole design method described in (Persson, 1992), (Persson and Åström, 1992), and (Persson and Åström, 1993).

There are comparatively few papers on PID controllers that consider the random nature of disturbances. The papers (van der Grinten, 1963b), (Goff, 1966a), and (Fertik, 1975) are exceptions.

New Tuning Methods

5.1 Introduction

Many methods for designing PID controllers were presented in Chapter 4. From this we can conclude that there are many issues that have to be taken into account when designing a controller, e.g., load disturbance response, measurement noise, setpoint following, model requirements, and model uncertainty. It is also clear that there is a need for a variety of tuning methods; simple techniques that require little process knowledge as well as more elaborate methods that use more information about the process.

In this chapter we use the insight obtained in Chapter 4 to develop new methods for controller tuning. In Section 5.2 we discuss the key requirements on a good design method. In addition to the issues mentioned above, we consider the choice of design parameters and the process knowledge required. The method used to develop the new rules is quite straightforward. First we apply a reliable design method with the desired characteristics to a large test batch of representative processes. Then we try to correlate the controller parameters obtained with simple features that characterize the process dynamics.

In Sections 5.3 and 5.4 we present tuning rules that can be viewed as extensions of the Ziegler-Nichols rules. The main difference is that we are using three parameters to characterize process dynamics instead of two parameters used by Ziegler and Nichols. It is shown that the new methods give substantial improvements in control performance while retaining much of the simplicity of the Ziegler-Nichols rules.

Methods based on the step response of the process are presented in Section 5.3. In this case we characterize process dynamics with the parameters a and L used by Ziegler and Nichols and, in addition, the normalized dead time τ. These parameters are easily determined. The tuning rules obtained give the normal PID parameters and, in addition, the setpoint weighting.

In Section 5.4 we present frequency-domain methods. They are based on the parameters ultimate gain K_u, ultimate period T_u and

gain ratio κ. These parameters can be obtained from the conventional Ziegler-Nichols experiment or an experiment with relay tuning combined with a determination of the static gain of the process.

The methods used in Sections 5.3 and 5.4 are based on approximate process models. In Section 5.5 we present an efficient method of computing controller gains when the transfer function of the process in known. These results make it possible to judge the advantage in obtaining more process information.

In Section 5.6 we explore some consequences of the results of the previous sections. The closed-loop systems obtained with the new tuning rules may have many poles and zeros. The behavior of the closed-loop system, however, is dominated by a few poles and zeros. These can be related to the key features of the process. By investigating these relations we get interesting insight into the properties of the closed-loop system, which can be used to judge achievable performance directly from the process features.

Examples of using the new tuning rules are given in Section 5.7.

5.2 A Spectrum of Tools

The results of Chapter 4 clearly indicate that a sound tuning method should consider many different issues such as load disturbances, measurement noise, model requirements, and model accuracy. A good tuning method should also have design parameters so that the desired performance can be changed easily. Unfortunately, it is not possible to find a single tuning method that satisfies all requirements. Instead, we develop a spectrum of methods that differ in the effort required to use them and in the performance obtained.

The wide-spread use of the Ziegler-Nichols methods and its variants clearly indicates the need for simple methods that use minimal process information, but there is also a need for more complex methods that require more effort but, in return, give better control performance.

To develop the simple methods we must first find out if it is at all possible to obtain reliable tuning rules based on a simple characterization of process dynamics. We must also find features that are useful for characterizing the process dynamics. We have approached this in an empirical way by trial and error.

We start with a test batch of processes with known transfer functions. Controllers for these processes are then designed using a good tuning method, e.g., dominant pole design. We then attempt to find process features that admit simple description of the controller parameters. The choice of parameters that are useful to characterize process dynamics is guided by the analysis in Chapter 2. After sev-

eral attempts we find that it is possible to obtain reasonable tuning rules based on three parameters, and that the dimension-free parameters relative dead time τ and relative gain κ are useful.

Since the method is based on the parameters κ and τ, it is called Kappa-Tau tuning, which we also abbreviate as KT tuning. For more accurate tuning, it is suggested to use dominant pole design. This method will, however, require more process knowledge.

The Test Batch

The results of an investigation depend critically on the chosen test batch. We have chosen processes that are representative for the dynamics of typical industrial processes. The following systems were used for stable processes.

$$\begin{aligned} G_1(s) &= \frac{e^{-s}}{(1+sT)^2} && T = 0.1, \ldots, 10 \\ G_2(s) &= \frac{1}{(s+1)^n} && n = 3, 4, 8 \\ G_3(s) &= \frac{1}{(1+s)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)} && \alpha = 0.2, 0.5, 0.7 \\ G_4(s) &= \frac{1-\alpha s}{(s+1)^3} && \alpha = 0.1, 0.2, 0.5, 1, 2 \end{aligned} \tag{5.1}$$

To cover processes with integration we also include models obtained by adding an integrator to the systems listed above.

The test batch (5.1) does not include the transfer function

$$G(s) = K_p \frac{e^{-sL}}{1+sT} \tag{5.2}$$

because this model is not representative for typical industrial processes. Tuning based on the model (5.2) typically give a controller gain that is too high. This is remarkable because tuning rules have traditionally been based on this model.

Simple Tuning Rules

To obtain the simple tuning rules we use the Ziegler-Nichols rules as a starting point. These rules are based on process data in the form of two parameters: a and L for the step-response method, and T_u and K_u for the frequency-response method. The properties of the Ziegler-Nichols rules were discussed extensively in Section 4.3. The results can be summarized as follows.

A. The responses are too oscillatory.
B. Different tuning rules are required for setpoint response and for load disturbance response.
C. The rules give poor results for systems with long normalized dead time.
D. There is no tuning parameter.

Drawback A is easy to deal with. It is sufficient to modify the parameters in the tables. Item B can be dealt with by tuning for load disturbances and using setpoint weighting to obtain the desired setpoint response. Item C is more difficult to deal with because it is necessary to have more process information. A first step is to characterize the process by three parameters instead of two. It turns out that this gives a substantial improvement.

As a tuning parameter we use the maximum sensitivity M_s. Recall that this parameter is defined as

$$M_s = \max_{\omega} \left| \frac{1}{1 + G_p(i\omega)G_c(i\omega)} \right|$$

where $G_c(s)$ is the controller transfer function and $G_p(s)$ the process transfer function. The parameter also has a nice geometrical interpretation in the Nyquist diagram. The shortest distance from the critical point -1 to the Nyquist curve of G_pG_c is $1/M_s$. This admits a direct interpretation as a robustness measure, because it tells how much the process can change without causing instability. Typical values of M_s are in the range of 1.2 to 2. Larger values of M_s give systems that are faster but less robust. It is also useful that the range to consider is not too large.

5.3 Step-Response Methods

In this section we describe simple tuning rules based on step-response data. The need for such techniques are clear in an historical perspective. A simple tuning method of this type is useful for manual tuning and it can also be incorporated into automatic tuners. It turns out that it is possible to obtain substantial improvements compared to other approaches at the cost of a modest increase in complexity.

Stable Processes

First we consider the case when the processes are stable. Process dynamics are characterized by three parameters: the static gain K_p, the apparent lag T, and the apparent dead time L. In Chapter 2 we

discussed different ways to determine these parameters experimentally. In that section we also found that it is difficult to determine more than three parameters from a step response. There is some arbitrariness in the determination of L and T. Here we determine L as the intersection of the tangent with the steepest slope with the time axis. Parameter T is determined as the time when the step response reaches 63% of its final value. An alternative is to determine the average residence time, T_{ar}, by the method of moments and to determine T from

$$T = T_{ar} - L$$

To present the results it is convenient to reparameterize the process. Guided by the Ziegler-Nichols formula we use the parameter

$$a = K_p \frac{L}{T}$$

instead of K_p and the relative dead time

$$\tau = \frac{L}{L+T} = \frac{L}{T_{ar}}$$

instead of T.

Integrating Processes

Parameters a and L can be determined from a step-response experiment for processes with integration. Since the process is not stable it will, however, not reach a steady state. The initial part of the response can be determined, but the experiment has to be interrupted after some time.

The relative dead time τ is zero for processes with integration. It can thus be attempted to use the formulas derived for stable processes with $\tau = 0$. It is, however, useful to base the tuning on more information about the initial part of the step response. This can be obtained from the impulse response of the system, which can be approximated by the pulse response if the pulse is sufficiently short. If the transfer function of the process is $G(s)$, we have

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

where $H(s)$ is a stable transfer function. A step-response experiment performed on $H(s)$ is equivalent to a pulse-response experiment performed on $G(s)$. Thus, we can determine the steady-state gain K_p' and the average residence time T_{ar}' for the transfer function $H(s)$ using the methods discussed in Chapter 2. It can be shown that

$$a = K_p' T_{ar}' = K_p'(L' + T')$$
$$L = T_{ar}' = L' + T'$$

Parameters a and L are therefore easy to obtain from the pulse experiment. The normalized dead time τ', associated with transfer function $H(s)$, can be used as an additional parameter. Tuning rules based on a, L, and τ' can be developed for processes with integration. They are similar to the rules for stable processes.

In the old literature on controller tuning the expression "processes with self-regulation" was used to describe stable processes and "processes without self-regulation" was used to describe processes with integration. It is interesting to observe that these classes always were treated separately in classical papers on controller tuning.

Normalization of Controller Parameters

The PI controller has three parameters: the gain K, the integration time T_i, and the setpoint weighting b. It is convenient to represent these parameters in dimension-free form by suitable normalization. The normalized controller gain is aK, and the normalized integration time T_i/L. This is the same normalization used in the Ziegler-Nichols rules. (Compare with Section 4.3.) In some cases the integration time will be normalized by T instead of L. Notice that parameter T is not defined for processes with integration. This is the reason why it is better to base the tuning on L than on T.

The Method

An empirical method is used to find the new tuning rules. Controller parameters are computed for the different processes in the test batch, using dominant pole design. We then attempt to find relations between the normalized controller parameters and the normalized process parameters. This is done by plotting the normalized controller parameters as a function of normalized dead time τ. For example, we investigate whether the normalized controller gain can be expressed as

$$aK = f(\tau)$$

and analogous expressions for the other parameters. It turns out that it is indeed possible to find approximations of this type. The functions obtained can be well approximated by functions having the form

$$f(\tau) = a_0 e^{a_1\tau + a_2\tau^2} \tag{5.3}$$

Many other functions can also be used.

PI Control for Stable Processes

The simplified tuning rules for PI controllers are treated first. Figure 5.1 shows the normalized controller parameters as a function of

normalized dead time for PI control of stable processes. This figure is based on the systems in the test batch (5.1). The curves drawn correspond to the results obtained by curve fitting. The tuning obtained by the Ziegler-Nichols rules are shown by the dashed lines in the figure.

To get some insight we consider the results for $M_s = 2$, which corresponds to the data labeled $\times$ in the figure. Figure 5.1 shows that the normalized controller gain ranges from 0.35 to 1.5, the normalized integration time from 0.2 to 9, and the setpoint weighting from 0.4 to 0.6. This shows clearly that it is impossible to obtain good tuning rules that do not depend on τ. The deviations from the solid lines in the figure is about $\pm 20\%$. With tuning rules based on three parameters it is thus possible to obtain reasonable tuning rules, at least for the classes of systems given in the test batch (5.1). If the range of τ is restricted to values between 0.2 and 0.6, the gain ratio varies between 0.35 and 0.50, but the normalized integration time varies between 0.6 and 3. The behavior for $M_s = 1.4$ is similar but the ranges of the

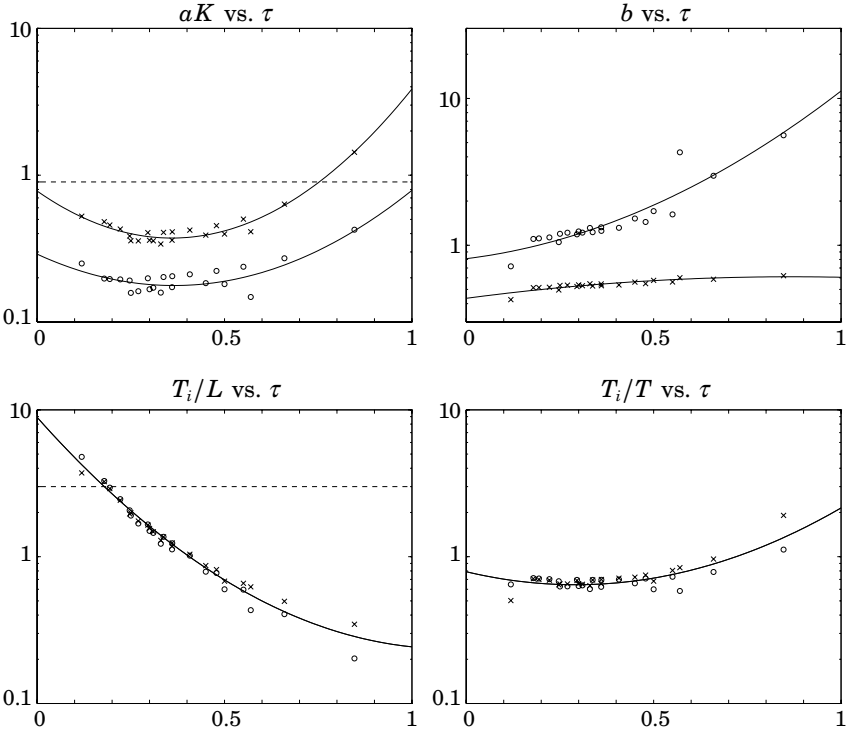

Figure 5.1 Tuning diagrams for PI control of stable processes. Controller parameters are obtained by applying dominant pole design with $M_s = 1.4$, marked with $\circ$, and $M_s = 2$, marked with $\times$ to the systems in the test batch (5.1). The dashed lines correspond to the Ziegler-Nichols tuning rule.

normalized parameters are even larger. It is thus not possible to find a good tuning rule that does not depend on τ even for a restricted range of values. This explains why tuning rules for PI control of the Ziegler-Nichols type have performed so poorly.

The figure also shows that there is a significant difference between the controller gains obtained when the design parameter M_s has the values 1.4 and 2. Notice, however, that the integral time appears to be independent of the design parameter M_s. This means that for PI control of stable processes we can choose integration time independent of M_s and simply use the gain to adjust M_s. The setpoint weighting does not vary much with τ when $M_s = 2$, but it changes significantly when $M_s = 1.4$.

Figure 5.1 illustrates the well-known difficulties with Ziegler Nichols tuning of PI controllers. It indicates that the gain obtained from the Ziegler-Nichols rule should be reduced. It also shows that control of processes with small and large τ is very different. For lag-dominated processes, i.e., small values of τ, the proportional gain and the integral time should be smaller. Another way to express this is that proportional action should be stronger and integral action weaker. For dead-time dominated process we have the reverse situation. Also, recall that in both extremes there are other controllers that will perform better than PI controllers.

Figure 5.1 also suggests that if the integral time is normalized with apparent time constant T instead of apparent dead time L, the Ziegler-Nichols method can be replaced by the following simple tuning rule, $aK = 0.4$, $T_i = 0.7T$, and $b = 0.5$. This gives quite good tuning for relative dead times in the range $0.1 < \tau < 0.7$. This means that the ratio of apparent dead time to apparent time constant is between 0.1 and 2. Table 5.1 gives the coefficients a_0, a_1, and a_2 of functions of the form (5.3) that are fitted to the data in Figure 5.1. The corresponding graphs are shown in solid lines in the figure.

Table 5.1 Tuning formula for PI control obtained by the step-response method. The table gives parameters of functions of the form $f(\tau) = a_0 \exp(a_1\tau + a_2\tau^2)$ for the normalized controller parameters.

	$M_s = 1.4$			$M_s = 2.0$		
	a_0	a_1	a_2	a_0	a_1	a_2
aK	0.29	−2.7	3.7	0.78	−4.1	5.7
T_i/L	8.9	−6.6	3.0	8.9	−6.6	3.0
T_i/T	0.79	−1.4	2.4	0.79	−1.4	2.4
b	0.81	0.73	1.9	0.44	0.78	−0.45

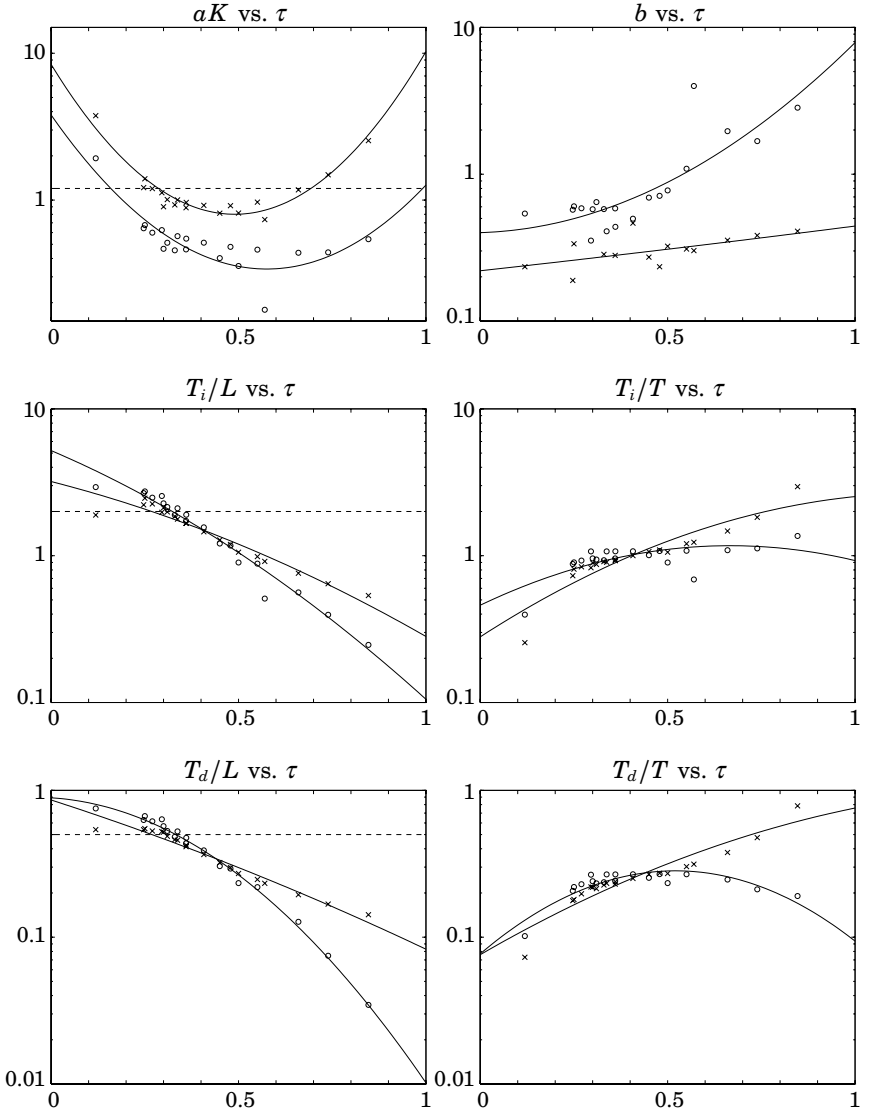

Figure 5.2 Tuning diagrams for PID control. Controller parameters are obtained by applying dominant pole design with $M_s = 1.4$, marked with $\circ$, and $M_s = 2$, marked with $\times$, to the systems in test batch (5.1). The dashed lines correspond to the Ziegler-Nichols tuning rule.

PID Control for Stable Processes

Tuning rules for PID control are developed in the same way as the rules for PI control. Process dynamics are characterized by the parameters a, L, and τ which were also used for PI control. The controller parameters are normalized as aK, T_i/L, and T_d/L. The parameters are determined for the systems in the test batch (5.1) using the dominant pole design method. The controller parameters are then normalized and plotted against normalized dead time τ. The results are given in Figure 5.2, where curves fitted to the data are given in solid lines and the Ziegler-Nichols tuning is shown in dashed lines.

Consider the values obtained for $M_s = 2$, i.e., the points marked $\times$ in Figure 5.2. The normalized gain varies from 0.8 to 3, the normalized integration time from 0.55 to 2.5, the normalized derivation time from 0.15 to 0.55, and the setpoint weighting from 0.2 to 0.4. Notice that the ranges of parameters are significantly smaller than for PI control. This explains why it is easier to find tuning rules that do not depend on τ for PID than for PI controllers. With tuning rules based on three parameters, like those illustrated by full lines in Figure 5.2, it is possible to obtain controller parameters with a precision of about 25% for the processes in the test batch.

The Ziegler-Nichols tuning rules are represented by dashed lines in the figure. There is reasonable agreement with the Ziegler-Nichols rules when τ is between 0.2 and 0.4. This is in sharp contrast with Figure 5.1, where no agreement with the Ziegler-Nichols rule was obtained for any τ. This corresponds well with the observation that the Ziegler-Nichols step-response rules work better for PID than for PI control. The general pattern indicated in Figure 5.2 is that both integration time T_i and derivative time T_d should be decreased with increasing τ. The value of the normalized gain depends on M_s. It is slightly less than the value obtained with the Ziegler-Nichols rule for $M_s = 2$. Both the integral time and the derivative time, however, should depend on τ. The figure also shows that proportional action dominates for small τ and integral action for large τ. In Table 5.2 we give the results of curve fitting for Figure 5.2. Figure 5.2 shows that constant gain can be used for values of τ between 0.2 and 0.7, if the normalization is performed with T instead of L.

Integrating Processes

Tuning rules for processes with integration were developed by applying the dominant pole design method to processes with the transfer functions

$$G_i(s) = \frac{1}{s} H_i(s)$$

Table 5.2 Tuning formula for PID control obtained by the step-response method. The table gives parameters of functions of the form $f(\tau) = a_0 \exp(a_1\tau + a_2\tau^2)$ for the normalized controller parameters.

	$M_s = 1.4$			$M_s = 2.0$		
	a_0	a_1	a_2	a_0	a_1	a_2
aK	3.8	−8.4	7.3	8.4	−9.6	9.8
T_i/L	5.2	−2.5	−1.4	3.2	−1.5	−0.93
T_i/T	0.46	2.8	−2.1	0.28	3.8	−1.6
T_d/L	0.89	−0.37	−4.1	0.86	−1.9	−0.44
T_d/T	0.077	5.0	−4.8	0.076	3.4	−1.1
b	0.40	0.18	2.8	0.22	0.65	0.051

where $H_i(s)$ are the transfer functions given in the test batch (5.1). The apparent dead time L', the apparent time constant T', and the gain K_p' of the transfer functions $H_i(s)$ were determined. The normalized controller parameters aK, T_i/L, T_d/L, and b have then been plotted as functions of the normalized dead time for $G_i(s)$.

PI Control for Integrating Processes

Figure 5.3 gives the results for PI control. The figure shows that there is some variation in the normalized parameters aK and T_i/L with τ'. For $M_s = 2$, the values of aK varies between 0.5 and 0.7, and the values of T_i/L are between 3 and 5. The parameter b changes more, from 0.3 to 0.7. The variation is larger for $M_s = 1.4$. The parameter values given by the Ziegler-Nichols rule are shown by dashed lines in Figure 5.3. Our rules give controller gains that are about 30 % lower and integration times that are about 50% longer than the values obtained by the Ziegler-Nichols method. The functions fitted to the data in Figure 5.3 are given in Table 5.3.

For PI control it appears that reasonable tuning can be based on formulas for $\tau = 0$, and that there will be a modest improvement from knowledge of τ'. A comparison with Figure 5.1 shows that the curve for T_i/L has to be modified a little for small values of τ.

PID Control for Integrating Processes

Figure 5.4 shows the results obtained for PID control of processes with integration. Notice that the normalized controller parameters vary significantly with τ' in this case. For the case $M_s = 2$ parameter aK varies between 0.8 and 5, parameter T_i/L is in the range of 1 to

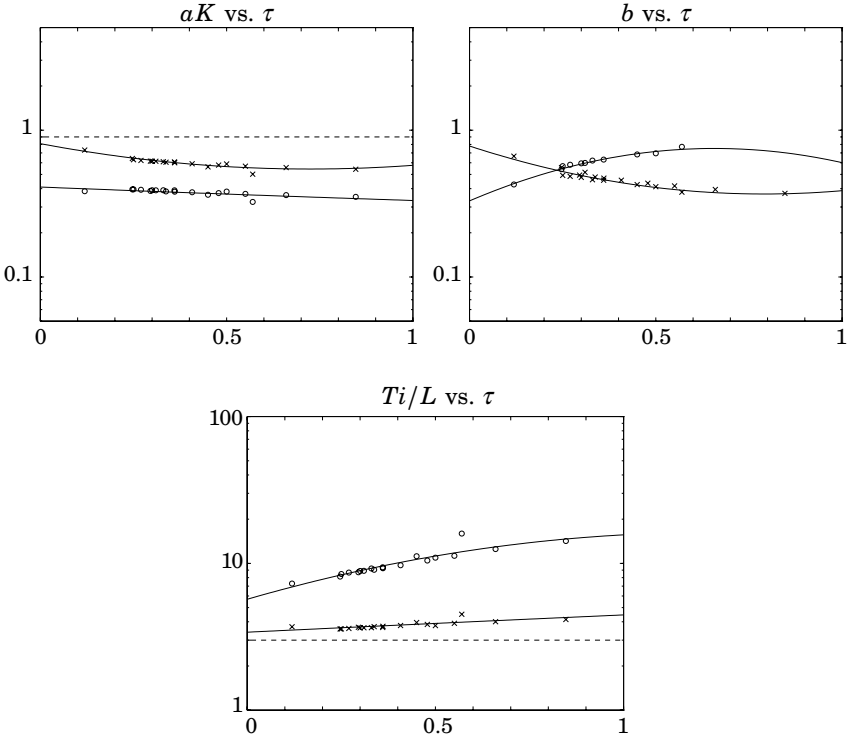

Figure 5.3 Tuning diagrams for PI control for processes with integration. Controller parameters are obtained by applying dominant pole design with $M_s = 1.4$, marked with ∘, and $M_s = 2$, marked with × to the systems in the test batch (5.1) complemented with an integrator. The dashed lines correspond to the Ziegler-Nichols tuning rule.

Table 5.3 Tuning formula for PI control obtained by the step-response method for processes with integration. The table gives parameters of functions of the form $f(\tau) = a_0 \exp(a_1\tau + a_2\tau^2)$ for the normalized controller parameters.

	$M_s = 1.4$			$M_s = 2.0$		
	a_0	a_1	a_2	a_0	a_1	a_2
aK	0.41	−0.23	0.019	0.81	−1.1	0.76
T_i/L	5.7	1.7	−0.69	3.4	0.28	−0.0089
b	0.33	2.5	−1.9	0.78	−1.9	1.2

3.5, and T_d/L is in the range of 0.25 to 0.4. There are even larger variations for $M_s = 1.4$. To obtain good PID control of processes with integration it is therefore essential to know τ'.

The controller parameters obtained by the Ziegler-Nichols rule are shown with dashed lines in the figure. It is interesting to observe that the gain given by our rules is larger and the integration time is smaller for small τ'. The parameters obtained when fitting functions of the form (5.3) to the data are given in Table 5.4. The ratio T_i/T_d varies significantly with parameter τ'. For $M_s = 2$ it increases from 2.5 for $\tau' = 0$ to 12 for $\tau' = 1$. The variations in the ratio is even larger for designs with $M_s = 1.4$.

Summary

The results show that for PI control we can obtain tuning formulas based on τ that can be applied also to processes with integration. The formulas are obtained simply by extending the formulas for stable processes to $\tau = 0$. A small improvement can be obtained for small values of τ by also determining parameter τ'.

For PID control it is necessary to have separate tuning formulas for stable processes and processes with integration. For processes with integration good tuning cannot be obtained by tuning formulas that are only based on parameter τ. It is necessary to provide additional information, e.g., by providing the parameter τ'. The tuning formulas derived for processes with integration can also be applied when τ is small, say $\tau < 0.2$, which corresponds to $T > 4L$.

5.4 Frequency-Response Methods

In this section, the tuning rules based on frequency-domain methods are developed. In the tradition of Ziegler and Nichols we characterize the process by the ultimate gain K_u, the ultimate period T_u, and the gain ratio $\kappa = 1/K_pK_u$. (Compare with Chapter 2.) The controller parameters are normalized as K/K_u, T_i/T_u and T_d/T_u. The tuning rules are obtained in the same way as for the step-response method. Controllers for the different processes are designed using the dominant pole design with two values of the design parameter, $M_s = 1.4$ and $M_s = 2$. It has then been attempted to find relations between the normalized controller parameters and the normalized process parameters. The results can be conveniently represented as graphs where normalized controller parameters are given as functions of the gain ratio κ.

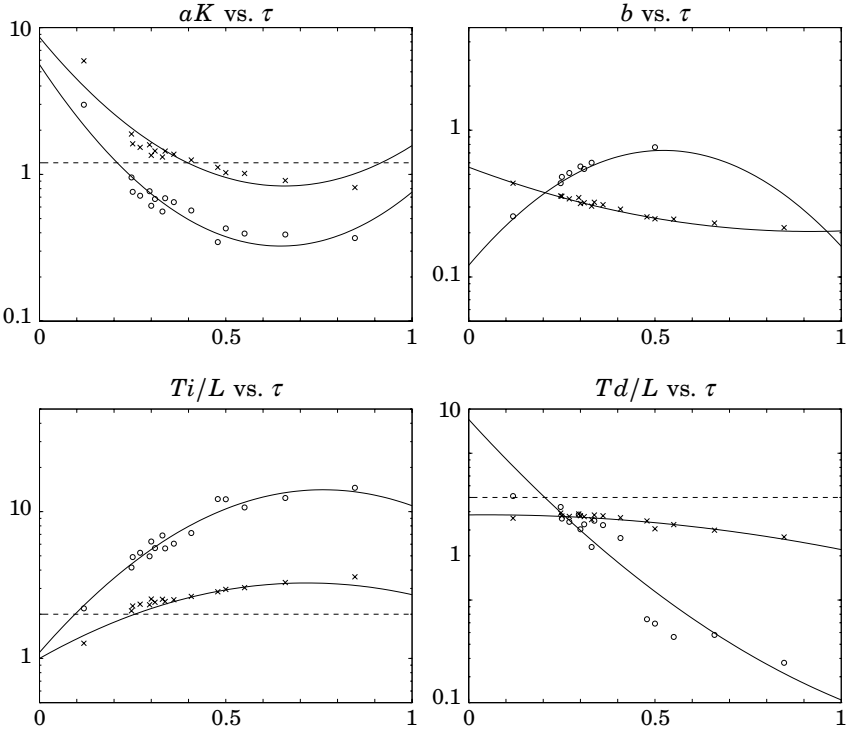

Figure 5.4 Tuning diagrams for PID control of processes with integration. Controller parameters are obtained by applying dominant pole design with $M_s = 1.4$, marked with $\circ$, and $M_s = 2$, marked with $\times$ to the systems in the test batch (5.1) complemented with an integrator. The dashed lines correspond to the Ziegler-Nichols tuning rule.

Table 5.4 Tuning formula for PID control based on the step-response method for processes with integration. The table gives parameters of functions of the form $f(\tau) = a_0 \exp(a_1\tau + a_2\tau^2)$ for the normalized controller parameters.

	$M_s = 1.4$			$M_s = 2.0$		
	a_0	a_1	a_2	a_0	a_1	a_2
aK	5.6	−8.8	6.8	8.6	−7.1	5.4
T_i/L	1.1	6.7	−4.4	1.0	3.3	−2.3
T_d/L	1.7	−6.4	2.0	0.38	0.056	−0.60
b	0.12	6.9	−6.6	0.56	−2.2	1.2

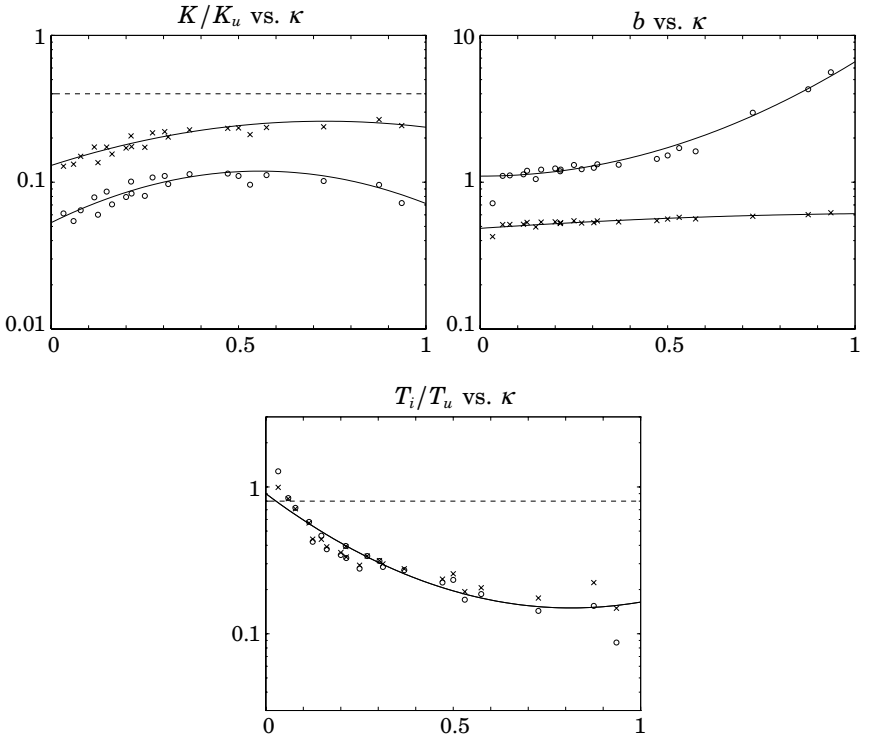

Figure 5.5 Tuning diagrams for PI control based on K_u, T_u, and κ. Controller parameters are obtained by applying dominant pole design with $M_s = 1.4$, marked with $\circ$, and $M_s = 2$, marked with $\times$, to the systems in test batch (5.1). The dashed lines correspond to the Ziegler-Nichols tuning rule.

Table 5.5 Tuning formula for PI control based on the frequency-response method. The table gives parameters of functions of the form $f(\kappa) = a_0 \exp(a_1\kappa + a_2\kappa^2)$ for the normalized controller parameters.

	$M_s = 1.4$			$M_s = 2.0$		
	a_0	a_1	a_2	a_0	a_1	a_2
K/K_u	0.053	2.9	−2.6	0.13	1.9	−1.3
Ti/Tu	0.90	−4.4	2.7	0.90	−4.4	2.7
b	1.1	−0.0061	1.8	0.48	0.40	−0.17

PI Control for Stable Processes

Figure 5.5 shows the normalized parameters of a PI controller as a function of κ. Consider the case of $M_s = 2$, i.e., the points marked $\times$ in Figure 5.5. The normalized gain ranges from 0.15 to 0.3, the normalized integration time from 0.15 to 1, and the setpoint weighting from 0.4 to 0.6. The ranges of variation are smaller than for the step-response method (compare with Figure 5.1). The variations are, however, too large to admit tuning rules that do not depend on κ. If we choose tuning rules that do depend on κ, e.g., those shown in straight lines in Figure 5.5, we can find rules that give controller parameters within $\pm 20\%$ for the systems in the test batch (5.1). If we are satisfied with less precision it suffices to let the integration time depend on κ. The behavior for $M_s = 1.4$ is similar, but the setpoint weighting changes over a wider range.

The normalized controller gain and the setpoint weighting depend on the design parameter M_s, but the same value of the integral time can be used for all M_s. This is similar to what we found for the step-response method. A comparison with Figure 5.1 shows that there is slightly less scatter with the step-response method than with the frequency-domain method.

The variation of the integration time with κ is particularly noticeable in Figure 5.5. This reflects the situation that the proportional action is larger than integral action for processes that are lag dominant. The reverse situation occurs for processes where the dynamics are dominated by dead time.

The Figure 5.5 also shows why Ziegler-Nichols tuning is not very good in this case. The controller gain is too high for all values of gain ratio κ, and the integral time is too short except for very small values of κ. This agrees well with the observation that the Ziegler-Nichols rules for PI control do not work well.

Table 5.5 gives the coefficients of functions of the form (5.3) fitted to the data in Figure 5.5. The corresponding graphs are shown in solid lines in the figure.

PID Control for Stable Processes

Figure 5.6 shows the normalized parameters of a PID controller as a function of κ. Consider the situation for $M_s = 2$. The normalized gain varies from 0.45 to 0.9, the normalized integral time from 0.2 to 0.55, the derivative time from 0.06 to 0.15, and the setpoint weighting from 0.2 to 0.4. Notice that the ranges are significantly smaller than for PI control. The situation is similar for $M_s = 1.4$, with the exception that the setpoint weighting varies over a larger range in this case. Thus, it is easier, in this case, to find tuning rules that only depend

on two parameters. With tuning rules based on three parameters it is, however, possible to find tuning rules that give an accuracy of 25%, at least for the test batch (5.1).

The figure shows that different values of gain K and setpoint weighting b are obtained for $M_s = 1.4$ and $M_s = 2$. The curves for the integral and derivative times do not vary so much with M_s.

A comparison with Figure 5.5 shows that the range of parameter variations with κ are much less for PID control than for PI control. This supports the well-known observation that rules of the Ziegler-Nichols type work better for PID than for PI control. Figure 5.6 also shows that the normalized gain obtained with $M_s = 2$ is quite close to the gain given by the Ziegler-Nichols rule, whereas the integral and derivative times are smaller for most values of κ.

Table 5.6 gives the parameters a_0, a_1, and a_2 of functions of the form (5.3) fitted to the data in Figure 5.6.

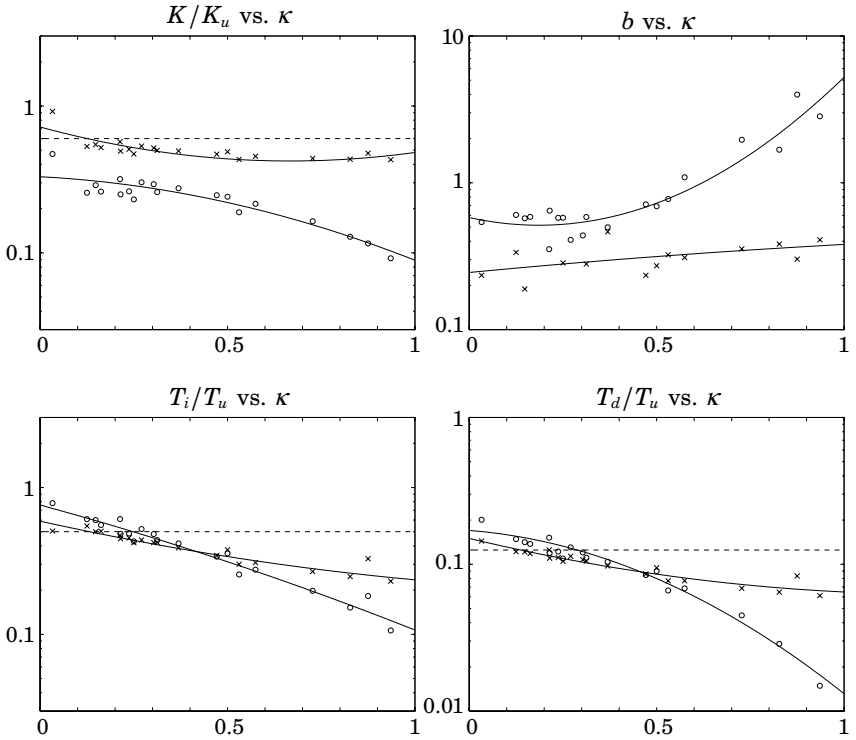

Figure 5.6 Tuning diagrams for PID control based on K_u, T_u and κ. Controller parameters are obtained by applying dominant pole design with $M_s = 1.4$, marked with ∘, and $M_s = 2$ marked with ×, to the systems in test batch (5.1). The dashed lines correspond to the Ziegler-Nichols tuning rule.

Table 5.6 Tuning formula for PID control based on the frequency-response method. The table gives parameters of functions of the form $f(\kappa) = a_0 \exp(a_1\kappa + a_2\kappa^2)$ for the normalized controller parameters.

	$M_s = 1.4$			$M_s = 2.0$		
	a_0	a_1	a_2	a_0	a_1	a_2
K/K_u	0.33	−0.31	−1.0	0.72	−1.6	1.2
T_i/T_u	0.76	−1.6	−0.36	0.59	−1.3	0.38
T_d/T_u	0.17	−0.46	−2.1	0.15	−1.4	0.56
b	0.58	−1.3	3.5	0.25	0.56	−0.12

The Relation Between T_i and T_d

In many tuning rules the ratio of T_i and T_d is fixed. In Figure 5.7 we show the ratio of T_i/T_d obtained with the new tuning rules. The figure shows that the design for $M_s = 2$ gives ratios that are close to 4. This is the same ratio as in the Ziegler-Nichols method. For $M_s = 1.4$, the ratio is close to 4 for $\kappa < 0.6$. For higher values of κ the ratio becomes larger.

Processes with Integration

For integrating processes, the ultimate gain and the ultimate period can be determined as described in Section 2.6, but static gain K_p is not defined. Processes with integration have $\kappa = 0$. For PI control it is possible to extrapolate the previous formulas to $\kappa = 0$ but for PID control it is necessary to have additional information. One possibility is to provide information about other points on the Nyquist curve of the process. A good choice is ω_{90}, i.e., the frequency where the phase

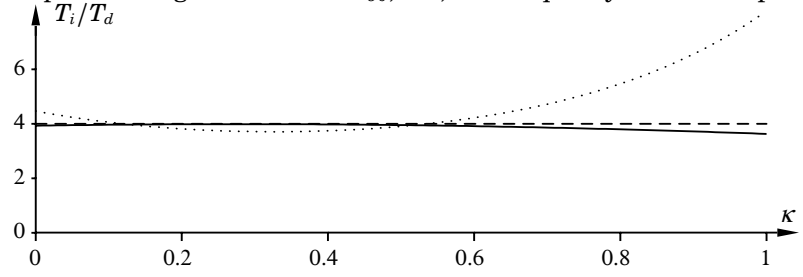

Figure 5.7 Ratio of T_i to T_d obtained with the new tuning rule based on frequency response. The full line corresponds to $M_s = 2$ and the dotted line to $M_s = 1.4$. The dashed line shows the ratio given by the Ziegler-Nichols methods $T_i/T_d = 4$.

lag is 90°.

5.5 Complete Process Knowledge

The methods presented in Sections 5.3 and 5.4 are approximate methods based on partial knowledge of the process. These methods are sufficient in many cases. There are, however, situations where more accuracy is required. This can be achieved either by on-line refinement or by using a more accurate model. There are many empirical rules for on-line tuning that can be refined further by selecting different rule sets depending on the values of τ or κ.

A more accurate model can be obtained by using system identification. This will typically give a model in the form of a pulse transfer function. This model can be transformed into an ordinary transfer function in several different ways. Since the tuning methods are based on the transfer function of the process, it is attractive to use frequency response techniques. The multifrequency method is a technique where a signal that is a sum of sinusoids is chosen as the input. The phases of the sinusoids are chosen so that the amplitude of the signal is minimized. The frequencies chosen can be based on the knowledge of the ultimate frequency ω_u. In this way it is possible to obtain the value of the transfer function for several frequencies in one test. A transfer function can then be fitted to the data, and the dominant pole design technique can then be used. In this section we present alternative methods of performing the computations required for dominant pole design. An alternative to this is to determine a pulse transfer function by applying the system identification methods discussed in Section 2.7 and to develop a design procedure that is based on the pulse transfer function. Such a method can be obtained using ideas similar to those discussed in this chapter.

PI Control

Consider a process with the transfer function $G(s)$. Let the PI controller be parameterized as

$$G_c(s) = k + \frac{k_i}{s} \tag{5.4}$$

In this particular case the design problem can be formulated as follows. Find the parameters k and k_i such that k_i is as large as possible and so that the robustness constraint

$$|1 + G_\ell(i\omega)| = a(k, k_i, \omega) \geq m_0 \tag{5.5}$$

where $G_\ell = G(s)G_c(s)$ is the loop transfer function. The problem is a constrained optimization problem that, unfortunately, is not convex. To solve it we use an iterative method with good initial conditions.

The idea of the algorithm is to evaluate the function

$$m(k, k_i) = \min_{\omega \in \Omega} a(k, k_i, \omega) = m_0 \tag{5.6}$$

for several values of the controller parameters and then to determine the value of k, which maximizes k_i subject to the constraint.

Determination of the function m requires minimization with respect to ω. This is done by a simple search over the interval $\Omega = [\omega_1, \omega_2]$.

The function m can be locally approximated by

$$m(k, k_i) = a + b_0 k_i + b_1 k + \frac{1}{2}\left(c_0 k_i^2 + 2c_1 k k_i + c_2 k^2\right) \tag{5.7}$$

Maximizing k_i with respect to k subject to the constraint (5.6) gives

$$b_1 + c_1 k_i + c_2 k = 0 \tag{5.8}$$

This gives the following relation between k and k_i:

$$k = -\frac{b_1 + c_1 k_i}{c_2} \tag{5.9}$$

Inserting this into Equation (5.7) and using the condition $m(k, k_i) = m_0$ gives

$$A_0 k_i^2 + 2A_1 k_i + A_2 = 0 \tag{5.10}$$

where

$$A_0 = c_0 - \frac{c_1^2}{c_2}$$

$$A_1 = b_0 - \frac{b_1 c_1}{c_2}$$

$$A_2 = 2(a_0 - m_0) - \frac{b_1^2}{c_2}$$

Solving Equation (5.10) gives

$$k_i = \frac{-A_1 \pm \sqrt{A_1^2 - A_0 A_2}}{A_0} \tag{5.11}$$

Having obtained k_i the value of k is then given by Equation (5.9).

Parameters for PID controllers can be determined in a similar way.

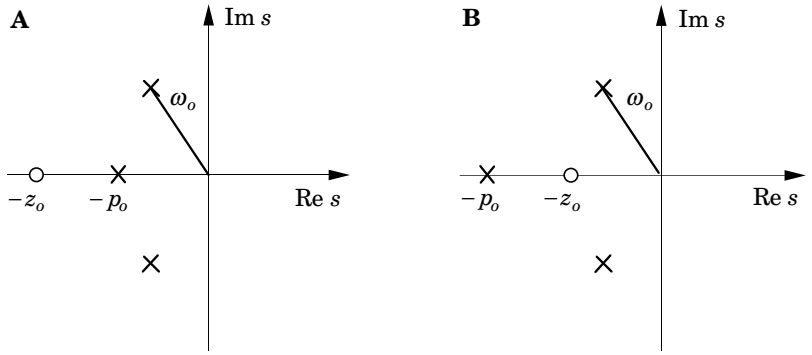

Figure 5.8 Configuration of dominant poles and zeros for a system with PI control. Case A corresponds to systems where the open-loop poles are clustered and case B is the case when the open-loop poles are widely spread.

5.6 Assessment of Performance

In this section we explore some properties of the closed-loop systems obtained with the design methods discussed in the previous sections. The closed-loop systems obtained with PID control typically have many poles and zeros. The behavior of the system is, however, characterized by only a small number of poles. (Compare with Figure 4.4.) The key idea with the dominant pole design procedure was actually to position a few of the dominant poles. In this section we explore the dominant poles further. In particular we investigate how they are related to features of the open loop system and the design parameters.

A preliminary assessment of the nature of the control problem can be made based on the value of the normalized dead time or the gain ratio. For small values ($\tau < 0.1$ or $\kappa < 0.06$) it is often possible to obtain improved control by more complex strategies than PID control. Similarly when the values are close to one ($\tau > 0.7$ or $\kappa > 0.7$) consider using dead-time compensation.

PI Control

Many systems with PI control can be characterized by the closed-loop poles that are closest to the origin in the complex plane. It is often sufficient to consider only three closed-loop poles. In a typical case there are two complex and one real pole (see Figure 5.8). The responses of a system with three poles is a linear combination of the

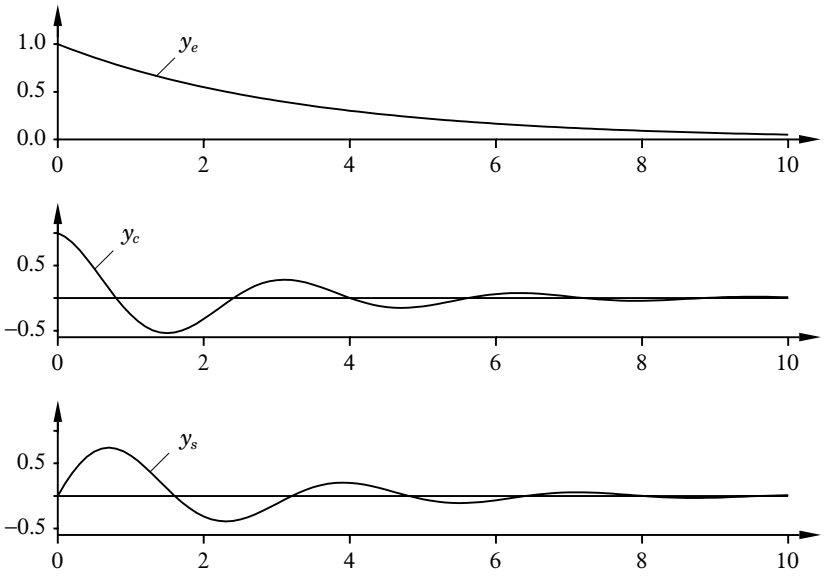

Figure 5.9 Signal modes for a system with one real pole and one complex pole pair. The modes are a damped exponential y_e, a damped cosine y_c, and a damped sine function y_s.

signals

$$y_e = e^{-\alpha_0 t}$$
$$y_s = e^{-\zeta\omega_0 t} \sin \omega_0 \sqrt{1-\zeta^2}$$
$$y_c = e^{-\zeta\omega_0 t} \cos \omega_0 \sqrt{1-\zeta^2}$$

The signal y_e is a decaying exponential, y_s and y_c are exponentially damped sine and cosine functions. The responses may also contain a component due to the excitation, e.g., a constant for the step disturbances. The signals are illustrated in Figure 5.9. The damped sine and cosine waves are often the dominating components and the exponential function corresponds to the creeping behavior found on some occasions.

Since the responses are well approximated by the functions shown in Figure 5.9, it is easy to visualize responses if we know the parameters α_0, ω_0, and ζ, and the amplitudes of the signals. The amplitudes of the different components depend on the parameters and the excitation of the system in a fairly complicated way.

The Real Pole

The real pole at $s = -\alpha_0$ determines the decay rate of the exponential function. The time constant is $T_0 = 1/\alpha_0$, where α_0 is approximately

equal to $1/T_i$. This explains the sluggish response obtained when T_i is too large, compare with Figure 5.9.

For processes, where the open-loop poles are clustered together, the configuration of the poles is as shown in Figure 5.8A. The pole located in $-\alpha_0$ is thus to the right of the zero $-z_0 = -1/T_i$. For systems where the open-loop poles are widely separated, the pole configuration is as shown in Figure 5.8B, where the real pole is to the left of the zero $-z_0$. The cases can approximately be separated through the inequality

$$T_0 > 2\left(L + \sum_{k=1}^{n} T_k\right) = 2(T_{ar} - T_0) \tag{5.12}$$

where T_0 is the time constant associated with the slowest pole (α_0), T_k are the time constants associated with the remaining poles, and T_{ar} is the average residence time (see Section 2.4). The inequality is obtained by analyzing the root locus of the system.

Complex Poles

The damped sine and cosine modes are determined by parameters ω_0 and ζ. The period of the oscillation is

$$T_p = \frac{2\pi}{\omega_0\sqrt{1-\zeta^2}}$$

and the ratio of two successive peaks are

$$d = e^{-2\pi\zeta/\sqrt{1-\zeta^2}}$$

(Compare with Section 4.2.) Knowing parameters ω_0 and ζ, it is thus easy to visualize the shape of the mode.

If the parameters of the controller are determined by the dominant pole design, we can determine how ω_0, ζ, and α_0 depend on the system and the specifications. In this way it is possible to relate the properties of the closed-loop system directly to the features of the process.

To find good relations we use the normalized quantities $\omega_0 L$ and $\alpha_0 T_i$. The relative damping is dimension free by itself. Figure 5.10 shows the relations obtained for the test batch (5.1). For $M_s = 2$ the quantity $\omega_0 L$ ranges from 0.5 to 1.5; ζ ranges from 0.3 to 0.6, and $\alpha_0 T_i$ from 0.8 to 1.2. The value of $\omega_0 L$ is less than one for small τ or κ and larger than one for large values. The variation with τ or κ is larger for $M_s = 1.4$ than for $M_s = 2$. For $M_s = 1.4$ the relative damping depends strongly on τ or κ; it is larger than one for large values of τ or κ. This means that the closed-loop system has three real poles. The value of $\alpha_0 T_i$ is smaller than one in most cases. This

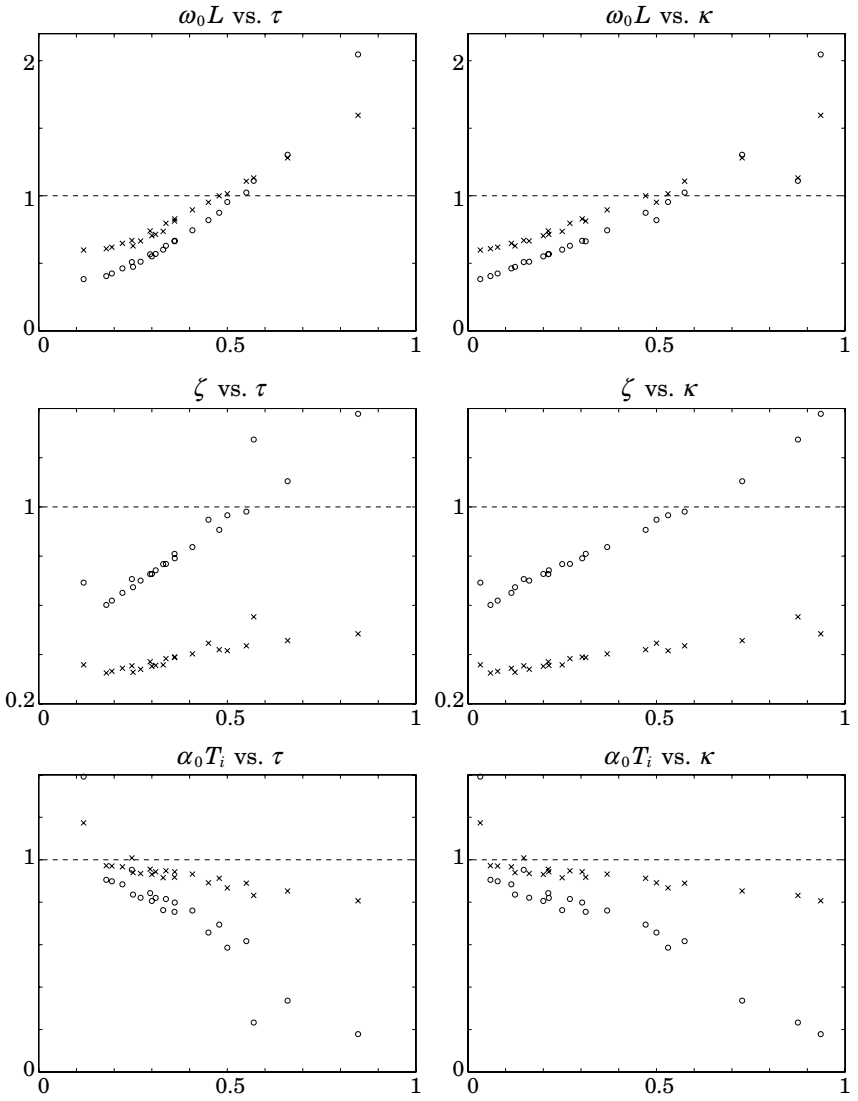

Figure 5.10 Dependence of ω_0, ζ, and α_0 on the system characteristics and the specifications M_s on closed-loop sensitivity. Points marked with *o* correspond to $M_s = 1.4$, and points marked with $\times$ correspond to $M_s = 2.0$.

indicates that the pole-zero configuration shown in Figure 5.8A is the most common case. Figure 5.10 also shows that the quantity $\alpha_0 T_i$ is close to one independently of τ or κ for $M_s = 2$, but that the value varies significantly with τ or κ for $M_s = 1.4$.

By using the relations in Figure 5.10, we can reach a reasonable estimate of the properties of the closed-loop systems obtained by the

Table 5.7 Characterization of the closed-loop poles obtained in Example 5.1.

α	τ	κ	M_s	ω_0	ζ	α_0	z_0	$\omega_0 L$	$\alpha_0 T_i$
0.2	0.12	0.033	1.4	2.5	0.70	1.9	1.4	0.38	1.4
0.5	0.25	0.15	1.4	1.0	0.70	0.93	0.97	0.51	0.96
0.7	0.31	0.21	1.4	0.69	0.74	0.67	0.68	0.57	0.81
0.2	0.12	0.033	2.0	4.0	0.36	2.1	1.8	0.60	1.2
0.5	0.25	0.15	2.0	1.3	0.35	1.0	1.02	0.67	1.0
0.7	0.31	0.21	2.0	0.86	0.36	0.77	0.81	0.71	0.94

design procedures directly from the process characteristics. This is illustrated by a simple example.

EXAMPLE 5.1

Consider a system with the transfer function.

$$G(s) = \frac{1}{(s+1)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}$$

In this case it is easy to vary the spread of the process poles by varying parameter α. The process has one dominating pole with time constant $T_0 = 1$. The average residence time is

$$T_{ar} = 1 + \alpha + \alpha^2 + \alpha^3$$

Thus, the inequality (5.12) gives the value where the poles are considered as clustered to $a = 0.3425$. In Table 5.7 we summarize some parameters for the system. The table shows that the approximate estimates from Figure 5.10 give reasonably good estimates in this case. □

Systems with PID control can be analyzed in a similar manner. In this case it is necessary to consider more closed-loop poles. There are also two zeros corresponding to $1/T_i'$ and $1/T_d'$, where T_i' and T_d' are the integral and derivative time constants for the series representation of the PID controller (compare with Section 3.4).

5.7 Examples

To illustrate the effectiveness of the tuning methods we apply them in a few examples.

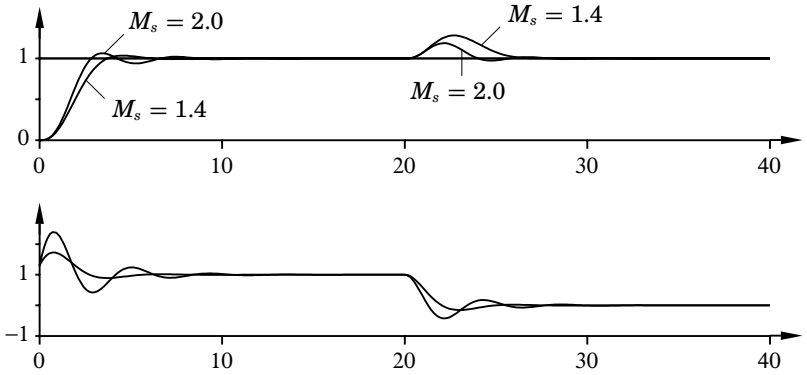

Figure 5.11 Setpoint and load-disturbance response of a process with transfer function $1/(s+1)^3$ controlled by a PID controller tuned with the new simple tuning rules with $M_s = 1.4$ and 2.0. The upper diagram shows setpoint $y_{sp} = 1$ and process output y, and the lower diagram shows control signal u.

EXAMPLE 5.2 Three equal lags

In Examples 4.1 and 4.2, the Ziegler-Nichols methods were applied to the process model

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

It has ultimate gain $K_u = 8$, ultimate period $T_u = 3.6$, and gain ratio $\kappa = 0.125$. The new frequency-response method gives the following parameters for a PID controller:

	$M_s = 1.4$	$M_s = 2.0$
K	2.5	4.8
T_i	2.2	1.8
T_d	0.56	0.46
b	0.52	0.27

The step-response method gives parameters that differ less than 10% from these values.

Figure 5.11 shows the response of the closed-loop systems to a step change in setpoint followed by a step change in the load. The control is significantly better than in Examples 4.1 and 4.2, where the Ziegler-Nichols methods were used. Compare with Figures 4.7 and 4.8. □

In the next example, the new design methods are applied to a process with a significant dead time.

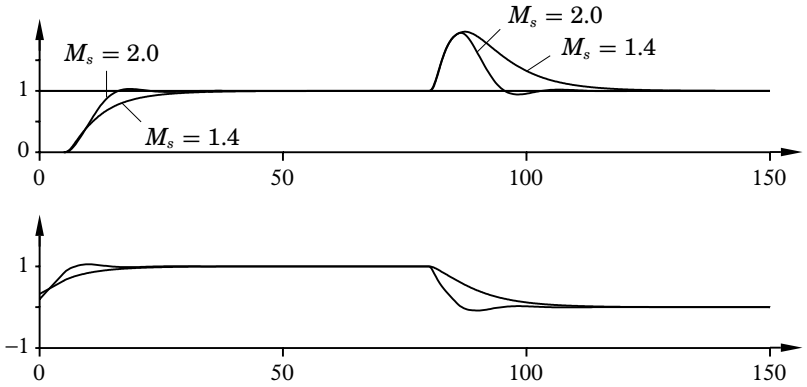

Figure 5.12 Setpoint and load-disturbance response of a process with transfer function $e^{-5s}/(s+1)^3$ controlled by a PID controller tuned with the new simple tuning rules with $M_s = 1.4$ and 2.0. The upper diagram shows setpoint $y_{sp} = 1$ and process output y, and the lower diagram shows control signal u.

EXAMPLE 5.3 Dead-time dominant process

The Ziegler-Nichols methods were applied to the process model

$$G(s) = \frac{e^{-5s}}{(s+1)^3}$$

in Example 4.4, which showed that the Ziegler-Nichols methods are not suitable for processes with large normalized dead time τ and large gain ratio κ. The process has ultimate gain $K_u = 1.25$, ultimate period $T_u = 15.7$, and gain ratio $\kappa = 0.8$. The new frequency-response method gives the following parameters for a PID controller:

	$M_s = 1.4$	$M_s = 2.0$
K	0.17	0.54
T_i	2.6	4.2
T_d	0.48	1.1
b	1.9	0.36

The step-response method gives controller parameters that differ less than 10% from these parameters.

Figure 5.12 shows the response of the closed-loop systems to a step change in setpoint followed by a step change in the load. The control is significantly better than in Example 4.4 (compare with Figure 4.12). The new design method gives a higher gain and shorter integral and derivative times than the Ziegler-Nichols method. Notice that the value of design parameter M_s is crucial in this example. □

The next example illustrates the new design method applied on an integrating process.

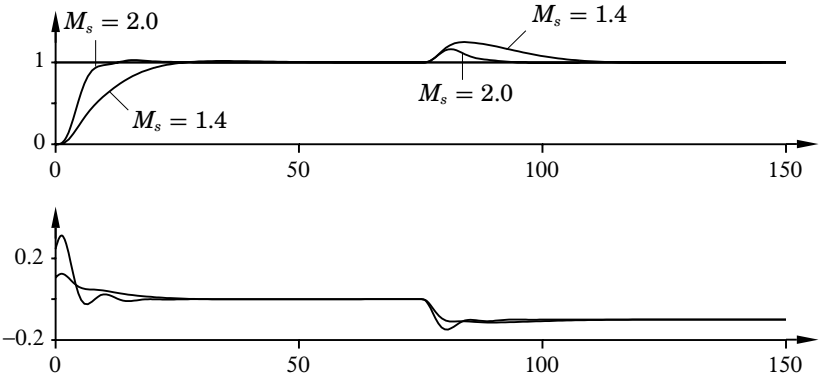

Figure 5.13 Setpoint and load-disturbance response of a process with transfer function $1/s(s+1)^3$ controlled by a PID controller tuned with the new simple tuning rules with $M_s = 1.4$ and 2.0. The upper diagram shows setpoint $y_{sp} = 1$ and process output y, and the lower diagram shows control signal u.

EXAMPLE 5.4 Integrating process

Consider the integrating process

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

The stable process $H(s)$ has static gain $K_p = 1$, apparent dead time $L' = 0.81$, apparent time constant $T' = 3.7$, and relative dead time $\tau' = 0.18$. This gives the following parameters for process $G(s)$:

$$a = K_p'(L' + T') = 4.5$$
$$L = L' + T' = 4.5$$

Table 5.4 gives the following PID controller parameters:

	$M_s = 1.4$	$M_s = 2.0$
K	0.32	0.67
T_i	14	7.6
T_d	2.6	1.7
b	0.33	0.39

Figure 5.13 shows the response of the closed-loop systems to a step change in setpoint followed by a step change in the load. The figure shows that the responses are in accordance with the specifications.

□

5.8 Conclusions

In this section we have developed new methods for tuning PID controllers. The methods are based on specifications in terms of rejection of load disturbances and measurement noise, sensitivity to modeling errors, and setpoint response. Rejection of load disturbances is the primary design criterion that is optimized by minimizing the integrated error. Modeling errors are captured by requiring that the maximum sensitivity be less than a specified value M_s. This value is a design variable that can be chosen by the user. Reasonable variables range from $M_s = 1.4$ to $M_s = 2$. The standard value is $M_s = 2$, but smaller values can be chosen if responses without overshoot are desired.

The method is based on the dominant pole design, wich requires that the transfer function of the process is known. The methods described in this section require the same parameters as the Ziegler-Nichols methods, and, in addition, κ for the frequency domain method and τ for the step response method. The tuning method, therefore, is called the Kappa-Tau method or the KT method for short. The tuning method works for processes that are typically encountered in process control. The KT-tuning techniques may be viewed as a generalization of the Ziegler-Nichols method.

For the step-response method the process is characterized by the apparent dead time, the apparent time constant, and the static gain. For the frequency-domain method, the parameters ultimate gain, ultimate period, and static gain characterize the process. The tuning rules are conveniently expressed using the normalized variables gain ratio κ and normalized dead time τ.

The KT method gives insight into the shortcomings of the Ziegler-Nichols rules. First, they avoid the poor damping obtained with the Ziegler-Nichols rule. Secondly, they give good tuning also for processes with long dead time. The results also show that knowledge about the gain ratio or the normalized dead time is required for tuning a PI controller. For PID control with small values of κ and τ, e.g., processes with integration, it is shown that improved tuning requires additional knowledge. This can be obtained from the impulse response of the system. The results admit assessment of the performance that can be achieved with the new tuning rules based on simple process characteristics.

5.9 References

The results given in this chapter were motivated by the desire to obtain tuning rules that are simple and significantly better than the

Ziegler-Nichols rules. The chapter is based on the ideas in (Hang *et al.*, 1991), (Åström *et al.*, 1992). The idea to use dimension-free parameters was used there. Notice, however, that the definitions of normalized dead time τ, and gain ratio κ are different. The reason for making changes is that it is helpful to have parameters that range from zero to one. Much more primitive tuning procedures were used in the previous work. The dominant pole design, which is the basis for the work, is given in (Persson, 1992), see also (Persson and Åström, 1992), (Persson and Åström, 1993), and (Persson, 1992). The idea of using dimension-free quantities for performance assessment is discussed in (Åström, 1991). They are also useful for an autonomous controller, see (Åström, 1992).

Automatic Tuning and Adaptation

6.1 Introduction

By combining the methods for determination of process dynamics (described in Chapter 2) with the methods for computing the parameters of a PID controller (described in Chapter 4), methods for automatic tuning of PID controllers can be obtained. By automatic tuning (or auto-tuning) we mean a method where the controller is tuned automatically on demand from a user. Typically the user will either push a button or send a command to the controller. An automatic tuning procedure consists of three steps:

- Generation of a process disturbance.
- Evaluation of the disturbance response.
- Calculation of controller parameters.

This is the same procedure that an experienced operator uses when tuning a controller manually. The process must be disturbed in some way in order to determine the process dynamics. This can be done in many ways, e.g., by adding steps, pulses, or sinusoids to the process input. The evaluation of the disturbance response may include a determination of a process model or a simple characterization of the response.

Industrial experience has clearly indicated that automatic tuning is a highly desirable and useful feature. Automatic tuning is sometimes called tuning on demand or one-shot tuning. Commercial PID controllers with automatic tuning facilities have only been available since the beginning of the eighties. There are several reasons for this. The recent development of microelectronics has made it possible to incorporate the additional program code needed for the automatic tuning at a reasonable cost. The interest in automatic tuning at universities is also quite new. Most of the research effort has been devoted to the related, but more difficult, problem of adaptive control.

Automatic tuning can also be performed using external equipment. These devices are connected to the control loop only during the tuning phase. When the tuning experiment is finished, the products suggest controller parameters. Since these products are supposed to work together with controllers from different manufacturers, they must be provided with quite a lot of information about the controller in order to give an appropriate parameter suggestion. The information required includes controller structure (standard, series, or parallel form), sampling rate, filter time constants, and units of the different controller parameters (gain or proportional band, minutes or seconds, time or repeats/time). The fact that PID controllers are parameterized in so many ways creates unnecessary difficulties.

Tuning facilities are also starting to appear in the distributed control systems. In this case it is possible to have a very powerful interaction of the user because of the graphics and computational capabilities available in the system.

Even when automatic tuning devices are used, it is important to obtain a certain amount of process knowledge. This is discussed in the next section. Automatic tuning is only one way to use the adaptive technique. Section 6.3 gives an overview of several adaptive techniques, as well as a discussion about the use of these techniques. The automatic tuning approaches can be divided into two categories, namely model-based approaches and rule-based approaches. In the model-based approaches, a model of the process is obtained explicitly, and the tuning is based on this model. Section 6.4 treats approaches were the model is obtained from transient response experiments, frequency response experiments, and parameter estimation. In the rule-based approaches, no explicit process model is obtained. The tuning is based instead on rules similar to those rules that an experienced operator uses to tune the controller manually. The rule-based approach is treated in Section 6.5.

Some industrial products with adaptive facilities are presented in Section 6.6. Four single-station controllers are presented: Foxboro EXACT (760/761), Alfa Laval Automation ECA400, Honeywell UDC 6000, and Yokogawa SLPC-181 and 281. Three tuning packages to be used within DCS systems, Fisher-Rosemount Intelligent Tuner and Gain Scheduler, Honeywell Looptune, and ABB DCS Tuner are also presented, as well as the process analyzer Techmation Protuner.

Adaptive techniques are closely related to diagnosis procedures. Section 6.7 gives an overview of both manual and automatic on-line diagnosis procedures. The chapter ends with conclusions and references in Sections 6.8 and 6.9

6.2 Process Knowledge

In this chapter we will discuss several methods for automatic tuning. Before going into details we must remark that poor behavior of a control loop can not always be corrected by tuning the controller. It is absolutely necessary to understand the reason for the poor behavior.

The process may be poorly designed so that there are long dead times, long time constants, nonlinearities, and inverse responses. Sensors and actuators may be poorly placed or badly mounted, and they may have bad dynamics. Typical examples are thermocouples with heavy casings that make their response slow, or on-off valve motors with long travel time. Valves may be oversized so that they only act over a small region. The sensor span may be too wide so that poor resolution is obtained, or it may also have excessive sensor noise. The procedure of investigating whether a process is well designed from a control point of view is called loop auditing.

There may also be failure and wear in the process equipment. Valves may have excessive stiction. There may be backlash due to wear. Sensors may drift and change their properties because of contamination.

If a control loop is behaving unsatisfactorily, it is essential that we first determine the reason for this before tuning is attempted. It would, of course, be highly desirable to have aids for the process engineer to do the diagnosis. Automatic tuning may actually do the wrong thing if it is not applied with care. For example, consider a control loop that oscillates because of friction in the actuator. Practically all tuning devices will attempt to stabilize the oscillation by reducing the controller gain. This will only increase the period of the oscillation!

Remember—no amount of so called "intelligence" in equipment can replace real process knowledge.

6.3 Adaptive Techniques

Techniques for automatic tuning grew out of research in adaptive control. Adaptation was originally developed to deal with processes with characteristics that were changing with time or with operating conditions. Practically all adaptive techniques can be used for automatic tuning. The adaptive controller is simply run until the parameters have converged and the parameters are then kept constant. The drawback with this approach is that adaptive controllers may require prior information. There are many special techniques that can be used for this purpose. Industrial experience has shown that this is probably the most useful application of adaptive techniques. Gain scheduling is also a very effective technique to cope with processes that change their

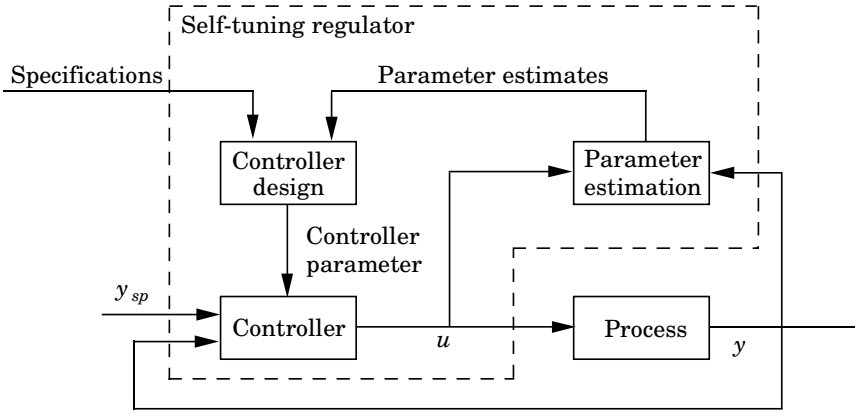

Figure 6.1 Block diagram of an indirect adaptive controller.

characteristics with operating conditions. An overview of these techniques will be given in this section. In this book the phrase *adaptive techniques* will include auto-tuning, gain scheduling, and adaptation.

Adaptive Control

By adaptive control we mean a controller whose parameters are continuously adjusted to accommodate changes in process dynamics and disturbances. Adaptation can be applied both to feedback and feedforward control parameters. It has proven particularly useful for feedforward control. The reason for this is that feedforward control requires good models. Adaptation is, therefore, almost a prerequisite for using feedforward control. Adaptive control is sometimes called continuous adaptation to emphasize that parameters are changed continuously.

There are two types of adaptive controllers based on direct and indirect methods. In a direct method, controller parameters are adjusted directly from data in closed-loop operation. In indirect methods, the parameters of a process model are updated on-line by recursive parameter estimation. (Compare with Section 2.7 where parameter estimation was discussed briefly.) The controller parameters are then obtained by some method for control design. In direct adaptive control the parameters of the controller are updated directly. The self-tuning regulator is a typical example of a direct adaptive controller; the model reference system is an example of an indirect adaptive controller. There is a large number of methods available both for direct and indirect methods. They can conveniently be described in terms of the methods used for modeling and control design.

A block diagram of a direct adaptive controller is shown in Fig-

ure 6.1. There is a parameter estimator that determines the parameters of the model based on observations of process inputs and outputs. There is also a design block that computes controller parameters from the model parameters. If the system is operated as a tuner, the process is excited by an input signal. The parameters can either be estimated recursively or in batch mode. Controller parameters are computed and the controller is commissioned. If the system is operated as an adaptive controller, parameters are computed recursively and controller parameters are updated when new parameter values are obtained.

Automatic Tuning

By automatic tuning (or auto-tuning) we mean a method where a controller is tuned automatically on demand from a user. Typically the user will either push a button or send a command to the controller. Industrial experience has clearly indicated that this is a highly desirable and useful feature. Automatic tuning is sometimes called tuning on demand or one-shot tuning. Auto-tuning can be built into the controllers. Practically all controllers can benefit from tools for automatic tuning. This will drastically simplify the use of controllers. Single loop controllers and distributed systems for process control are important application areas. Most of these controllers are of the PID type. This is a vast application area because there are millions of controllers of this type in use. Automatic tuning is currently widely used in PID controllers.

Auto-tuning can also be performed with external devices that are connected to a process. Since these systems have to work with controllers from different manufacturers, they must be provided with information about the controller structure in order to give an appropriate parameter suggestion. Such information includes controller structure (standard, series, or parallel form), sampling rate, filter time constants, and units of the different controller parameters (gain or proportional band, minutes or seconds, time or repeats/time).

Gain Scheduling

Gain scheduling is a technique that deals with nonlinear processes, processes with time variations, or situations where the requirements on the control change with the operating conditions. To use the technique it is necessary to find measurable variables, called scheduling variables, that correlate well with changes in process dynamics. The scheduling variable can be, for instance, the measured signal, the control signal, or an external signal. For historical reasons the phrase *gain scheduling* is used even if other parameters than

the gain, e.g., derivative time or integral time, are changed. Gain scheduling is a very effective way of controlling systems whose dynamics change with the operating conditions. Gain scheduling has not been used much because of the effort required to implement it. When combined with auto-tuning, however, gain scheduling is very easy to use.

A block diagram of a system with gain scheduling is shown in Fig. 6.2. The system can be viewed as having two loops. There is an inner loop, composed of the process and the controller, and an outer loop, which adjusts the controller parameters based on the operating conditions.

The notion of gain scheduling was originally used for flight control systems, but it is being used increasingly in process control. It is, in fact, a standard ingredient in some single-loop PID controllers. For process control applications significant improvements can be obtained by using just a few sets of controller parameters.

Gain scheduling is often an alternative to adaptation. It has the advantage that it can follow rapid changes in the operating conditions. The key problem is to find suitable scheduling variables. Possible choices are the control signal, the process variable, or an external signal. Production rate is often a good choice in process control applications, since time constants and time delays are often inversely proportional to production rate.

Development of a schedule may take a substantial engineering effort. The availability of automatic tuning can significantly reduce the effort because the schedules can then be determined experimentally. A scheduling variable is first determined. Its range is quantitized into a number of discrete operating conditions. The controller parameters are then determined by automatic tuning when the system is running in one operating condition. The parameters are stored in a table. The procedure is repeated until all operating conditions are covered. In this way it is easy to install gain scheduling into a

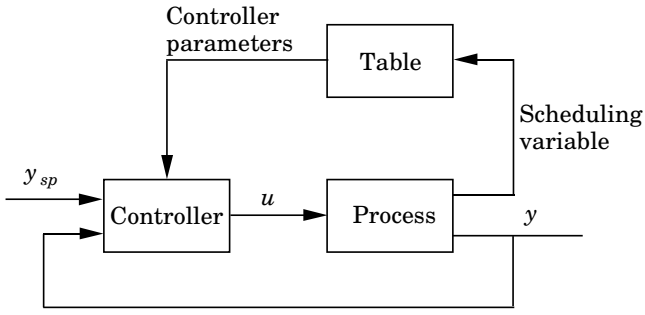

Figure 6.2 Block diagram of a system with gain scheduling.

computer-controlled system by programming a table for storing and recalling controller parameters and appropriate commands to accomplish this.

Uses of Adaptive Techniques

We have described three techniques that are useful in dealing with processes that have properties changing with time or with operating conditions. In Figure 6.3 is a diagram that guides the choice among the different techniques.

Controller performance is the first thing to consider. If the requirements are modest, a controller with constant parameters and conservative tuning can be used. With higher demands on performance, other solutions should be considered. If the process dynamics are constant, a controller with constant parameters should be used. The parameters of the controller can be obtained using auto-tuning.

If the process dynamics or the nature of the disturbances are changing, it is useful to compensate for these changes by changing the controller. If the variations can be predicted from measured signals, gain scheduling should be used because it is simpler and gives superior and more robust performance than the continuous adaptation. Typical examples are variations caused by nonlinearities in the control loop. Auto-tuning can be used to build up the gain schedules.

There are also cases where the variations in process dynamics

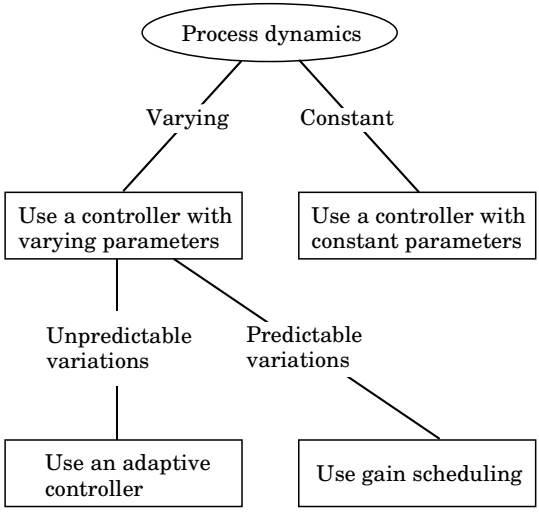

Figure 6.3 When to use different adaptive techniques.

are not predictable. Typical examples are changes due to unmeasurable variations in raw material, wear, fouling etc. These variations cannot be handled by gain scheduling, since no scheduling variable is available, but must be dealt with by adaptation. An auto-tuning procedure is often used to initialize the adaptive controller. It is then sometimes called pre-tuning or initial tuning.

Feedforward control deserves special mentioning. It is a very powerful method for dealing with measurable disturbances. Use of feedforward control, however, requires good models of process dynamics. It is difficult to tune feedforward control loops automatically on demand, since the operator often cannot manipulate the disturbance used for the feedforward control. To tune the feedforward controller it is necessary to wait for an appropriate disturbance. Adaptation, therefore, is particularly useful for the feedforward controller.

6.4 Model-Based Methods

This section gives an overview of automatic tuning approaches that are based on an explicit derivation of a process model. Models can be obtained in many ways, as seen in Chapter 2. In this section we discuss approaches based on transient responses, frequency responses, and parameter estimation.

Transient Response Methods

Auto-tuners can be based on open-loop or closed-loop transient response analysis. Methods for determining the transient response were discussed in Section 2.3. The most common methods are based on step or pulse responses, but there are also methods that can use many other types of perturbations.

Open-Loop Tuning

A simple process model can be obtained from an open-loop transient response experiment. A step or a pulse is injected at the process input, and the response is measured. To perform such an experiment, the process must be stable. If a pulse test is used, the process may include an integrator. It is necessary that the process be in equilibrium when the experiment is begun.

There are, in principle, only one or two parameters that must be set *a priori*, namely the amplitude and the signal duration. The amplitude should be chosen sufficiently large, so that the response is easily visible above the noise level. On the other hand, it should

be as small as possible in order not to disturb the process more than necessary and to keep the dynamics linear. The noise level can be determined automatically at the beginning of the tuning experiment. However, even if the noise level is known, we cannot decide a suitable magnitude of a step in the control signal without knowing the gain of the process. Therefore, it must be possible for the operator to decide the magnitude.

The duration of the experiment is the second parameter that normally is set *a priori*. If the process is unknown, it is very difficult to determine whether a step response has settled or not. An intuitive approach is to say that the measurement signal has reached its new steady state if its rate of change is sufficiently small. The rate of change is related, however, to the time constants of the process, which are unknown. If a pulse test is used, the duration of the pulse should also be related to the process time constants.

Many methods can be used to extract process characteristics from a transient response experiment. Most auto-tuners determine the static gain, the dominant time constant, and the apparent dead time. The static gain is easy to find accurately from a step-response experiment by comparing the stationary values of the control signal and the measurement signal before and after the step change. The time constant and the dead time can be obtained in several ways (see Section 2.3). The method of moments, presented in Section 2.4, is an appealing method, which is relatively insensitive to high-frequency disturbances.

The transient response methods are often used in a pre-tuning mode in more complicated tuning devices. The main advantage of the methods, namely that they require little prior knowledge, is then exploited. It is also easy to explain the methods to plant personnel. The main drawback with the transient response methods is that they are sensitive to disturbances. This drawback is less important if they are used only in the pre-tuning phase.

Closed-Loop Tuning

Automatic tuning based on transient response identification can also be performed on line. The steps or pulses are then added either to the setpoint or to the control signal. There are also auto-tuners that do not introduce any transient disturbances. Perturbations caused by setpoint changes or load disturbances are used instead. In these cases it is necessary to detect that the perturbations are sufficiently large compared to the noise level.

Closed-loop tuning methods cannot be used on unknown processes. Some kind of pre-tuning must always be performed in order to close the loop in a satisfactory way. On the other hand, they do not usually require any additional *a priori* information. The magnitude of

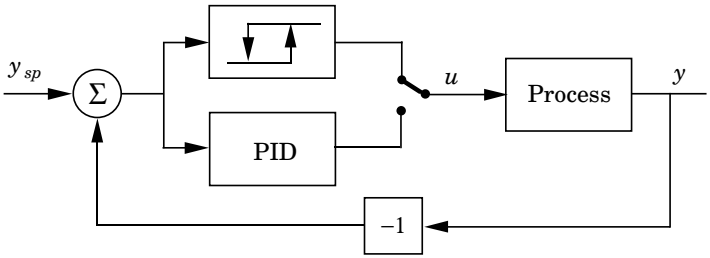

Figure 6.4 The relay auto-tuner. In the tuning mode the process is connected to relay feedback.

the step changes in setpoint are easily determined from the desired, or accepted, change in the measurement signal.

Since a proper closed-loop transient response is the goal for the design, it is appealing to base tuning on closed-loop responses. It is easy to give design specifications in terms of the closed-loop transient response, e.g., damping, overshoot, closed-loop time constants, etc. The drawback is that the relation between these specifications and the PID parameters is normally quite involved. Heuristics and logic are required therefore.

Frequency Response Methods

There are also auto-tuners that are based on frequency response methods. In Section 2.5, it was shown how frequency response techniques could be used to determine process dynamics.

Use of the Relay Method

In traditional frequency response methods, the transfer function of a process is determined by measuring the steady-state responses to sinusoidal inputs. A difficulty with this approach is that appropriate frequencies of the input signal must be chosen *a priori*. A special method, where an appropriate frequency of the input signal is generated automatically, was described in Section 2.5. The idea was simply to introduce a nonlinear feedback of the relay type in order to generate a limit cycle oscillation. With an ideal relay the method gives an input signal to the process with a period close to the ultimate frequency of the open-loop system.

A block diagram of an auto-tuner based on the relay method is shown in Figure 6.4. Notice that there is a switch that selects either relay feedback or ordinary PID feedback. When it is desired to tune the system, the PID function is disconnected and the system is connected to relay control. The system then starts to oscillate. The

period and the amplitude of the oscillation is determined when steady-state oscillation is obtained. This gives the ultimate period and the ultimate gain. The parameters of a PID controller can then be determined from these values, e.g., using the Ziegler-Nichols frequency response method. The PID controller is then automatically switched in again, and the control is executed with the new PID parameters.

This tuning device has one parameter that must be specified in advance, namely, the initial amplitude of the relay. A feedback loop from measurement of the amplitude of the oscillation to the relay amplitude can be used to ensure that the output is within reasonable bounds during the oscillation. It is also useful to introduce hysteresis in the relay. This reduces the effects of measurement noise and also increases the period of the oscillation. With hysteresis there is an additional parameter. This can be set automatically, however, based on a determination of the measurement noise level. Notice that there is no need to know time scales *a priori* since the ultimate frequency is determined automatically.

In the relay method, an oscillation with suitable frequency is generated by a static nonlinearity. Even the order of magnitude of the time constant of the process can be unknown. Therefore, this method is not only suitable as a tuning device; it can also be used in pre-tuning. It is also suitable for determination of sampling periods in digital controllers.

The relay tuning method also can be modified to identify several points on the Nyquist curve. This can be accomplished by making several experiments with different values of the amplitude and the hysteresis of the relay. A filter with known characteristics can also be introduced in the loop to identify other points on the Nyquist curve. If the static process gain is determined, the KT tuning method presented in Chapter 5 can be used.

On-Line Methods

Frequency response analysis can also be used for on-line tuning of PID controllers. The relay feedback technique can be used, as described in Section 2.5. By introducing bandpass filters, the signal content at different frequencies can be investigated. From this knowledge, a process model given in terms of points on the Nyquist curve can be identified on line. In this auto-tuner the choice of frequencies in the bandpass filters is crucial. This choice can be simplified by using the tuning procedure described above in a pre-tuning phase.

Parameter Estimation Methods

A common tuning procedure is to use recursive parameter estimation to determine a low-order discrete time model of the process. The parameters of the low-order model obtained are then used in a design scheme to calculate the controller parameters. An auto-tuner of this type can also be operated as an adaptive controller that changes the controller parameters continuously. Auto-tuners based on this idea, therefore, often have an option for continuous adaptation.

The main advantage of auto-tuners of this type is that they do not require any specific type of excitation signal. The control signal can be a sequency of manual changes of the control signal, for example, or the signals obtained during normal operation. A drawback with auto-tuners of this type is that they require significant prior information. A sampling period for the identification procedure must be specified; it should be related to the time constants of the closed-loop system. Since the identification is performed on line, a controller that at least manages to stabilize the system is required. Systems based on this identification procedure need a pre-tuning phase, which can be based on the methods presented earlier in this section.

6.5 Rule-Based Methods

This section treats automatic tuning methods that do not use an explicit model of the process. Tuning is based instead on the idea of mimicking manual tuning by an experienced process engineer.

Controller tuning is a compromise between the requirement for fast control and the need for stable control. Table 6.1 shows how stability and speed change when the PID controller parameters are changed. Note that the table only contains rules of thumb. There are exceptions. For example, an increased gain often results in more stable control when the process contains an integrator. The same rules can also be illustrated in tuning maps. See, for example, the tuning map for PI control in Figure 4.13.

Table 6.1 Rules of thumb for the effects of the controller parameters on speed and stability in the control loop.

	Speed	Stability
K increases	increases	reduces
T_i increases	reduces	increases
T_d increases	increases	increases

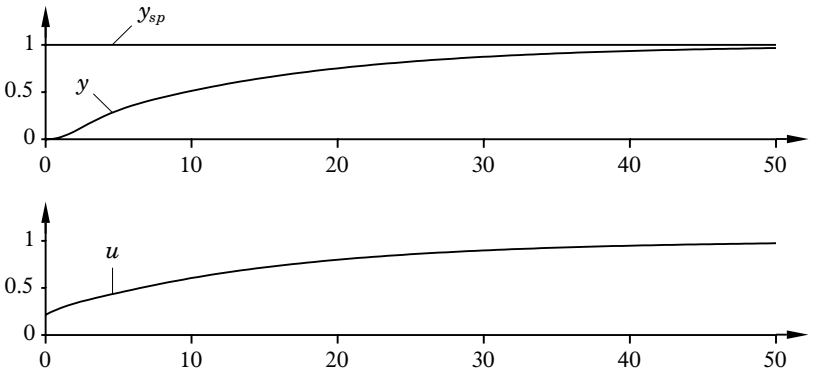

Figure 6.5 A setpoint response where a correct rule is to increase the gain and decrease the integral time. The upper diagram shows setpoint y_{sp} and process output y, and the lower diagram shows control signal u.

The rule-based automatic tuning procedures wait for transients, setpoint changes, or load disturbances, in the same way as the model-based methods. When such a disturbance occurs, the behavior of the controlled process is observed. If the control deviates from the specifications, the controller parameters are adjusted based on some rules.

Figures 6.5 and 6.6 show setpoint changes of control loops with a poorly tuned PI controller. The response in Figure 6.5 is very sluggish. Here, a correct rule is to increase the gain and to decrease the integral time. Figure 6.6 also shows a sluggish response because of a too large integral time. The response is also oscillatory because of a too high gain. A correct rule, therefore, is to decrease both the gain and the integral time.

If graphs like those in Figures 6.5 and 6.6 are provided, it is easy for an experienced operator to apply correct rules for controller tuning. To obtain a rule-based automatic tuning procedure, the graphs must be replaced by quantities that characterize the responses. Commonly used quantities are overshoot and decay ratio to characterize the stability of the control loop, and time constant and oscillation frequency to characterize the speed of the control loop.

It is rather easy to obtain relevant rules that tell whether the different controller parameters should be decreased or increased. However, it is more difficult to determine *how much* they should be decreased or increased. The rule-based methods are, therefore, more suitable for continuous adaptation where rather small successive changes in the controller parameters are performed after each transient.

The rule-based methods have a great advantage compared to the

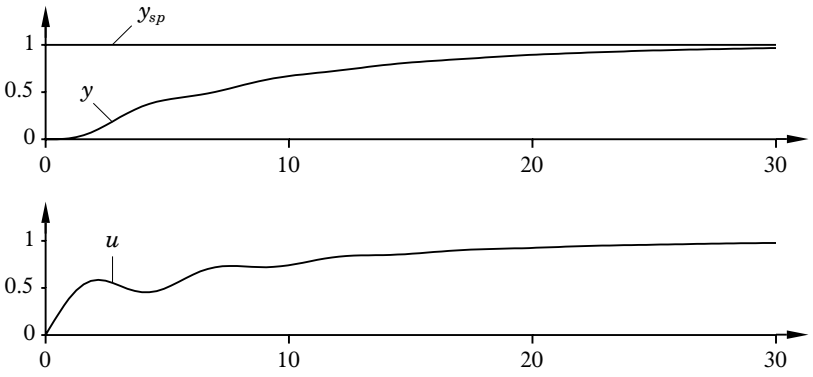

Figure 6.6 A setpoint response where a correct rule is to decrease the gain and decrease the integral time. The upper diagram shows setpoint y_{sp} and process output y, and the lower diagram shows control signal u.

model-based approaches when they are used for continuous adaptation, namely, that they handle load disturbances efficiently and in the same way as setpoint changes. The model-based approaches are well suited for setpoint changes. However, when a load disturbance occurs, the transient response is caused by an unknown input signal. To obtain an input-output process model under such circumstances is not so easy.

A drawback with the rule-base approaches is that they normally assume that the setpoint changes or load disturbances are isolated steps or pulses. Two setpoint changes or load disturbances applied shortly after each other may result in a process output that invokes an erroneous controller tuning rule.

6.6 Commercial Products

In this section, some industrial products with automatic tuning facilities will be presented. Four controllers are presented; the Foxboro EXACT (760/761), which uses step-response analysis for automatic tuning, and pattern recognition technique and heuristic rules for its adaptation; the Alfa Laval Automation ECA400 controller, which uses relay auto-tuning and model-based adaptation; the Honeywell UDC 6000 controller, which uses step-response analysis for automatic tuning and a rule base for adaptation; and the Yokogawa SLPC-181 and 281, which use step-response analysis for auto-tuning and a model-based adaptation.

Four tuning devices are also described. Intelligent Tuner and

Gain Scheduler is a software package used in distributed control systems by Fisher-Rosemount. Looptune is a tuning program package to be used within the DCS system Honeywell TDC 3000. DCS Tuner is a software package for controller tuning in the ABB Master system. The Techmation Protuner is a process analyzer, that is only connected to the control loop during the tuning and analyzing phase.

Foxboro EXACT (760/761)

The single-loop adaptive controller EXACT was released by Foxboro in October 1984. The reported application experience in using this controller has been favorable. The adaptive features are also available in DCS products.

Process Modeling

Foxboro's system is based on the determination of dynamic characteristics from a transient, which results in a sufficiently large error. If the controller parameters are reasonable, a transient error response of the type shown in Figure 6.7 is obtained. Heuristic logic is used to detect that a proper disturbance has occurred and to detect peaks e_1, e_2, e_3, and oscillation period T_p.

Control Design

The user specifications are given in terms of maximum overshoot and maximum damping. They are defined as

$$\begin{aligned} \text{damping} &= \frac{e_3 - e_2}{e_1 - e_2} \\ \text{overshoot} &= \left| \frac{e_2}{e_1} \right| \end{aligned} \tag{6.1}$$

for both setpoint changes and load disturbances. Note that the definition of damping here is different from the damping factor associated with a standard second-order system.

The controller structure is of the series form. From the response to a setpoint change or a load disturbance, the actual damping and overshoot pattern of the error signal is recognized, and the period of oscillation T_p measured. This information is used by the heuristic rules to directly adjust the controller parameters to give the specified damping and overshoot. Examples of heuristics are to decrease proportional band PB, integral time T_i and derivative time T_d, if distinct peaks are not detected. If distinct peaks have occurred and both damping and overshoot are less than the maximum values, PB is decreased.

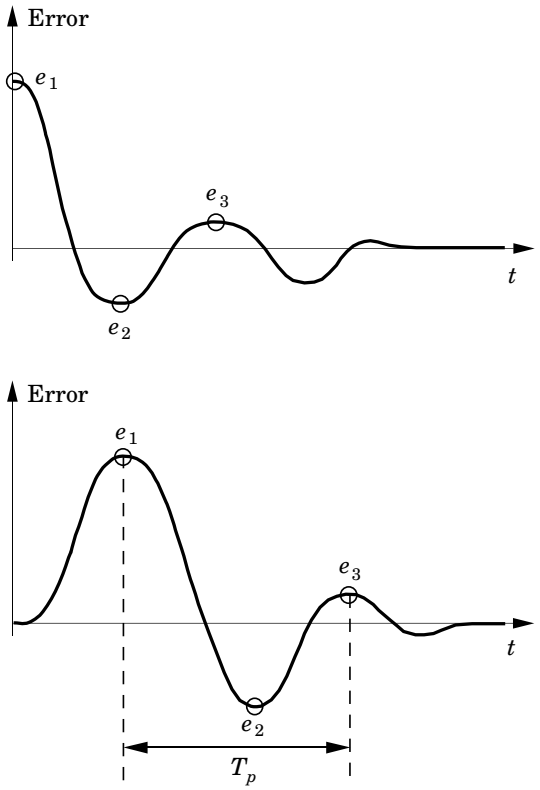

Figure 6.7 Response to a step change of setpoint (upper curve) and load (lower curve).

Prior Information and Pre-Tuning

The controller has a set of required parameters that must be given either by the user from prior knowledge of the loop or estimated using the pre-tune function (Pre-Tune is Foxboro's notation for auto-tuning). The required parameters are

- Initial values of PB, T_i and T_d.
- Noise band (NB). The controller starts adaptation whenever the error signal exceeds two times NB.
- Maximum wait time ($W_{\max}$). The controller waits for a time of $W_{\max}$ for the occurrence of the second peak.

If the user is unable to provide the required parameters, a pre-tune function that estimates these quantities can be activated. To activate the pre-tune function, the controller must first be put in manual. When the pre-tune function is activated, a step input is generated. The process parameters static gain K_p, dead time L and time con-

stant T are then obtained from a simple analysis of the process reaction curve. The controller parameters are calculated using a Ziegler-Nichols-like formula:

$$\begin{aligned} PB &= 120K_pL/T \\ T_i &= 1.5L \\ T_d &= T_i/6 \end{aligned} \tag{6.2}$$

Notice that the controller is parameterized in the series form. Maximum wait time, $W_{\max}$, is also determined from the step response as:

$$W_{\max} = 5L$$

The noise band is determined during the last phase of the pre-tune mode. The control signal is first returned to the level before the step change. With the controller still in manual and the control signal held constant, the output is passed through a high-pass filter. The noise band is calculated as an estimate of the peak-to-peak amplitude of the output from the high-pass filter.

The estimated noise band (NB) is used to initialize the derivative term. Derivative action is decreased when the noise level is high in order to avoid large fluctuations in the control signal. The derivative term is initialized using the following logic:

1. Calculate a quantity $Z = (3.0 - 2NB)/2.5$;
2. if $Z > 1$ then set $T_d = T_i/6$;
3. if $Z < 0$ then set $T_d = 0$;
4. if $0 < Z < 1$ then set $T_d = Z \cdot T_i/6$.

Apart from the set of required parameters, there is also a set of optional parameters. If these are not supplied by the user then the default values will be used. The optional parameters are as follows (default values in parenthesis):

- Maximum allowed damping (0.3)
- Maximum allowed overshoot (0.5)
- Derivative factor (1). The derivative term is multiplied by the derivative factor. This allows the derivative influence to be adjusted by the user. Setting the derivative factor to zero results in PI control.
- Change Limit (10). This factor limits the controller parameters to a certain range. Thus, the controller will not set the PB, T_i and T_d values higher than ten times or lower than one tenth of their initial values if the default of 10 is used for the change limit.

Alfa Laval Automation ECA400

This controller was announced by Alfa Laval Automation in 1988. It has the adaptive functions of automatic tuning, gain scheduling, and continuous adaptation of feedback and feedforward control. An earlier product, ECA40, which only has auto-tuning and gain scheduling was announced in 1986.

Automatic Tuning

The auto-tuning is performed using the relay method in the following way. The process is brought to a desired operating point, either by the operator in manual mode or by a previously tuned controller in automatic mode. When the loop is stationary, the operator presses a tuning button. After a short period, when the noise level is measured automatically, a relay with hysteresis is introduced in the loop, and the PID controller is temporarily disconnected (see Figure 6.4). The hysteresis of the relay is determined automatically from the noise level. During the oscillation, the relay amplitude is adjusted so that a desired level of the oscillation amplitude is obtained. When an oscillation with constant amplitude and period is obtained, the relay experiment is interrupted and $G_p(i\omega_0)$, i.e., the value of the transfer function G_p at oscillation frequency ω_0, is calculated using describing function analysis.

Control Design

The PID algorithm in the ECA400 controller is of series form. The identification procedure provides a process model in terms of one point $G_p(i\omega_0)$ on the Nyquist curve. By introducing the PID controller $G_c(i\omega)$ in the control loop, it is possible to give the Nyquist curve of the compensated system G_pG_c a desired location at frequency ω_0. For most purposes, the PID parameters are chosen so that $G_p(i\omega_0)$ is moved to the point where

$$G_p(i\omega_0)G_c(i\omega_0) = 0.5e^{-i135\pi/180} \tag{6.3}$$

This design method can be viewed as a combination of phase- and amplitude-margin specification. Since there are three adjustable parameters, K, T_i, and T_d, and the design criterion (6.3) only specifies two parameters, it is required, furthermore, that

$$T_i = 4T_d \tag{6.4}$$

For some simple control problems, where the process is approximately a first-order system, the derivative action is switched off and only a PI controller is used. This kind of process is automatically detected.

For this PI controller, the following design is used:

$$\begin{aligned} K &= 0.5/|G_p(i\omega_0)| \\ T_i &= 4/\omega_0 \end{aligned} \tag{6.5}$$

There is also another situation in which it is desirable to switch off the derivative part, namely for processes with long dead time. If the operator tells the controller that the process has a long dead time, a PI controller with the following design will replace the PID controller.

$$\begin{aligned} K &= 0.25/|G_p(i\omega_0)| \\ T_i &= 1.6/\omega_0 \end{aligned} \tag{6.6}$$

Gain Scheduling

The ECA400 controller also has gain scheduling. Three sets of controller parameters can be stored. The parameters are obtained by using the auto-tuner three times, once at every operating condition. The actual value of the scheduling variable, which can be the controller output, the measurement signal, or an external signal, determines which parameter set to use. The ranges of the scheduling variable where different parameters are used can also be given by the user.

Adaptive Feedback

The ECA400 controller uses the information from the relay feedback experiment to initialize the adaptive controller. Figure 6.8 shows the principle of the adaptive controller. The key idea is to track the point on the Nyquist curve obtained by the relay auto-tuner. It is performed in the following way. The control signal u and the measurement signal

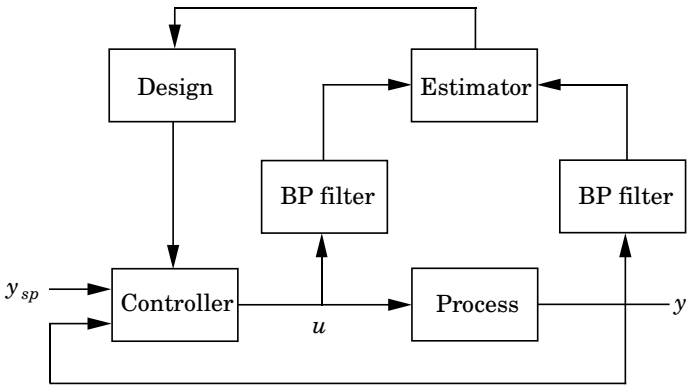

Figure 6.8 The adaptive control procedure in ECA400.

y are filtered through narrow band-pass filters centered at frequency ω_0. This frequency is obtained from the relay experiment. The signals are then analyzed in a least-squares estimator which, provides an estimate of the point $G_p(i\omega_0)$.

Adaptive Feedforward Control

Adaptive feedforward is another feature in ECA400. The adaptive feedforward control procedure is initialized by the relay auto-tuner. A least-squares algorithm is used to identify parameters a and b in the model

$$y(t) = au(t - 4h) + bv(t - 4h) \tag{6.7}$$

where y is the measurement signal, u is the control signal and v is the disturbance signal that should be fed forward. The sampling interval h is determined from the relay experiment as $h = T_0/8$, where T_0 is the oscillation period. The feedforward compensator has the simple structure

$$\Delta u_{ff}(t) = k_{ff}(t)\Delta v(t) \tag{6.8}$$

where the feedforward gain k_{ff} is calculated from the estimated process parameters

$$k_{ff}(t) = -0.8\,\frac{\hat{b}(t)}{\hat{a}(t)} \tag{6.9}$$

Operator Interface

The initial relay amplitude is given a default value suitable for a wide range of process control applications. This parameter is not critical since it will be adjusted after the first half period to give an admissible amplitude of the limit cycle oscillation. The operation of the auto-tuner is then very simple. To use the tuner, the process is simply brought to an equilibrium by setting a constant control signal in manual mode. The tuning is activated by pushing the tuning switch. The controller is automatically switched to automatic mode when the tuning is complete.

The width of the hysteresis is set automatically, based on measurement of the noise level in the process. The lower the noise level, the lower the amplitude required from the measured signal. The relay amplitude is controlled so that the oscillation is kept at a minimum level above the noise level.

The following are some optional settings that may be set by the operator:

Control Design [normal, PI, dead time]. The default design method [normal] can result in either a PI or a PID controller depending on whether the process is of first order or not. (See the discussion about

control design above.) The operator can force the controller to be PI by selecting [PI]. The PI design method suited for processes with long dead time is chosen by selecting the option [dead time]. The control design can be changed either before or after a tuning, as well as during the adaptation.

Reset [Yes/No]. A reset of information concerning the auto-tuner, the gain-scheduling, or the adaptive controller can be done. Some information from a tuning is saved and used to improve the accuracy of the subsequent tunings, including the noise level, the initial relay amplitude, and the period of the oscillation. If a major change is made in the control loop, for instance if the controller is moved to another loop, the operator can reset the tuning information. Resetting the adaptive controller will reset the controller parameters to those obtained by the relay auto-tuner.

Initial relay amplitude. In some very sensitive loops, the initial input step of the relay experiment may be too large. The initial step can then be changed by the operator.

Gain scheduling reference. The switches between the different sets of PID parameters in the gain-scheduling table can be performed with different choices of the scheduling reference. The scheduling can be based on the control signal, the measurement signal, or an external signal. The gain scheduling is active only when a reference signal is configured.

Honeywell UDC 6000

The Honeywell UDC 6000 controller has an adaptive function called Accutune. Accutune uses both model-based procedures and rule-based procedures. It can only be used on stable processes. Consequently, integrating processes can not be treated.

Initial Tuning

The adaptive procedure is initialized by a step-response experiment. The user brings the process variable to a point some distance away from the desired setpoint in manual and waits for steady state. Switching to automatic mode will initiate an open-loop step response experiment, where the size of the step is calculated to be so large that it is supposed to take the process variable to the setpoint.

During the experiment, the process variable and its derivative are continuously monitored. Dead time L is calculated as the time interval between the step change and the moment the process variable crosses a certain small limit.

If the derivative of the process variable continuously decreases from the start, it is concluded that the process is of first order. Static

gain K_p and time constant T_1 are then calculated as

$$\begin{aligned} K_p &= \frac{y_2 + \dot{y}_2 T_1}{\Delta u} \\ T_1 &= \frac{y_2 - y_1}{\dot{y}_1 - \dot{y}_2} \end{aligned} \tag{6.10}$$

where y_1 and y_2 are two measurements of the process variable, $\dot{y}_1$ and $\dot{y}_2$ are estimates of the derivatives of the process variable at the corresponding instants, and Δu is the step size of the control signal. The values y_i are calculated as distances from the value at the onset of the step. These calculations can be performed before the steady-state is reached. It is claimed that the process is identified in a time less than one third of the time constant. The controller is then switched to automatic mode and controlled to the setpoint. When this is done, a fine adjustment of the parameters is done by calculating the gain K_p from the steady-state levels.

If the derivative of the process variable increases to a maximum and then decreases, the process is identified as a second-order process. The step response of a second-order process with two time constants is

$$y(t) = K_p \left(1 + \frac{T_2 e^{-t/T_2} - T_1 e^{-t/T_1}}{T_1 - T_2} \right) \tag{6.11}$$

This equation is used to calculate static gain K_p and time constants T_1 and T_2. Process identification is performed in two steps. A first calculation is done shortly after the time of maximum slope. The controller is then switched to automatic mode and controlled to the setpoint. When steady state is reached, the parameters are recalculated using the additional information of steady state levels. The equations for calculating the process parameters are

$$\begin{aligned} K_p &= \frac{y(t_{\max}) + \dot{y}(t_{\max})(T_1 + T_2)}{\Delta u} \\ T_1 + T_2 &= \frac{1 - N^2}{N \ln\left(\frac{1}{N}\right)} t_{\max} \\ N &= \frac{T_1}{T_2} \end{aligned} \tag{6.12}$$

where $t_{\max}$ is the time from the start of the rise to the point of maximum slope. These three equations have four unknowns: K_p, T_1, T_2 and N. At the first time of identification, shortly after the time of maximum slope, it is assumed that $N = 6$, and K_p, T_1, and T_2, are determined from the equations. When steady state is reached, gain K_p is calculated from the steady-state levels. This provides the possibility to calculate the other three unknowns from the three equations above.

Adaptation

The UDC 6000 controller also has continuous adaptation. The adaptation mechanism is activated when the process variable changes more than 0.3% from the setpoint or if the setpoint changes more than a prescribed value. (More details are given below.)

The full details of the adaptive controller are not published, but some of the rules in the heuristic rule base will be presented. The controller monitors the behavior of the process variable and makes the following adjustments:

1. The controller detects oscillations in the process variable. If the oscillation frequency, ω_0, is less than $1/T_i$, then the integral time is increased to $T_i = 2/\omega_0$.
2. If the oscillation frequency ω_0 is greater than $1/T_i$, then the derivative time is chosen as $T_d = 1/\omega_0$.
3. If the oscillation remains after adjustment 1 or 2, the controller will cut its gain K in half.
4. If a load disturbance or a setpoint change gives a response with a damped oscillation, the derivative time is chosen to be $T_d = 1/\omega_0$.
5. If a load disturbance or a setpoint change gives a sluggish response, where the time to reach setpoint is longer than $L+T_1+T_2$, both integral time T_i and derivative time T_d are divided by a factor of 1.3.
6. If the static process gain K_p changes, the controller gain K is changed so that the product KK_p remains constant.

Control Design

The UDC 6000 uses a controller on series form that has the transfer function

$$G_c(s) = K\frac{(1+sT_i)(1+sT_d)}{sT_i(1+0.125sT_d)}$$

The design goal is to cancel the process poles with the two zeros in the controller. If there is no dead time in the process, the controller parameters are chosen in the following way:

First-order process	Second-order process
$K = 24/K_p$	$K = 6/K_p$
$T_i = 0.16T_1$	$T_i = T_1$
$T_d = 0$	$T_d = T_2$

For processes with dead time, the controller parameters are determined as follows:

First-order process	Second-order process
$K = \dfrac{3}{K_p\,(1 + 3L/T_i)}$	$K = \dfrac{3}{K_p\,(1 + 3L/T_i)}$
$T_i = T_1$	$T_i = T_1 + T_2$
$T_d = 0$	$T_d = \dfrac{T_1 T_2}{T_1 + T_2}$

Operator Interface

The following are some optional parameters that may be set by the operator:

- Select whether adaptation should be performed during setpoint changes only, or during both setpoint changes and load disturbances.
- Set the minimum value of setpoint change that will activate the adaptation. Range: ±5% to ±15%.

Yokogawa SLPC-181, 281

The Yokogawa SLPC-181 and 281 both use a process model as a first-order system with dead time for calculating the PID parameters. A nonlinear programming technique is used to obtain the model. The PID parameters are calculated from equations developed from extensive simulations. The exact equations are not published.

Two different controller structures are used.

$$\begin{aligned} &1: \quad u = K\left(-y + \frac{1}{T_i}\int e dt - T_d\frac{dy_f}{dt}\right) \\ &2: \quad u = K\left(e + \frac{1}{T_i}\int e dt - T_d\frac{dy_f}{dt}\right) \end{aligned} \tag{6.13}$$

where

$$\frac{dy_f}{dt} = \frac{N}{T_d}(y - y_f) \tag{6.14}$$

The first structure is recommended if load disturbance rejection is most important, and structure 2 if setpoint responses are most important. The setpoint can also be passed through two filters in series:

$$\text{Filter 1:} \quad \frac{1 + \alpha_i s T_i}{1 + s T_i} \qquad \text{Filter 2:} \quad \frac{1 + \alpha_d s T_d}{1 + s T_d} \tag{6.15}$$

Table 6.2 Setpoint response specifications used in the Yokogawa SLPC-181 and 281.

Type	Features	Criteria
1	no overshoot	no overshoot
2	5% overshoot	ITAE minimum
3	10% overshoot	IAE minimum
4	15% overshoot	ISE minimum

where α_i and α_d are parameters set by the user, mainly to adjust the overshoot of the setpoint response. The effects of these two filters are essentially equivalent to setpoint weighting. It can be shown that $\alpha_i = b$, where b is the setpoint weighting factor.

The user specifies the type of setpoint response performance according to Table 6.2. A high overshoot will, of course, yield a faster response. The controller has four adaptive modes:

Auto mode. The adaptive control is on. PID parameters are automatically updated.

Monitoring mode. In this mode, the computed model and the PID parameters are only displayed. This mode is useful for validating the adaptive function or checking the process dynamics variations during operation.

Auto startup mode. This is used to compute the initial PID parameters. An open loop step response is used to estimate the model.

On-demand mode. This mode is used to make a setpoint change. When the on-demand tuning is requested, a step change is applied to the process input in closed loop. The controller estimates the process model using the subsequent closed-loop response.

The controller constantly monitors the performance of the system by computing the ratio of the variances of process output and model output. This ratio is expected to be about 1. If it is greater than 2 or less than 0.5, a warning message for retuning of the controller is given. Dead time and feedforward compensation are available for the constant gain controller, but they are not recommended by the manufacturer to be used in conjunction with adaptation.

Fisher-Rosemount Intelligent Tuner and Gain Scheduler

The intelligent tuner is a software package that runs in the distributed control systems Provox™ and RS3™. It is interesting to note that the same software runs on two different systems. Tuning is done on demand from the operator. There are good facilities for an

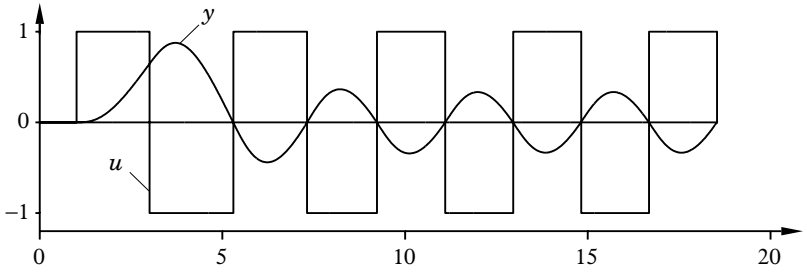

Figure 6.9 Input (u) and output (y) signals during an experiment with relay feedback.

instrument engineer to influence both the tuning experiment and the control design. Information about the process is obtained from an experiment with relay feedback. Several different tuning methods can be chosen.

It is easy for the operator to follow the experiment because the control variable and the process variable are plotted on the console during the experiment. A typical behavior of the signals is illustrated in Figure 6.9. Tuning starts with the process in equilibrium. Then there is an initialization phase in which a step change is made in the control variable. The process output is monitored, and the step is reversed when the process output has changed a specified amount. Relay feedback is initiated when the output equals the setpoint again. The step size is typically 3%, 5%, or 10% of the range of the control variable. There is a default value but a particular value can be chosen from the console. The amplitude of the process variable is typically between 1% and 3% of the signal span.

An hysteresis is normally used to overcome the problem of noisy signals in a relay feedback experiment. This has the drawback of giving too high a value of the ultimate period. A different method that gives a more accurate value of the ultimate period can also be used in the Fisher-Rosemount system. The initialization phase gives information about the dead time of the process. After a switch the relay is prohibited from switching during a time approximately equal to the dead time. The recommendation is to use this only in very noisy environments.

The tuning experiment gives process data in terms of the apparent dead time L, the ultimate gain K_u, and the ultimate period T_u. From these parameters, process static gain K_p, dominating time constant T, and apparent dead time L are calculated and displayed on the screen. In a distributed system there may be communication delays. Special care has been taken to make sure that the delay is short and to account for it when calculating the ultimate gain and the ultimate period.

Tuning is done from a display that can be brought up on the console. The display shows curves; it also has menus and buttons to execute the tuning. The program has the following methods for computing the controller parameters.

PID	Standard PID controller with a design that gives a moderately fast response with small overshoot and a phase margin of about 45°.
PID-60	PID controller with slow response and small overshoot.
P	Proportional controller with Ziegler-Nichols design.
PI	PI control based on Ziegler-Nichols design.
PID-ZN	PID controller based on Ziegler-Nichols design, fast but with considerable overshoot.
PID-45N	PID control for a noisy control loop. The controller has 45° phase margin and less derivative action than the standard PID design.
PID-60N	Similar to PID-45N but with less overshoot.
PI-DT	PI design for processes with dominating dead time, smaller gain and integral time than the PI design.
IMC	Design based on the internal model principle. (See Section 4.5.) The details are not available. This design does not work for processes with integral action.
IMC-NSR	Similar to IMC design for processes with integral action.

There are also recommendations for choosing different control designs. PI control is the primary choice for flow-pressure and level control, while PID is recommended for pH and temperature control. In addition, the closed loop can be modified by selecting SLOW, NORMAL, or FAST. With these choices the controller gain is multiplied with the factors 0.50, 1.00, and 1.25. It is possible to change the design without retuning the process.

After a tuning the new controller parameters are displayed together with the old controller parameters. It is then possible to accept, reject, redesign, or modify the controller parameters.

The intelligent gain scheduler is a complement to the tuner. The scheduling variable is the process variable, the control variable, or an external signal. The range of the scheduling variable is divided into three regions. The values of the controller parameters in each region are stored in a table. Interpolation between the entries is done with a fuzzy scheme. Values from the tuner are entered automatically in the tables. They can also be entered manually.

Typical applications of gain scheduling are pH control, split-range control, level control in vessels with unusual geometries, and surge tank control.

Honeywell Looptune

Looptune is a tuning-program package for the DCS system Honeywell TDC 3000. It offers the following features:

- A search algorithm, which performs incremental improvements of the controller parameters until an optimum is reached.
- A relay feedback technique to tune the controller according to the Ziegler-Nichols frequency response method.
- Gain scheduling, where the schedule contains three sets of controller parameters.

In this presentation, we focus on the search algorithm.

The Objective Function

The control performance is optimized by adjusting the controller parameters in such a way that an objective function (J) is minimized. The objective function is

$$J = (1 - w)\sigma_e^2 + w\sigma_u^2$$

where σ_e^2 is the variance of the control error $e = y_{sp} - y$; σ_u^2 is the variance of the control signal; and w is a weighting factor with range $0 \leq w \leq 1$, specified by the user. (This is the only parameter that has to be specified by the user.) If a small value of w is chosen, the error variance term dominates the objective function. This means that objective function J is minimized if the controller is tuned for tight control. If w is large, the control signal variance term dominates, and objective function J is minimized when the controller is tuned to give a smoother control with smaller control actions.

The objective function is calculated in the following way. Setpoint y_{sp}, process output y, and control signal u are registered during an evaluation period. The variance of the control error (σ_e^2) and the variance of the control signal (σ_u^2) are then calculated in the following way:

$$\sigma_e^2 = \frac{1}{n}\sum_{i=1}^{n}(y_{sp}(i) - y(i))^2$$

$$\sigma_u^2 = \frac{1}{n}\sum_{i=1}^{n}(u(i) - \bar{u})^2$$

where n is the number of data points and

$$\bar{u} = \frac{1}{n}\sum_{i=1}^{n}u(i)$$

is the mean value of the control signal during the evaluation period.

The Search Procedure

The procedure consists of the following steps:

1. Collect n new values of y_{sp}, y, and u during an evaluation period.
2. Compute the value of the objective function and compare it with the previous value.
3. If the value of the objective function has decreased, the controller parameter is adjusted further in the same direction, otherwise the direction is reversed.
4. If the value of the objective function has reached its minimum, then stop the adjustment. If not, go back to step 1 again.

When a controller parameter has been adjusted according to these rules, the Looptuner goes on to the next controller parameter and starts to adjust it to an optimal value. It is claimed that normally several passes through the tuning constants are needed in order to arrive at the optimal controller parameters.

The evaluation period is a crucial parameter in this tuning procedure. It is determined automatically and it should be as short as possible in order to get a fast tuning procedure. On the other hand, it is important that it is long enough, so that a representative number of disturbances occur within the period. At least eight significant disturbances must occur within the evaluation period. The Looptune also has a rule-base that compensates the objective functions for variations in the disturbance behavior between the different evaluation periods.

The search procedure in the Looptune has two modes of operation:

One-shot tuning. The tuning is deactivated when all controller parameters have reached their optimal values.

Continuous adaptation. The adjustments goes on until it is interrupted by the operator.

ABB DCS Tuner

This DCS Tuner is a software package for on-line and off-line automatic tuning of PID controllers in the ABB Master system. It runs on personal computers and is connected to the ABB Master by a serial link interface. The DCS Tuner has four main functions: process identification, controller tuning, process analysis, and simulation. These four functions are described below.

Process Identification

The process identification can be based on either historical data or on-line data. The on-line data acquisition can be event initiated. This

means that the data used for process identification can be controlled based on a high- or low-level triggering. The triggering signal can be the control error, the process variable, or any other signal logged during the session. The major steps performed during the process model identification are:

1. Signal filering using Butterworth filters
2. Process dead-time estimation
3. Process model order identification
4. Least-squares identification of continuous-time and discrete-time models
5. Process model validation

These steps are normally performed automatically, but the operator may interfere by changing parameters.

Controller Tuning

The tuning procedure uses the discrete-time model to obtain the controller parameters. The tuning method is based on a dominant pole design procedure. The user may select among three different closed-loop performances: Fast, Normal, or Damped.

Process Analysis

Three functions are available for process analysis. They are called Statistics, FFT, and Correlation. The Statistics function provides statistical information about the different data series, such as mean, standard deviation etc. Frequency spectra of the different signals and frequency domain models of the process can be obtained using the Fast Fourier Transform (FFT). Finally, the Correlation function gives results of auto- and cross-correlation analysis.

Simulation

In the simulation mode, both open-loop and closed-loop simulation are available. The user has access to both controller and process parameters. This provides a possibility to check how different controller settings suit a certain process model, and also how changes in the process model influence the control loop performance.

Techmation Protuner

The Protuner is a process analyzer from Techmation Inc. It consists of a software package for personal computers and an interface module with cables to be connected to the process output and the control

signal of the control loop to be analyzed. The Protuner monitors a step-response experiment, calculates the frequency response of the process based on the experimental data, and suggests controller parameters based on several methods for controller tuning.

Prior Information

Before the process analysis is performed, the user must provide some information about the process and the controller. This is done using a couple of "Set-up menus." The following process information must be given:

- The ranges of the control and the measurement signals.
- It must be determined if the process is stable or if it has integral action.

To be able to set relevant controller parameters, the following data about the controller must be provided:

- P-type (gain or proportional band)
- I-type (seconds, seconds/repeat, minutes, or minutes/repeat)
- Controller structure (ideal, series, or parallel)
- Sampling rate
- Filter time constant (if there is a low-pass filter connected to the measurement signal).

Before the tuning experiment can be performed, the user must also specify a sample time. This is the time during which data will be collected during the experiment. It is important to choose the sample time long enough, so that the step response settles before the sample time has ended. In case of an open-loop experiment of an integrating process, the response must reach a constant rate of change when the experiment ends.

Determining the Process Model

The tuning procedure is based on a step-response experiment. It can be performed either in open or closed loop. The open-loop experiment is recommended. When the user gives a start command, the process output and the control signal are displayed on the screen, with a time axis that is given by the sample time defined by the user. The user then makes a step change in the control signal. If the experiment is performed in closed loop a step is instead introduced in the setpoint.

There are several facilities for editing the data obtained from the step-response experiment. Outliers can be removed and data can be filtered. These features are very useful because they make it possible to overcome problems that are often encountered when making experiments on industrial processes.

When the data has been edited the Protuner calculates the frequency response of the process. The result can be displayed in a Bode diagram, a Nyquist diagram, or a Nichols diagram. The static gain, the dominant time constant and the apparent dead time are also displayed, as well as the ultimate gain and the ultimate period.

Design Calculations

The controller parameters are calculated from the frequency response. A special technique is used. This is based on cancellation of process poles by controller zeros. The integral time and the derivative time are first determined to perform this cancellation. The gain is then determined to meet predetermined gain and phase margins.

The Protuner provides several design options. Controller parameters are given for the following closed-loop responses:

Slow: Critically damped response.

Medium: Slightly underdamped response.

Fast: Response with decay ratio 0.38.

The different design options are obtained by specifying different values of the gain and the phase margins. The Protuner provides different controller parameters depending on whether setpoint or load disturbances are considered. Both P, PI, and PID controller parameters are provided. The setpoint weightings for proportional and derivative action and the high frequency gain at the derivative part must be supplied by the user.

Evaluation

It is possible to evaluate the performance of the closed-loop system in several ways. The combined frequency response, i.e., the frequency response of the loop transfer function

$$G_\ell(i\omega) = G_p(i\omega)G_c(i\omega)$$

where $G_p(i\omega)$ is the process transfer function and $G_c(i\omega)$ is the controller transfer function, can be plotted in a Bode diagram, a Nyquist diagram, or a Nichols diagram. In this way, the phase and amplitude margins or the M_s value can be checked.

The Protuner also has a simulation facility. It is possible to simulate the closed-loop response of the process and the suggested controller. To do this, it is necessary to provide some additional controller parameters, namely setpoint weightings b and c, and derivative gain limitation factor N. Using the simulation facility, it is also possible to investigate the effects of noise and to design filters to reduce these effects.

6.7 Integrated Tuning and Diagnosis

It is well known that many control loops in the process industry do not perform satisfactory. Poor controller tuning is one of the major reasons for this, but there are other problems that are not solved by adjusting the controller parameters. Examples are nonlinearities in the valves (stiction, hysteresis, etc) and improperly sized valves and transmitters. It is important to investigate the control loop carefully and to discover these problems before initiating the controller tuning. (Compare with the discussion in Section 6.2.)

Friction in the Valve

A common cause of problems is high friction in the valve. There is, of course, always static friction (stiction) in the valve, but if the valve maintenance is insufficient, the friction may be so large that the control performance degrades. The amount of friction can easily be measured by making small changes in the control signal and checking how the process outputs react. To investigate the valve, it is preferable to use the position of the valve stem as the output. The process output can also be used, but then we are investigating the complete process. The experiment will also take a longer time because of the dynamics involved. The procedure is shown in Figure 6.10. In the figure, the process output only responds to the control signal when the changes in the control signal are large enough to overcome the static friction.

Friction in the valve results in stick-slip motion. This phenomena is shown in Figure 6.11. Because of the static friction, the process output will oscillate around the setpoint. The valve will only move

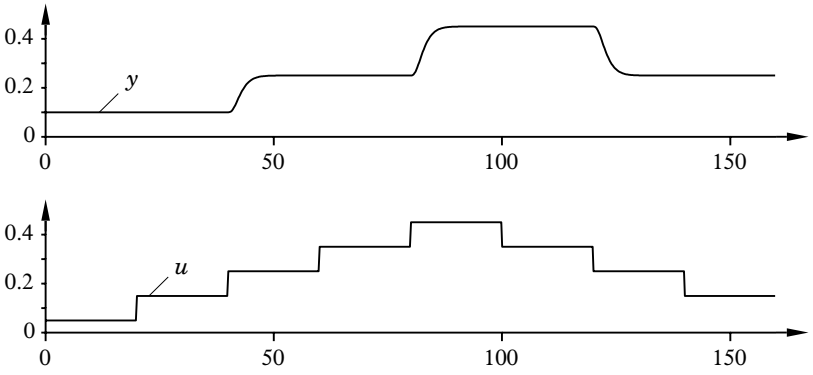

Figure 6.10 Procedure to check the amount of valve friction. The upper diagram shows process output y and the lower diagram shows control signal u.

when the control signal has changed sufficiently since the previous valve movement. When the valve moves, it moves too much. This causes the stick-slip motion. The pattern in Figure 6.11, where the measurement signal is close to a square wave and the control signal is close to a triangular wave, is typical for stick-slip motion.

Many operators detune the controller when they see oscillations like the one in Figure 6.11, since they believe that the oscillations are caused by a bad controller tuning. Unfortunately, most adaptive controllers do the same. What one should do, when a control loop starts to oscillate, is to first determine the cause of the oscillation. A good way to perform this determination is presented in Figure 6.12.

The first problem to determine is whether the oscillations are generated outside the control loop or generated inside the loop. This can be done by disconnecting the feedback, e.g., by switching the controller to manual mode. If the oscillation is still present, the disturbances must be generated outside the loop, otherwise they were generated inside the loop.

If the disturbances are generated inside the loop, the cause can be either friction in the valve or a badly tuned controller. Whether friction is present or not can be determined by making small changes in the control signal and checking if the measurement signal follows, as shown in Figure 6.10. If friction is causing the oscillations, the solution to the problem is valve maintenance.

If the disturbances are generated outside the control loop, one should try, of course, to find the source of the disturbances and try to eliminate it. This is not always possible, even if the source is found. One can then try to feed the disturbances forward to the controller, and in this way reduce their effect on the actual control loop.

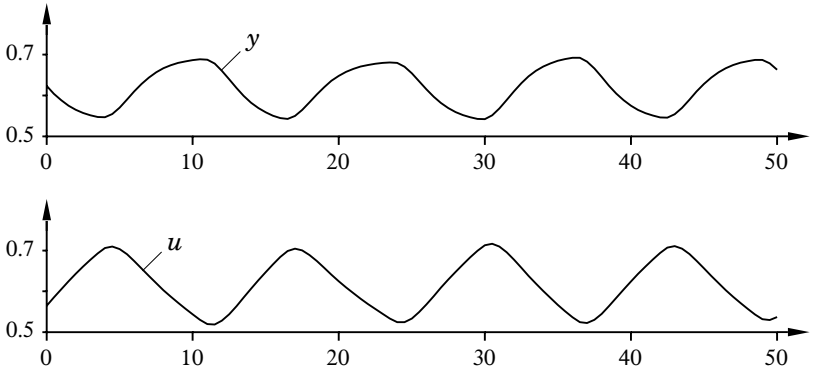

Figure 6.11 Stick-slip motion caused by friction in the valve. The upper diagram shows process output y and the lower diagram shows control signal u.

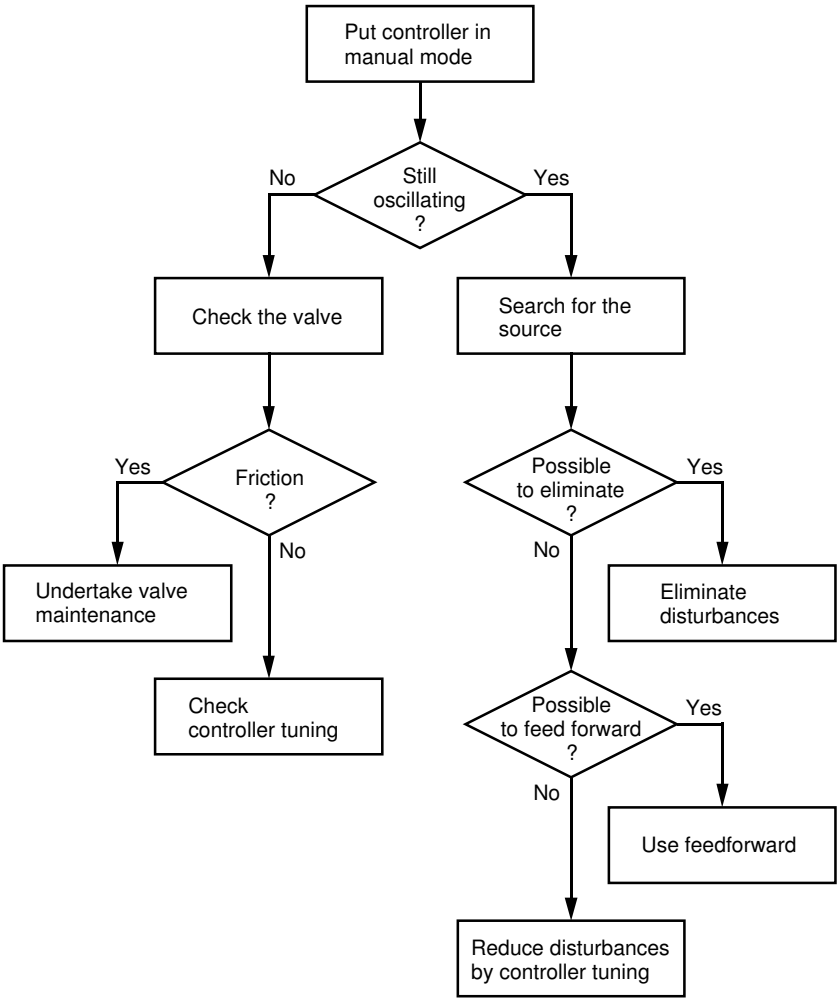

Figure 6.12 Diagnosis procedure to discover the cause of oscillations, and recommended actions to eliminate them.

Hysteresis in the Valve

Because of wear, there is often hysteresis (backlash) in the valve or actuator. The amount of hysteresis can be measured as shown in Figure 6.13. The experiment starts with two step changes in the control signal in the same direction. The hysteresis gap will close if the first step is sufficiently large. This means that the second step is performed without hysteresis. The third step is then made in the opposite direction. The control signal will then pass the whole gap before the valve moves. If the last two steps are of the same size,

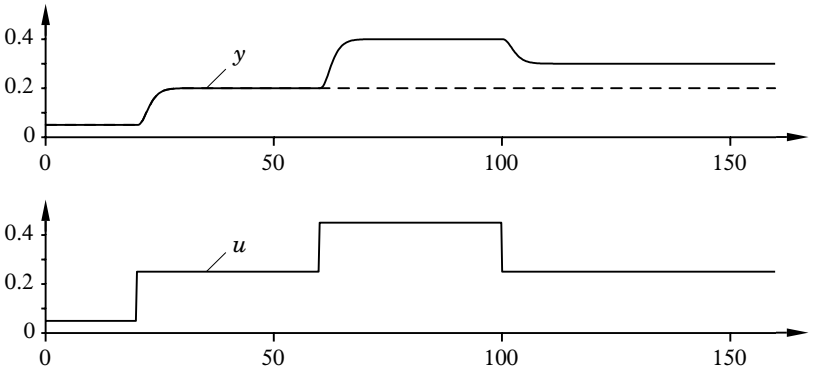

Figure 6.13 Procedure to check the amount of valve hysteresis. The upper diagram shows process output y and the lower diagram shows control signal u.

we can calculate the hysteresis as $\Delta y/K_p$, where Δy is the difference between the process outputs after the first and the third step, see Figure 6.13, and K_p is the static process gain (also easily obtained from Figure 6.13). If the control signal is ramped (or moved in small steps) upwards, and then downwards again, we obtain the result in Figure 6.14. Here, the hysteresis can easily be determined as the horizontal distance between the two lines.

Figure 6.15 shows closed-loop control of a process with large hysteresis in the valve. The control signal has to travel through the gap in order to move the valve. Therefore, we get the typical linear drifts in the control signal as shown in Figure 6.15.

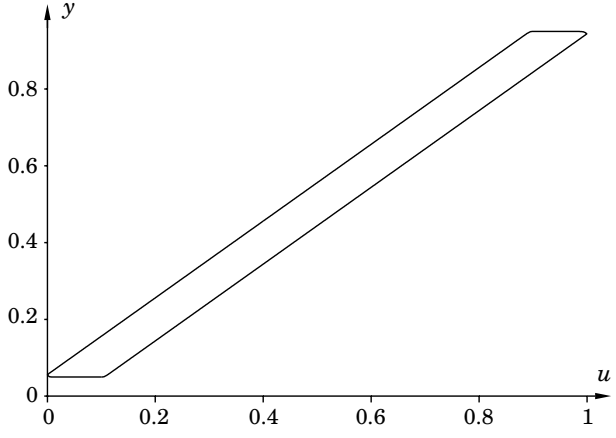

Figure 6.14 Characteristic of a valve with hysteresis. The diagram shows process output y as function of control signal u.

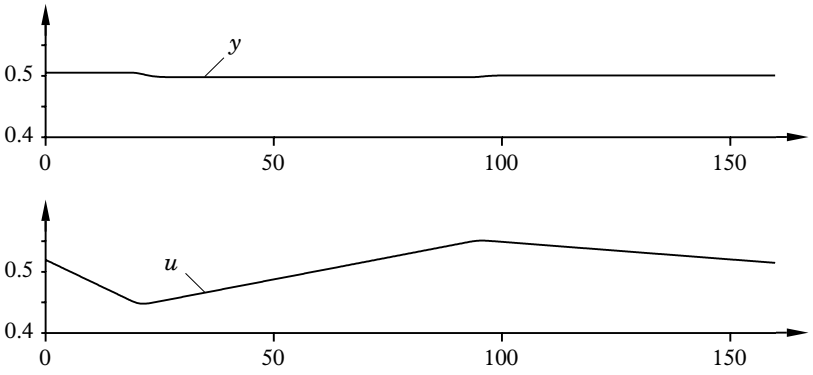

Figure 6.15 Closed-loop control with valve hysteresis. The upper diagram shows process output y and the lower diagram shows control signal u.

If a relay auto-tuner is applied to a process with hysteresis, the estimated process gain will be smaller than the true value. This gives too large a controller gain. An auto-tuner based on a step-response experiment will work properly if the gap is closed before the step-response experiment is performed. (Compare with the second step in Figure 6.13.)

Other Nonlinearities

Even valves with a small static friction and hysteresis often have a nonlinear characteristic. The total characteristic of the process can be obtained by checking the static relation between the control signal

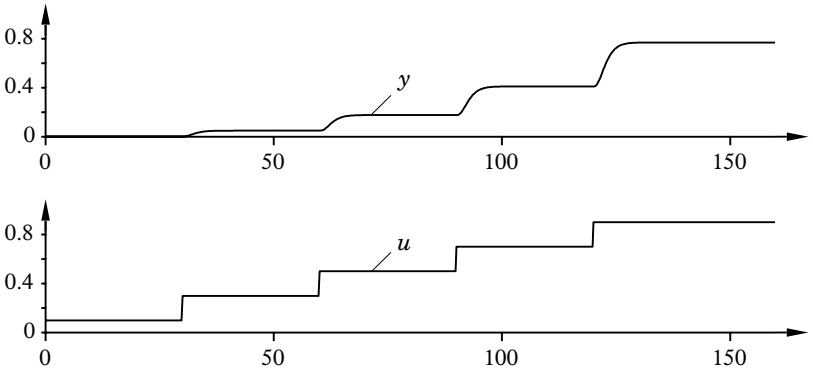

Figure 6.16 A procedure to determine the static process characteristic. The upper diagram shows process output y and the lower diagram shows control signal u.

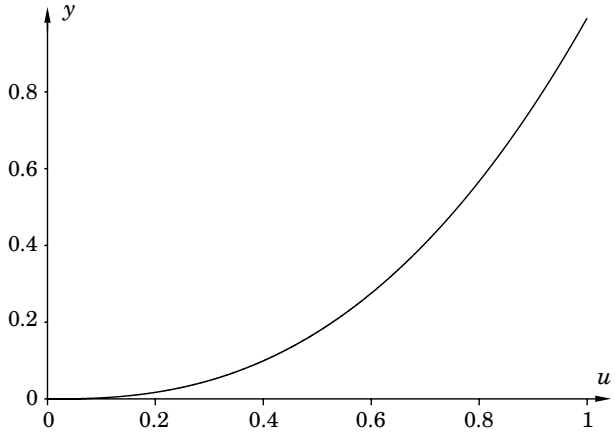

Figure 6.17 The static process characteristic, showing process output y as function of control signal u.

and the measured signal. See Figure 6.16. The characteristic shown in Figure 6.16 is obviously nonlinear. It has a higher gain at larger valve positions. If the stationary values of the measured signal are plotted against the control signal, we obtain the static process characteristic. See Figure 6.17. A plot like this reveals whether gain scheduling is suitable or not.

A nonlinear relation between the control signal and the measurement signal can be obtained for reasons other than nonlinearities in the valve. For example, there might be nonlinearities in the sensor or transmitter. As pointed out in Section 6.3, it is important to understand the cause of the nonlinearity in order to determine a suitable gain-scheduling reference.

Noise

Another important issue to consider before tuning the controller is the disturbances acting on the control loop. We have pointed out that it is important to know if the major disturbances are setpoint changes (the servo problem) or load disturbances (the regulator problem).

It is also important to investigate the level of the measurement noise and its frequency content. (Compare with Section 2.8.) If the noise level is high, it might be necessary to filter the measurement signal before it enters the control algorithm. This is an easy way to get rid of high-frequency noise. If there are disturbances with a large frequency content near the ultimate frequency, it is not possible to use low-pass filtering to remove them. Feedforward is one possibility, if the disturbances can be measured at their source. Notch filters can

be used if the noise is concentrated to a narrow frequency range. See Section 2.8 where noise modeling and measurements were discussed.

Sampling Rates and Prefilters

Selection of sampling rates and the associated prefilter are important in all digital controllers. For single-loop controllers is is customary to choose a constant sampling rate, often between 0.1 s and 1 s. Faster rates are introduced when permitted by the processor speed. Distributed control systems have somewhat greater flexibility.

The sampling rate should of course be chosen based on the bandwidth of the control loop. In process control systems, sampling rates have as a rule been chosen routinely without the proper considerations of these issues. The reason for this is simply that there is not much one can do when the bandwidth of the control loop is not known. There are many new possibilities in this area when auto-tuning is used. After a tuning it is possible to choose the sampling rate and the prefilter in a rational way. The control quality can often be increased significantly by such a procedure. The prefilter should be matched to the sampling rate. This can easily be achieved by using dual sampling rates. The process variable is filtered and sampled with a fixed, fast sampling rate. Digital filtering with a variable bandwidth is then applied and the filtered signal is sampled at the rate appropriate for the control loop.

On-Line Detection

We have shown some tests that can be performed manually by the operator in order to ensure that the control loop is properly designed. The problems mentioned can also show up after a while, when the control loop is in automatic mode. On-line detection procedures are, therefore, of interest, and the research in this area has gained a lot of interest in recent years. Most controllers have a primitive form of diagnosis in the use of alarms on limits on the measured signals. The operator thus gets an alarm when signals exceed certain specified alarm limits. More sophisticated detection procedures, where alarms are given when problems like those mentioned above arise, will be available in industrial products within the next few years.

A common approach to fault detection is shown in Figure 6.18. If a model of the process is available, the control signal can be fed to the input of the process model. By comparing the output of the model with the true process output, one can detect when the process dynamics change. If the model is good, the difference between the model output and the process output (e) is small. If the process dynamics change,

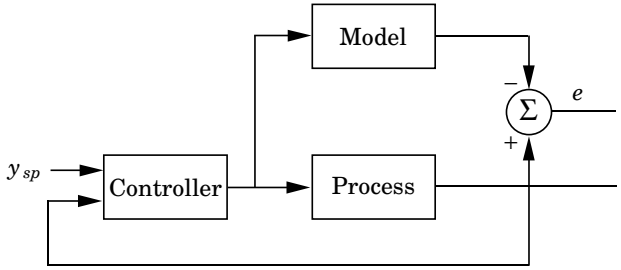

Figure 6.18 Model-based fault detection.

e will no longer be small, since the two responses to the control signal are different. It is also possible to compare other signals in the process and the model rather than the output signals. These fault detection methods are called observer-based methods.

Another fault detection approach is to use a recursive parameter estimator in the same way as the model-based continuous adaptive controller, and to base the detection on the changes in the parameter estimates. These methods are called identification-based methods.

Integrated Tuning and Diagnosis

The diagnosis procedures are related to the adaptive techniques in several ways. We have pointed out the importance of checking valves before applying an automatic tuning procedure. If not, the automatic tuning procedure will not provide the appropriate controller parameters. For this reason, it would be desirable to have these checks incorporated in the automatic tuning procedures. Such devices are not yet available, and the appropriate checks, therefore, must be made by the operator.

The on-line detection methods are related to the continuous adaptive controller. The adaptive controller monitors the control loop performance and changes the controller parameters, if the process dynamics change. The on-line fault detection procedures also monitor the control-loop performance. They give an alarm instead of changing the controller parameters if the process dynamics change. As an example, in Figure 6.12 we have seen that it is important to determine *why* the performance has changed before actions are taken. Most adaptive controllers applied to a process with stiction will detune the controller, since they interpret the oscillations as caused by a badly tuned controller. Consequently, it is desirable to supply the adaptive controllers with on-line detection methods, so that reasons for bad control-loop performance, other than poor controller tuning, are detected. The lack of these kinds of detection procedures in adaptive

controllers are perhaps the major reason for the relatively few applications of continuous adaptive control available today.

6.8 Conclusions

The adaptive techniques are relatively new. Even though they have been tried industrially for only a few years, there are currently several thousand loops where adaptive techniques are used. This is not a negligible number, but it is still a very small fraction of all control loops in operation. It is clear that auto-tuning is useful. It can certainly help operators and instrument engineers keep the control loops well tuned. The benefits are even larger for more complex loops. For example, the derivative action is often switched off in manually tuned systems because of tuning problems, in spite of the fact that it improves performance.

Concerning the particular method to use, it is too early to draw definite conclusions. There are many different ways to determine process characteristics, many methods for design of PID controllers, and many ways of combining such techniques to create auto-tuners. Judging from the systems that are now on the market, it appears that many different ideas have been successfully implemented. However, some patterns do emerge. It appears that more sophisticated methods require more prior information. This is probably what has led to the introduction of the pre-tune mode, which often has been an afterthought. It would seem that a useful approach to this problem is to combine several different approaches. It also seems very natural to combine adaptive techniques with diagnosis and loop assessment.

6.9 References

Controllers with automatic tuning grew out of research on automatic control. Overviews of adaptive techniques are found in (Dumont, 1986), (Åström, 1987a), and (Bristol, 1970). More detailed treatments are found in the books (Harris and Billings, 1981), (Åström and Wittenmark, 1989), and (Hang *et al.*, 1993b). Overviews of different approaches and different products are found in (Isermann, 1982), (Gawthrop, 1986), (Kaya and Titus, 1988), (Morris, 1987), (Yamamoto, 1991), and (Åström *et al.*, 1993).

Many different approaches are used in the automatic tuners. The systems described in (Nishikawa *et al.*, 1984), (Kraus and Myron, 1984), and (Takatsu *et al.*, 1991) are based on transient response techniques. The paper (Hang and Sin, 1991) is based on cross correlation.

The use of orthonormal series representation of the step response of the system is proposed in (Zervos *et al.*, 1988). The system in (Åström and Hägglund, 1984) uses relay feedback. Other ways to use relay feedback are discussed in (Schei, 1992) and (Leva, 1993). The paper (Voda and Landau, 1995) describes a technique where the methods BO and SO are combined with relay tuning. The hysteresis of the relay is adjusted automatically so that the frequency ω_{135}, where the process has a phase lag at 135°, is determined. The paper (Hang *et al.*, 1993a) discusses effects on load disturbances when relay feedback is used.

Traditional adaptive techniques based on system identification and control design have also been used. Identification is often based on estimation of parameters in a transfer function model. Examples of this approach are in (Hawk, 1983), (Hoopes *et al.*, 1983), (Yarber, 1984a), (Yarber, 1984b), and (Cameron and Seborg, 1983). There are also systems where the controller is updated directly as in (Radke and Isermann, 1987), (Marsik and Strejc, 1989), and (Rad and Gawthrop, 1991).

Gain scheduling is a very powerful technique that was developed in parallel with adaptation. An example demonstrating the benefits of gain scheduling is given in (Whatley and Pott, 1984). The paper (Hägglund and Åström, 1991) describes commercial controllers that combine gain scheduling with automatic tuning and adaptation. Automatic tuning can be particularly useful for start-up, this is discussed in (Hess *et al.*, 1987).

Several schemes are based on pattern recognition and attempts to mimic an experienced operator. Rules in the form of logic or fuzzy logic are often used. Some examples are found in the papers (Bristol, 1967), (Porter *et al.*, 1987), (Anderson *et al.*, 1988), (Klein *et al.*, 1991), (Pagano, 1991), and (Swiniarski, 1991).

The information about the commercial systems is very uneven. Some systems are described in detail in journal publications. Other systems are only described in manuals and other material from the manufacturer of the devices. Several tuning aids are implemented in hand-held computers or as software in PCs where the user is entering the process information through a keyboard. Some examples are (Blickley, 1988), (Tyreus, 1987), and (Yamamoto, 1991). A brief presentation of the PIDWIZ system, which is based on a hand-held calculator, is given in (Blickley, 1988).

There is much information about the Foxboro EXACT controller. It is based on early work by Bristol on pattern recognition see, e.g., (Bristol, 1967), (Bristol, 1970), (Bristol *et al.*, 1970), (Bristol, 1977), (Bristol and Kraus, 1984), and (Bristol, 1986). The product is described in (Kraus and Myron, 1984), and operational experience is presented in (Higham, 1985) and (Callaghan *et al.*, 1986).

The systems based on relay feedback are also well documented. The principles are presented in (Åström and Hägglund, 1984). Many details about implementation and applications are given in (Åström and Hägglund, 1988) and (Hägglund and Åström, 1991). The controllers of this type now include automatic tuning, gain scheduling, and continuous adaptation of feedback and feedforward gains.

The papers (McMillan *et al.*, 1993b) and (McMillan *et al.*, 1993a) describe the Fisher Rosemount products for tuning and gain scheduling. The Yokogawa systems are discussed in (Takatsu *et al.*, 1991) and (Yamamoto, 1991).

We have not found any journal articles describing the Protuner and Honeywell's tuners. The interested reader is recommended to contact the companies directly.

There have been comparisons of different auto-tuners and adaptive controllers, but few results from those studies have reached the public domain. Some papers that deal with the issue are (Nachtigal, 1986a), (Nachtigal, 1986b), (Dumont, 1986), (Dumont *et al.*, 1989).

Fault detection and isolation is discussed in (Frank, 1990), (Isermann, 1984), (Isermann *et al.*, 1990), and (Patton *et al.*, 1989). The paper (Hägglund, 1993) describes a fault detection technique that has been incorporated in a commercial controller. More elaborate controllers that combine control and diagnosis are discussed in (Antsaklis *et al.*, 1991) and (Åström, 1992).

Control Paradigms

7.1 Introduction

So far we have only discussed simple control problems with one control variable and one measured signal. Typical process control systems can be much more complex with many control variables and many measured signals. The bottom-up approach is one way to design such systems. In this procedure the system is built up from simple components. The systems can be implemented in many different ways. Originally it was done by interconnection of separate boxes built of pneumatic or electronic components. Today the systems are typically implemented in distributed control systems consisting of several hierarchically connected computers. The software for the distributed control system is typically constructed so that programming can be done by selecting and interconnecting the components. The key component, the PID controller, has already been discussed in detail. In Section 3.5 we showed that integrator windup could be avoided by introducing nonlinearities in the PID controller. In Chapter 6 it was demonstrated that controllers could be tuned automatically, also that the changes in system behavior could be dealt with by gain scheduling and adaptation. In this chapter, we present some of the other components required to build complex automation systems. We also present some of the key paradigms that guide the construction of complex systems.

A collection of paradigms for control are used to build complex systems from simple components. The components are controllers of the PID type, linear filters, and static nonlinearities. Typical nonlinearities are amplitude and rate limiters and signal selectors. Feedback is an important paradigm. Simple feedback loops are used to keep process variables constant or to make them change in specified ways. (Feedback has been discussed extensively in the previous chapters.) The key problem is to determine the control variables that should be chosen to control given process variables. Another problem is that there may be interaction between different feedback loops. In this chapter we discuss other paradigms for control. Cascade control

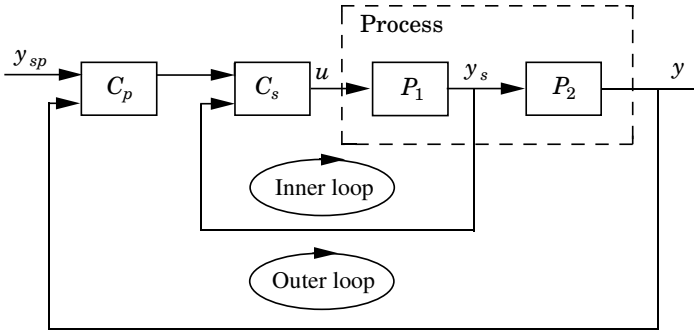

Figure 7.1 Block diagram of a system with cascade control.

is one way to use several measured signals in a feedback loop. (See Section 7.2.) Feedback is reactive in the sense that there must be an error before control actions are taken. Feedforward is another control concept that is proactive because control actions are taken before the disturbance has generated any errors. Feedforward control is discussed in Section 7.3. Model following is a control concept that makes it possible for a system to respond in a specified way to command signals. Section 7.4 presents this paradigm, which also can be combined very effectively with feedback and feedforward. Difficulties may arise when several feedback loops are used. In Section 7.5 we describe some nonlinear elements and some associated paradigms: surge tank control, ratio control, split range control, and selector control. In Sections 7.6 and 7.7 we discuss neural and fuzzy control. These methods can be viewed as special versions of nonlinear control. In Section 7.8 we discuss some difficulties that may arize in interconnected systems. Section 7.9 uses an example to illustrate how the different components and paradigms can be used. Some important observations made in the chapter are summarized in Section 7.10.

7.2 Cascade Control

Cascade control can be used when there are several measurement signals and one control variable. It is particularly useful when there are significant dynamics, e.g., long dead times or long time constants, between the control variable and the process variable. Tighter control can then be achieved by using an intermediate measured signal that responds faster to the control signal. Cascade control is built up by nesting the control loops, as shown in the block diagram in Figure 7.1. The system in this figure has two loops. The inner loop is called *the secondary loop*; the outer loop is called *the primary loop*. The reason

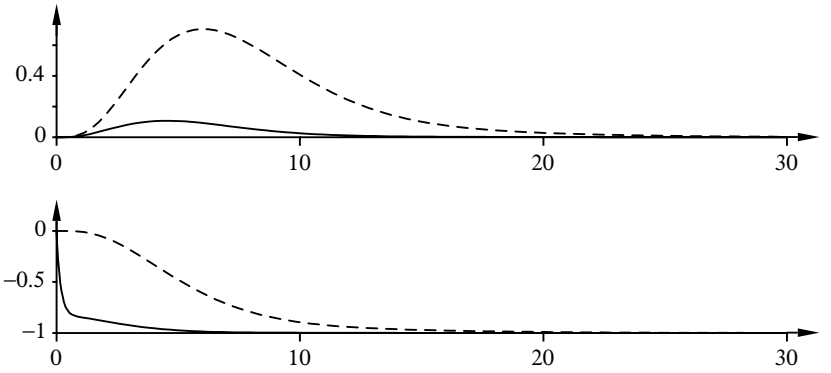

Figure 7.2 Responses to a load disturbance for a system with (full line) and without (dashed line) cascade control. The upper diagram shows process output y and the lower diagram shows control signal u.

for this terminology is that the outer loop deals with the primary measured signal. It is also possible to have a cascade control with more nested loops. The performance of a system can be improved with a number of measured signals, up to a certain limit. If all state variables are measured, it is often not worthwhile to introduce other measured variables. In such a case the cascade control is the same as state feedback. We will illustrate the benefits of cascade control by an example.

EXAMPLE 7.1 Improved load disturbance rejection

Consider the system shown in Figure 7.1. Let the transfer functions be

$$G_{p1} = \frac{1}{s+1}$$

and

$$G_{p2} = \frac{1}{(s+1)^3}$$

Assume that a load disturbance enters at the input of the process. There is significant dynamics from the control variable to the primary output. The secondary output does respond much faster than the primary output. Thus, cascade control can be expected to give improvements. With conventional feedback, it is reasonable to use a PI controller with the parameters $K = 0.37$ and $T_i = 2.2$. These parameters are obtained from the simple tuning rules presented in Chapter 5. The response of the system to a step change in the load disturbance is shown in Figure 7.2.

Since the response of the secondary measured variable to the control signal is quite fast, it is possible to use high loop gains in the secondary loop. If the controller in the inner loop is proportional with gain K_s, the dynamics from the setpoint of C_s to process output becomes

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

This is faster than the open loop dynamics, and higher controller gains can be used in the outer loop. With $K_s = 5$ in the inner loop and PI control with $K = 0.55$ and $T_i = 1.9$ in the outer loop, the responses shown in Figure 7.2 are obtained. The PI controller parameters are obtained from the simple tuning rules presented in Chapter 5. The figure shows that the disturbance response is improved substantially by using cascade control. Notice in particular that the control variable drops very much faster with cascade control. The main reason for this is the fast inner feedback loop, which detects the disturbance much faster than the outer loop.

The secondary controller is proportional and the loop gain is 5. A large part of the disturbance is eliminated by the inner loop. The remaining error is eliminated at a slower rate through the action of the outer loop. In this case integral action in the inner loop will always give an overshoot in the disturbance response. □

Choice of Secondary Measured Variables

It is important to be able to judge whether cascade control can give improvement and to have a methodology for choosing the secondary measured variable. This is easy to do if we just remember that the key idea of cascade control is to arrange a tight feedback loop around a disturbance. In the ideal case the secondary loop can be so tight so that the secondary loop is a perfect servo wherein the secondary measured variable responds very quickly to the control signal. The basic rules for selecting the secondary variable are:

- There should be a well-defined relation between the primary and secondary measured variables.
- Essential disturbances should act in the inner loop.
- The inner loop should be faster than the outer loop. The typical rule of thumb is that the average residence times should have a ratio of at least 5.
- It should be possible to have a high gain in the inner loop.

A common situation is that the inner loop is a feedback around an actuator. The reference variable in the inner loop can then represent a physical quantity, like flow, pressure, torque, velocity, etc., while

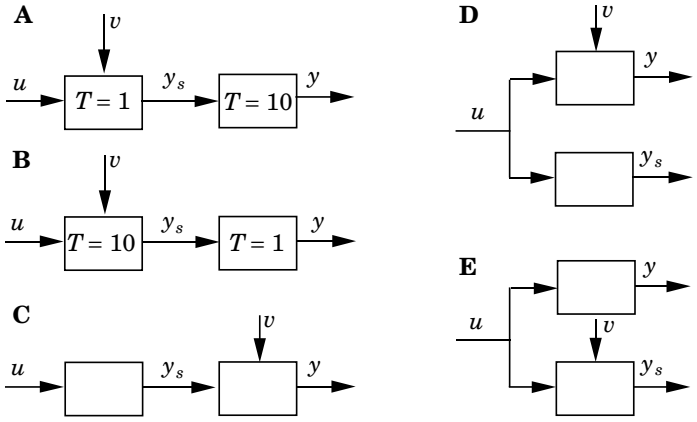

Figure 7.3 Examples of different process and measurement configurations.

the control variable of the inner loop could be valve pressure, control current, etc. This is also a typical example where feedback is used to make a system behave in a simple predictive way. It is also a very good way to linearize nonlinear characteristics.

A number of different control systems with one control variable and two measured signals are shown in Figure 7.3. In the figure the control variable is represented by u, the primary measured variable by y, the secondary measured variable by y_s, and the essential disturbance is v. With the rules given above it is only case A that is suitable for cascade control.

Choice of Control Modes

When the secondary measured signal is chosen it remains to choose the appropriate control modes for the primary and secondary controllers and to tune their parameters. The choice is based on the dynamics of the process and the nature of the disturbances. It is very difficult to give general rules because the conditions can vary significantly. In critical cases it is necessary to analyze and simulate. It is, however, useful to have an intuitive feel for the problems.

Consider the system in Figure 7.1. To have a useful cascade control, it is necessary that the process P_2 be slower than P_1 and that the essential disturbances act on P_1. We assume that these conditions are satisfied. The secondary controller can often be chosen as a pure proportional controller or a PD controller. In some cases integral action can be useful to improve rejection of low-frequency disturbances. With controllers not having integral action, there may be a static er-

ror in the secondary loop. This may not be a serious drawback. The secondary loop, as a rule, is used to eliminate fast disturbances. Slow disturbances can easily be eliminated by the primary loop, which will typically have integral action. There are also drawbacks to using integral control in the secondary loop. With such a system there will always be an overshoot in the response of the primary control loop. Integral action is needed if the process P_2 contains essential time delays and the process P_1 is such that the loop gain in the secondary loop must be limited.

The special case when the process P_2 is a pure integrator is quite common. In this case integral action in the inner loop is equivalent to proportional control in the outer loop. If integral action is used in the inner loop, the proportional action in the outer loop must be reduced. This is a significant disadvantage for the performance of the system. A good remedy is to remove the integrator in the inner loop and to increase the gain in the outer loop.

Tuning and Commissioning

Cascade controllers must be tuned in a correct sequence. The outer loop should first be put in manual when the inner loop is tuned. The inner loop should then be put in automatic when tuning the outer loop. The inner loop is often tuned for critical or overcritical damping or equivalently for a small sensitivity (M_s). If this is not done there is little margin for using feedback in the outer loop.

Commissioning of cascade loops also requires some considerations. The following procedure can be used if we start from scratch with both controllers in manual mode.

1. Adjust the setpoint of the secondary controller to the value of the secondary process variable.
2. Set the secondary controller in automatic with internal setpoint selected.
3. Adjust the primary controller so that its setpoint is equal to the process variable and so that its control signal is equal to the setpoint of the secondary controller.
4. Switch the secondary controller to external setpoint.
5. Switch the primary controller to automatic mode.

The steps given above are automated to different degrees in different controllers. If the procedure is not done in the right way there will be switching transients.

Integral Windup

If integral action is used in both the secondary and primary control loops, it is necessary to have a scheme to avoid integral windup. The inner loop can be handled in the ordinary way, but it is not a trivial task to avoid windup in the outer loop. There are three situations that must be covered:

1. The control signal in the inner loop can saturate.
2. The secondary control loop may be switched to internal setpoint.
3. The secondary controller is switched from automatic to manual mode.

The feedback loop, as viewed from the primary controller, is broken in all these cases, and it is necessary to make sure that its integral mode is dealt with properly. This problem is solved automatically in a number of process controllers that have cascade control capabilities, but if we build up the cascade control using two independent controllers, we have to solve the problem ourselves. This requires being able to inject a tracking signal into the primary controller.

If the output signal of the secondary controller is limited, the process variable of the secondary controller should be chosen as the tracking signal in the primary controller. This also needs a digital transfer from the secondary to the primary controller telling it when the tracking is to take place.

In the case where the secondary controller switches to working according to its local setpoint instead of the external one from the primary controller, the local setpoint should be sent back to the primary controller as a tracking signal. In this way one can avoid both integrator windup and jumps in the transition to cascade control.

When the secondary controller switches over to manual control, the process variable from the secondary controller should be sent back to the primary controller as a tracking signal.

Some Applications

Cascade control is a convenient way to use extra measurements to improve control performance. The following examples illustrate some applications.

EXAMPLE 7.2 Valve positioners

Control loops with pneumatic valves is a very common application. In this case the inner loop is a feedback around the valve itself where the valve position is measured. The inner loop reduces the influences of pressure variations and various nonlinearities in the pneumatic system. □

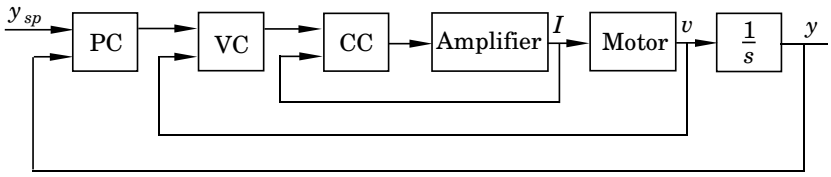

Figure 7.4 Block diagram of a system for position control. The system has three cascaded loops with a current controller (CC) with feedback from current (I), a velocity controller (VC) with feedback from velocity (v), and a position controller (PC) with feedback from position (y).

EXAMPLE 7.3 Motor control

Figure 7.4 is a block diagram of a typical motor control system. This system has three cascaded loops. The innermost loop is a current loop where the current is measured. The next loop is the velocity loop, which is based on measurement of the velocity. The outer loop is a position loop. In this case integral action in the velocity loop is equivalent to proportional action in the position loop. Furthermore, it is clear that the derivative action in the position loop is equivalent with proportional action in the velocity loop. From this it follows directly that there is no reason to introduce integral action in the velocity controller or derivative action in the position controller. □

EXAMPLE 7.4 Heat exchanger

A schematic diagram of a heat exchanger is shown in Figure 7.5. The purpose of the control system is to control the outlet temperature on the secondary side by changing the valve on the primary side. The control system shown uses cascade control. The secondary loop is a flow control system around the valve. The control variable of the primary loop is the setpoint of the flow controller. The effect of nonlinearities in the valve, as well as flow and pressure disturbances, are thus reduced by the secondary controller. □

Observers

Since cascade control can use many measured signals it is natural to ask when it is no longer worthwhile to include an extra signal. An answer to this question has been provided by control theory. The explanation is based on the notion of state of a system. The state of a system is the smallest number of variables, that together with future control signals, describes the future development of a system completely. The number of state variables is, thus, a natural measure of the number of measured signals that are worthwhile to include. If all

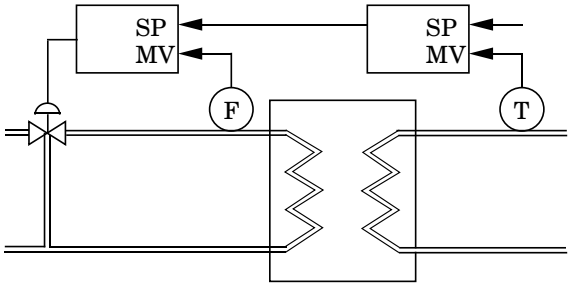

Figure 7.5 Schematic diagram of a heat exchanger with cascade control.

state variables are measured, it is also sufficient to use proportional feedback from these signals. This is called state feedback and can be viewed as a natural extension of cascade control.

Use of observers is another helpful idea from control theory. An observer is based on a mathematical model of a process. It is driven by the control signals to the process and the measured variables. Its output is an estimate of the state of the system. An observer offers the possibility of combining mathematical models with measurements to obtain signals that can not be measured directly. A combination of an observer with a state feedback from the estimated states is a very powerful control strategy.

7.3 Feedforward Control

Disturbances can be eliminated by feedback. With a feedback system it is, however, necessary that there be an error before the controller can take actions to eliminate disturbances. In some situations it is possible to measure disturbances before they have influenced the processes. It is then natural to try to eliminate the effects of the disturbances before they have created control errors. This control paradigm is called *feedforward*. The principle is simply illustrated in Figure 7.6. Feedforward can be used for both linear and nonlinear systems. It requires a mathematical model of the process.

As an illustration we will consider a linear system that has two inputs, the control variable u and the disturbance v, and one output y. The transfer function from disturbance to output is G_v, and the transfer function from the control variable to the output is G_u. The process can be described by the following equation:

$$Y(s) = G_u(s)U(s) + G_v(s)V(s) \tag{7.1}$$

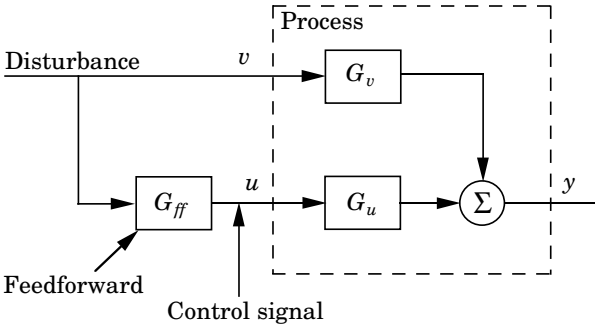

Figure 7.6 Block diagram of a system with feedforward control from a measurable disturbance.

where the Laplace transformed variables are denoted by capitals. The feedforward control law

$$U(s) = -\frac{G_v(s)}{G_u(s)}\, V(s) \tag{7.2}$$

makes the output zero for all disturbances v. The feedforward transfer function thus should be chosen as:

$$G_{ff}(s) = -\frac{G_v(s)}{G_u(s)} \tag{7.3}$$

The feedforward compensator is, in general, a dynamic system. The transfer function G_{ff} must, of course, be stable, which means that G_v must also be stable. If the processes are modeled as static systems, the feedforward compensator is also a static system. This is called *static feedforward*.

If the transfer functions characterizing the process are given by

$$G_u = \frac{K_u}{1 + sT_u} \qquad G_v = \frac{K_v}{1 + sT_v} \tag{7.4}$$

it follows from Equation (7.3) that the feedforward transfer function is

$$G_{ff} = -\frac{K_v}{K_u} \cdot \frac{1 + sT_u}{1 + sT_v} \tag{7.5}$$

In this case the feedforward compensator is a simple dynamic compensator of a lead-lag type.

Since the key idea is to cancel two signals, it is necessary that the model is reasonably accurate. A modeling error of 20% implies that only 80% of the disturbance is eliminated. Modeling errors are directly reflected in control errors. Feedforward is typically much more sensitive to modeling errors than feedback control.

Since it requires process models, feedforward is not used as much as feedback control. There are, however, many cases where a lead-lag filter, as given in Equation (7.5), or even a constant feedforward gives excellent results. The availability of adaptive techniques has drastically increased the range of applicability of feedforward. Certain standard controllers have a feedforward term. Feedforward is also easy to include in distributed control systems.

Feedback and feedforward have complementary properties. With feedback it is possible to reduce the effect of the disturbances with frequencies lower than the system bandwidth. By using feedforward we can also reduce the effects of faster disturbances. Feedback is relatively insensitive to variations in the process model while feedforward, which is used directly in a process model, is more sensitive to parameter variation. Feedback may cause instabilities while feedforward does not give rise to any stability problems. To obtain a good control system, it is desirable to combine feedback and feedforward.

Applications

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

EXAMPLE 7.5 Drum level control

A simplified diagram of a steam boiler is shown in Figure 7.7. The water in the raiser is heated by the burners. The steam generated in the raiser, which is lighter than the water, rises toward the drum.

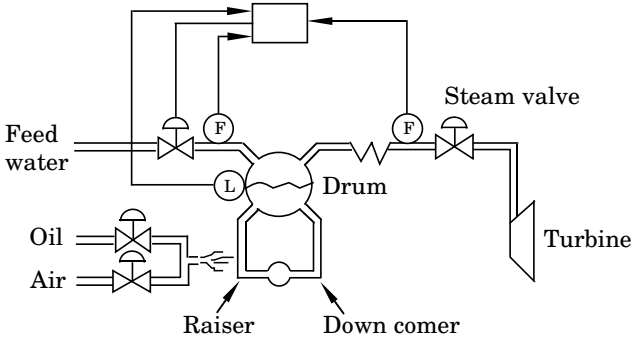

Figure 7.7 Schematic diagram of a drum boiler with level control.

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

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

The problem can be solved by introducing the control strategy shown in Figure 7.7. It consists of a combination of feedback and feedforward. There is a feedback from the drum level to the controller, but there is also a feedforward from the difference between steam flow and feed-water flow so that the feedwater flow is quickly matched to the steam flow. □

7.4 Model Following

When discussing PID-control in Chapter 4 the main emphasis was on load disturbance response. The setpoint response was shaped by setpoint weighting. In some cases it is desirable to have more accurate control of the setpoint response. This can be achieved by using a reference model that gives the desired response to setpoint changes. A simple approach is then to use the scheme shown in Figure 7.8 where the output of the reference model is fed into a simple feedback loop.

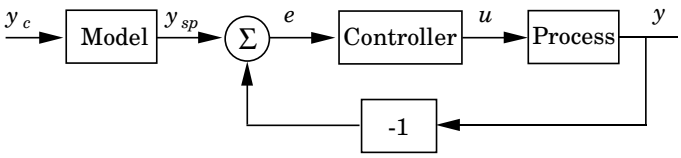

Figure 7.8 Block diagram of a system based on model following.

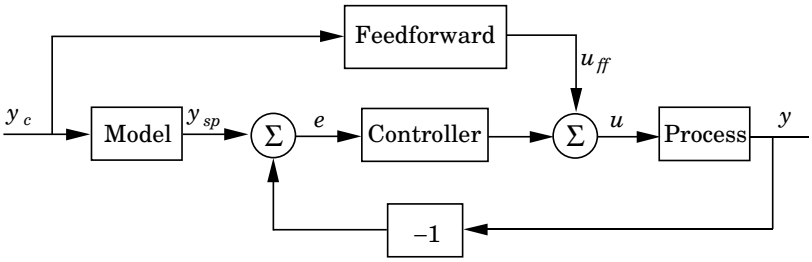

Figure 7.9 Block diagram of a system that combines model following and feedforward from the command signal.

The reference model is typically chosen as a dynamic system of first or second order. In this case we obtain model following by combining a simple controller with a model. It is necessary that the feedback loop be very fast relative to the response of the reference model.

The system can be improved considerably by introducing feedforward as shown in Figure 7.9. In this system we have also feedforward from the command signal. (Compare with Section 7.3.) The signal u_{ff} is such that it will produce the desired output if the models are correct. The error e will differ from zero when the output deviates from its desired behavior. The feedback path will then generate the appropriate actions. When implementing the system the boxes labeled model and feedforward are often combined into one unit which has the command signal y_c as input and y_{sp} and u_{ff} as outputs.

The system is called a two-degree-of-freedom system because the signal paths from setpoint to control and process output to control can be chosen independently. Use of setpoint weighting (see Section 3.4) is one way to obtain this to a small degree. The system in Figure 7.9 is the general version. For such systems it is common to design the feedback so that the system is insensitive to load disturbances and process uncertainties. The model and the feedforward elements are then designed to obtain the desired setpoint response. The feedback controller is often chosen as a PID controller. Model following is used when precise setpoint following is desired, for example, when several control loops have to be coordinated.

A General Controller Structure

The system in Figure 7.9 uses feedforward to improve command signal following. It is possible to combine this with feedforward from measured disturbances as discussed in Section 7.3. We will then obtain the general controller structure shown in Figure 7.10. In this case feedforward is used both to improve setpoint response and to reduce

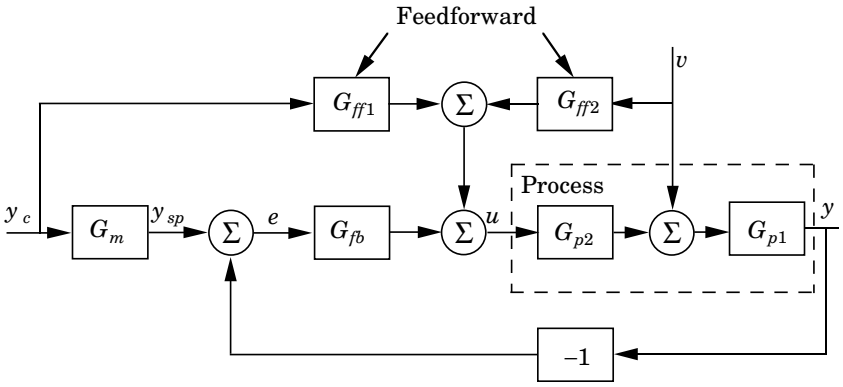

Figure 7.10 Block diagram of a system that combines feedback and feedforward.

the effects of a measurable disturbance.

The properties of the system are analyzed here. If the subsystems are linear and time invariant, we find that the Laplace transform of the control error is given by

$$E(s) = -\frac{G_{p1}(1 + G_{p2}G_{ff2})}{1 + G_pG_{fb}} V(s) + \frac{G_m - G_pG_{ff1}}{1 + G_pG_{fb}} Y_c(s) \tag{7.6}$$

where G_{ff1} and G_{ff2} are transfer functions for feedforwards from the command signal and from the disturbance. G_{fb} is the transfer function for the feedback. The process has the transfer function

$$G_p = G_{p1}G_{p2} \tag{7.7}$$

It follows from Equation (7.6) that the transfer function from command signal to error is given by

$$G(s) = \frac{G_m - G_pG_{ff1}}{1 + G_pG_{fb}} \tag{7.8}$$

This transfer function is small if the loop transfer function $G_\ell = G_pG_{fb}$ is large even without feedforward. The loop transfer function is typically large for frequencies smaller than the servo bandwidth. Therefore, the transfer function G is small for low frequencies even without feedforward. By choosing the feedforward so that

$$G_pG_{ff1} = G_m \tag{7.9}$$

we find that the transfer function becomes zero for all frequencies, irrespective of the feedback used. Since we are combining feedback and feedforward, we can let the feedback handle the low frequencies and use feedforward compensation only to deal with the high frequencies. This means that Equation (7.9) need only be satisfied for higher

frequencies. This makes the feedforward compensator simpler. Similarly it follows from Equation (7.6) that the transfer function from the disturbance to the control error is given by

$$G_v = -\frac{G_{p1}(1 + G_{p2}G_{ff2})}{1 + G_pG_{fb}}$$

In an analogy with the previous discussion, we find that this transfer function will be zero if

$$G_{p2}G_{ff2} = -1$$

holds. The transfer function will also be small if the loop transfer function G_ℓ is large.

These simple calculations illustrate the differences between feedback and feedforward. In particular they show that feedforward reduces disturbances by canceling two terms, while feedback reduces the disturbances by dividing them by a large number. This clearly demonstrates why feedforward is more sensitive than feedback.

Tuning Feedforward Controllers

Feedforward controllers must be well tuned. Unfortunately, it is difficult to tune such controllers. The main difficulty is that it is often not possible to change the disturbances in order to investigate the disturbance response. Therefore, it is necessary to wait for a natural disturbance before the performance of the feedforward can be observed. This makes tuning very time consuming.

7.5 Nonlinear Elements

Nonlinear elements have been discussed before. In Section 3.5 we used a limiter to avoid integral windup, in Section 3.4 we discussed the addition of nonlinearities to obtain "error squared on proportional" and similar control functions. In Chapter 6 it was shown that performance could be improved by gain scheduling. In this section we describe more nonlinear elements and also present some control paradigms that guide the use of these elements.

Limiters

Since all physical values are limited, it is useful to have limiting devices in control systems too. A simple amplitude limiter is shown in Figure 7.11. The limiter can mathematically be described as the

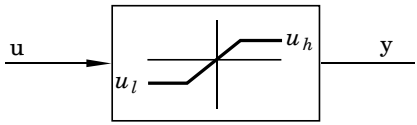

Figure 7.11 Block diagram of a simple amplitude limiter.

static nonlinearity

$$y = \begin{cases} u_\ell & \text{if } u \le u_\ell \\ u & \text{if } u_\ell < u < u_h \\ u_h & \text{if } u \ge u_h \end{cases}$$

It is also useful to limit the rate of change of signals. A system for doing this is shown in Figure 7.12. This circuit is called a *rate limiter* or a *ramp unit*. The output will attempt to follow the input signals. Since there is integral action in the system, the inputs and the outputs will be identical in steady state. Since the output is generated by an integrator with limited input signal, the rate of change of the output will be limited to the bounds given by the limiter. Rate limiters are used, for example, in model-following control of the type shown in Section 7.4. A more sophisticated limiter is shown in Figure 7.13. This limiter is called a *jump and rate limiter*. The output will follow the input for small changes in the input signal. At large changes the output will follow the input with a limited rate. The system in Figure 7.13 can be described by the following equations

$$\frac{dx}{dt} = \text{sat}(u - x)$$
$$y = x + \text{sat}(u - x)$$

where the saturation function is defined as

$$\text{sat}(x) = \begin{cases} -a & x \le -a \\ x & |x| < a \\ a & x \ge a \end{cases}$$

If $|u - x| \le a$ it follows from the equations describing the system that $y = u$, and if $u \ge x + a$ it follows that $dx/dt = a$. Thus, the output

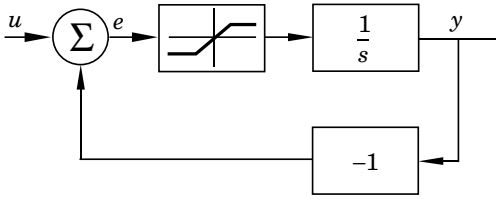

Figure 7.12 Block diagram of a rate limiter or a ramp unit.

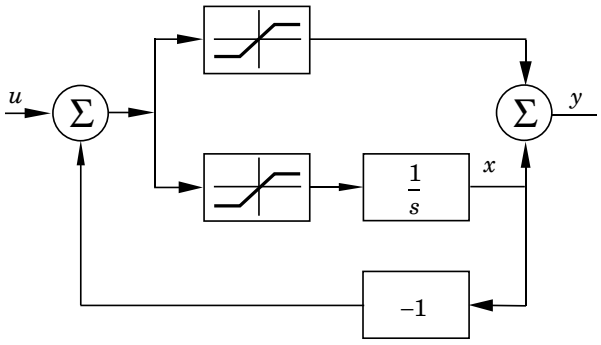

Figure 7.13 Jump and rate limiter.

signal will approach the input signal at the rate a.

Limiters are used in many different ways. They can be used to limit the command signals so that we are not generating setpoints that are demanding faster changes than a system can cope with. In Section 3.5 it was shown how amplitude limiters may be used to avoid integral windup in PID controllers.

Surge Tank Control

The control problems that were discussed in Chapter 4 were all regulation problems where the task was to keep a process variable as close to a given setpoint as possible. There are many other control problems that also are important. Surge tank control is one example. The purpose of a surge tank is to act as a buffer between different production processes. Flow from one process is fed to another via the surge tank. Variations in production rate can be accommodated by letting the level in the surge tank vary. Conventional level control, which attempts to keep the level constant, is clearly not appropriate in this case. To act as a buffer the level should indeed change. It is, however, important that the tank neither becomes empty nor overflow.

There are many approaches to surge tank control. A common, simple solution is to use a proportional controller with a low gain. Controllers with dead zones or nonlinear PI controllers are also used. Gain scheduling is a better method. The scheduling variable is chosen as the tank level. A controller with low gain is chosen when the level is between 10% and 90%, and a controller with high gain is used outside the limits. There are also special schemes for surge tank control.

In many cases there are long sequences of surge tanks and production units, as illustrated in Figure 7.14. Two different control structures, control in the direction of the flow or opposite to the flow, are shown in the figure. Control in the direction opposite to the flow

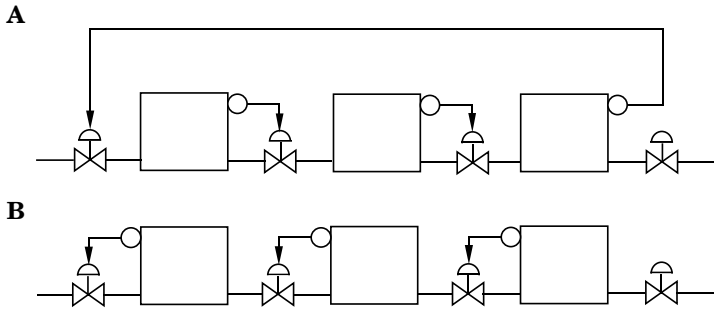

Figure 7.14 Different structures for surge tank control The material flow is from the left to the right. The scheme in A is called control in the direction of the flow. The scheme in B is called control in the direction opposite to the flow.

is superior, because then all control loops are characterized by first-order dynamics. With control in the direction of the flow, it is easy to get oscillations or instabilities because of the feedback from the end of the chain to the beginning.

Ratio Control

When mixing different substances it is desirable to control the proportions of the different media. In combustion control, for example, it is desirable to have a specified ratio of fuel to air. Similar situations occur in many other process control problems. Two possible ways to solve these problems are shown in Figure 7.15. One of the flows, y_k, is controlled in the normal way, and the other flow y is controlled as in Figure 7.15A, where the setpoint is desired ratio a and the measured value is the ratio y/y_k. This arrangement makes the control loop nonlinear, since the gain of the second controller depends on the signal y_k. A better solution is the one shown in Figure 7.15B, where the signal obtained by multiplying y_k by a and adding a bias b is used as the setpoint to a PI controller. The error signal is

$$e = a(y_k + b) - y$$

where a is the desired ratio. If the error is zero it follows that

$$y = ay_k + b$$

Ratio controllers can easily be implemented by combining ordinary PI and PID controllers with devices for adding and multiplying. The control paradigm is so common that they are often combined in one unit called a ratio controller, e.g., a Ratio PI controller (RPI). There are also PID controllers that can operate in ratio mode.

We illustrate ratio control with an example.

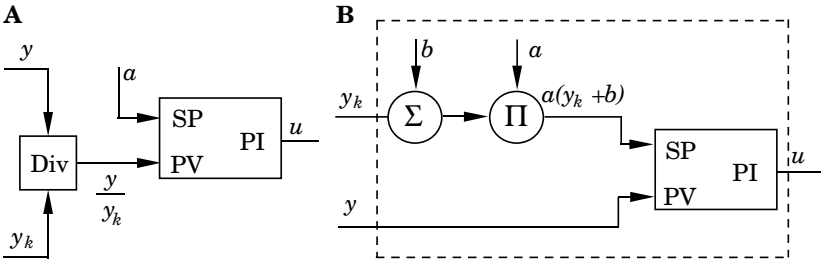

Figure 7.15 Block diagram of two ratio controllers.

EXAMPLE 7.6 Air-fuel control

Operation of a burner requires that the ratio between fuel flow and air flow is kept constant. One control system that achieves this can be constructed from an ordinary PI controller and an RPI controller as is shown in Figure 7.16. The fuel and the air circuits are provided with ordinary flow control. Fuel is controlled by a PI controller, the air flow is controlled with a ratio PI controller where the ratio signal is the fuel flow. The bias term b is used to make sure that there is an air flow even if there is no fuel flow. The system in Figure 7.16 is not symmetric. A consequence of this is that there will be air excess when the setpoint is decreased suddenly, but air deficiency when the setpoint is suddenly increased. □

Split Range Control

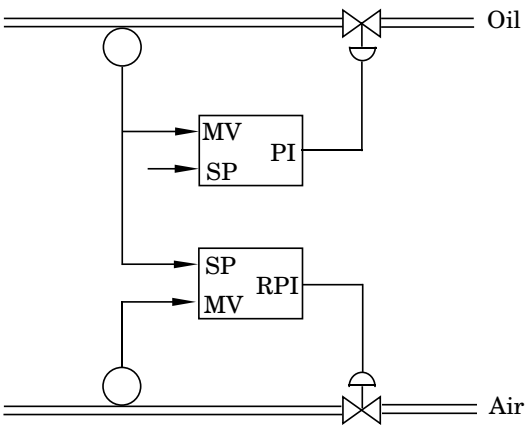

Figure 7.16 Block diagram of an air-fuel controller.

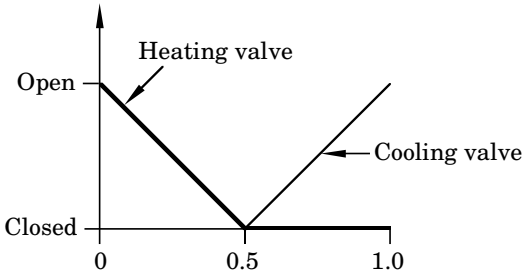

Figure 7.17 Illustration of the concept of split range control.

Cascade control is used when there is one control variable and several measured signals. The dual situation is used when there is one measured variable and several control variables. Systems of this type are common, e.g., in connection with heating and cooling. One physical device is used for heating and another for cooling. The heating and cooling systems often have different static and dynamic characteristics. The principle of split range control is illustrated in Figure 7.17, which shows the static relation between the measured variables and the control variables. When the temperature is too low, it is necessary to supply heat. The heater, therefore, has its maximum value when the measured variable is zero. It then decreases linearly until mid-range, where no heating is supplied. Similarly, there is no cooling when the measured variable is below mid-range. Cooling, however, is applied when the process variable is above mid-range, and it then increases.

There is a critical region when switching from heating to cooling. To avoid both heating and cooling at the same time, there is often a small dead zone where neither heating nor cooling is supplied. Switching between the different control modes may cause difficulties and oscillations.

Split range control is commonly used in systems for heating and ventilation. It is also useful applications when the control variable ranges over a very large range. The flow is then separated into parallel paths each controlled with a valve.

Selector Control

Selector control can be viewed as the inverse of split range control. In split range there is one measured signal and several actuators. In selector control there are many measured signals and only one actuator. A selector is a static device with many inputs and one output. There are two types of selectors: *maximum* and *minimum*. For a maximum selector the output is the largest of the input signals.

There are situations where there are several controlled process variables that must be taken into account. One variable is the primary controlled variable, but it is also required that other process variables remain within given ranges. Selector control can be used to achieve this. The idea is to use several controllers and to have a selector that chooses the controller that is most appropriate. One example of use is where the primary controlled variable is temperature and we must ensure that pressure does not exceed a certain range for safety reasons.

The principle of selector control is illustrated in Figure 7.18. The primary controlled variable is the process output y. There is an auxiliary measured variable z that should be kept within the limits $z_{\min}$ and $z_{\max}$. The primary controller C has process variable y, setpoint y_{sp}, and output u_n. There are also secondary controllers with measured process variables that are the auxiliary variable z and with setpoints that are bounds of the variable z. The outputs of these controllers are u_h and u_l. The controller C is an ordinary PI or PID controller that gives good control under normal circumstances. The output of the minimum selector is the smallest of the input signals; the output of the maximum selector is the largest of the inputs.

Under normal circumstances the auxiliary variable is larger than the minimum value $z_{\min}$ and smaller than the maximum value $z_{\max}$. This means that the output u_h is large and the output u_l is small. The maximum selector, therefore, selects u_n and the minimum selector also selects u_n. The system acts as if the maximum and minimum controller were not present. If the variable z reaches its upper limit, the variable u_h becomes small and is selected by the minimum selector. This means that the control system now attempts to control the

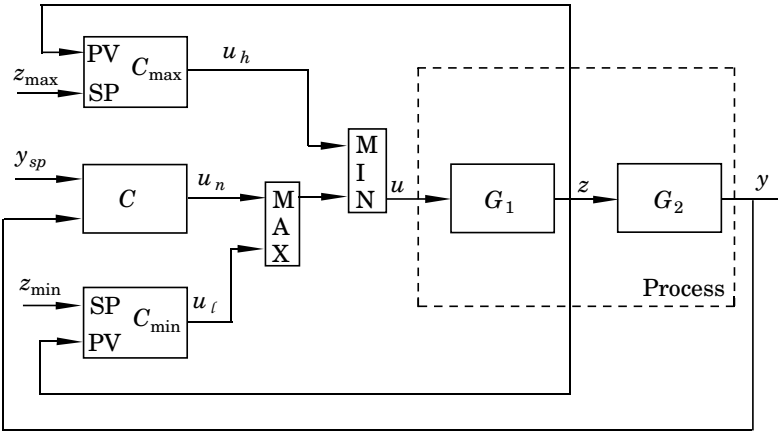

Figure 7.18 Selector control.

variable z and drive it towards its limit. A similar situation occurs if the variable z becomes smaller than $z_{\min}$.

In a system with selectors, only one control loop at a time is in operation. The controllers can be tuned in the same way as single-loop controllers. There may be some difficulties with conditions when the controller switches. With controllers having integral action, it is also necessary to track the integral states of those controllers that are not in operation. Selector control is very common in order to guarantee that variables remain within constraints. The technique is commonly used in the power industry for control in boilers, power systems, and nuclear reactors. The advantage is that it is built up of simple nonlinear components and PI and PID controllers. An alternative to selector control is to make a combination of ordinary controllers and logic. The following example illustrates the use of selector control.

EXAMPLE 7.7 Air-fuel control

In Example 7.6 we discussed air-fuel control. Ratio control has two disadvantages. When the power demand is increased, there may be lack of air because the setpoint of the air controller increases first when the dual controller has increased the oil flow. The system cannot compensate for perturbations in the air channel. A much improved system uses selectors, such as is shown in Figure 7.19. The system uses one minimum and one maximum selector. There is one PI controller for fuel flow and one PI controller for the air flow. The setpoint for the air controller is the larger of the command signal and the fuel flow.

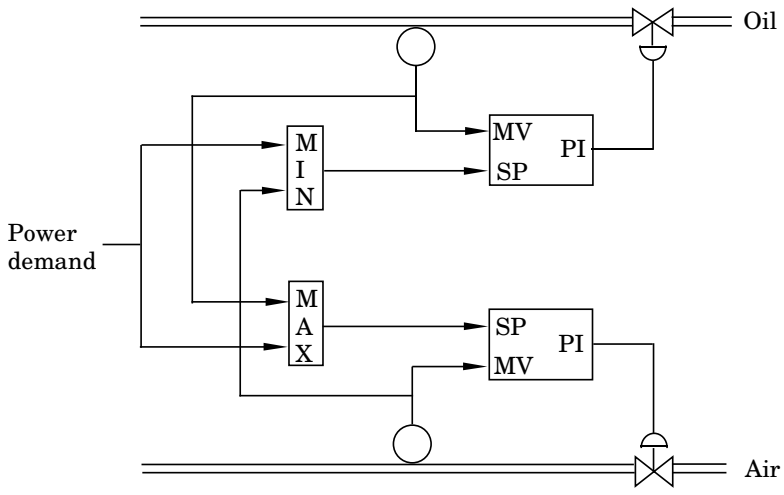

Figure 7.19 Air-fuel controller based on selectors. Compare with the ratio controller for the same system in Figure 7.16.

This means that the air flow will increase as soon as more energy is demanded. Similarly, the setpoint to the fuel flow is the smaller of the demand signal and the air flow. This means that when demand is decreased, the setpoint to the dual flow controller will immediately be decreased, but the setpoint to the air controller will remain high until the oil flow has actually decreased. The system thus ensures that there will always be an excess of air. It is important to maintain good air quality. It is particularly important in ship boilers because captains may pay heavy penalties if there are smoke puffs coming out of the stacks when in port.

Median Selectors

A median selector is a device with many inputs and many outputs. Its output selects the input that represents the current median of the input signals. A special case is the two-out-of-three selector, commonly used for highly sensitive systems. To achieve high reliability it is possible to use redundant sensors and controllers. By inserting median selectors it is possible to have a system that will continue to function even if several components fail.

7.6 Neural Network Control

In the previous section, we have seen that simple nonlinearities can be used very effectively in control systems. In this and the following sections, we will discuss some techniques based on nonlinearities, where the key idea is to use functions of several variables. It is not easy to characterize such functions in a simple way. The ideas described have been introduced under the names of neural and fuzzy control. At first sight these methods may seem quite complicated, but once we strip off the colorful language used, we will find that they are nothing but nonlinear functions.

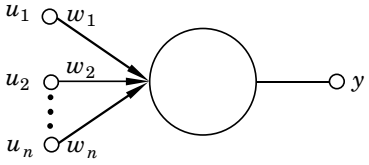

Figure 7.20 Schematic diagram of a simple neuron.

Neural Networks

Neural networks originated in attempts to make simple models for neural activity in the brain and attempts to make devices that could recognize patterns and carry out simple learning tasks. A brief description that captures the essential idea follows.

A Simple Neuron

A schematic diagram of a simple neuron is shown in Figure 7.20. The system has many inputs and one output. If the output is y and the inputs are $u_1, u_2, \ldots, u_n$ the input-output relation is described by

$$y = f(w_1u_1 + w_2u_2 + \ldots + w_nu_n) = f\left(\sum_{k=1}^{n} w_iu_i\right) \tag{7.10}$$

where the numbers w_i are called weights. The function f is a so-called sigmoid function, illustrated in Figure 7.21. Such a function can be represented as

$$f(x) = \frac{e^{\alpha x} - e^{-\alpha x}}{e^{\alpha x} + e^{-\alpha x}} \tag{7.11}$$

where α is a parameter. This model of a neuron is thus simply a nonlinear function. Some special classes of functions can be approximated by Equation (7.10).

Neural Networks

More complicated models can be obtained by connecting neurons together as shown in Figure 7.22. This system is called a neural network or a neural net. The adjective feedforward is often added to indicate that the neurons are connected in a feedforward manner. There are also other types of neural networks. In the feedforward network, the input neurons are connected to a layer of neurons, the outputs of the neurons in the first layer are connected to the neurons in the second

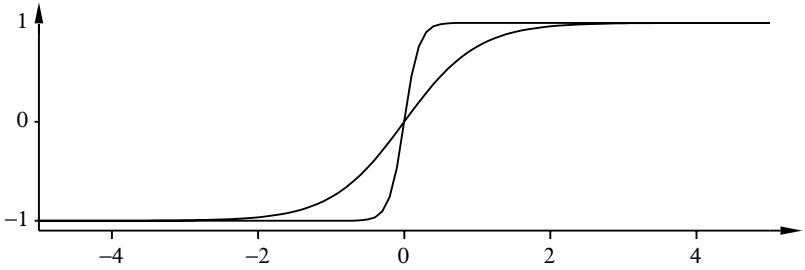

Figure 7.21 Sigmoid functions.

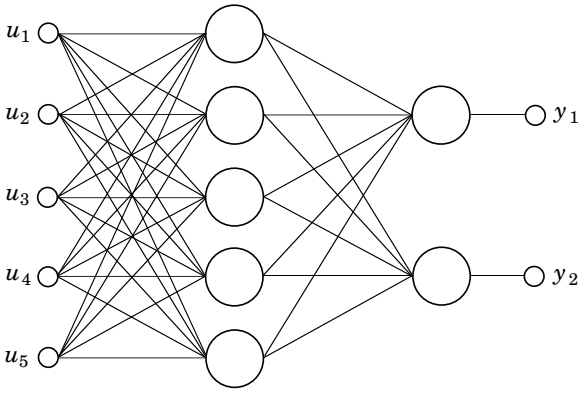

Figure 7.22 A feedforward neural network.

layer, etc., until we have the outputs. The intermediate layers in the net are called hidden layers.

Each neuron is described by Equation (7.10). The input-output relation of a neural net is thus a nonlinear static function. Conversely we can consider a neural net as one way to construct a nonlinear function of several variables. The neural network representation implies that a nonlinear function of several variables is constructed from two components: a single nonlinear function, the sigmoid function (7.11), which is a scalar function of one variable; and linear operations. It is thus a simple way to construct a nonlinearity from simple operations. One reason why neural networks are interesting is that practically all continuous functions can be approximated by neural networks having one hidden layer. It has been found practical to use more hidden layers because then fewer weights can be used.

Learning

Notice that there are many parameters (weights) in a neural network. Assuming that there are n neurons in a layer, if all neurons are connected, n^2 parameters are then required to describe the connections between two layers. Another interesting property of a neural network is that there are so-called learning procedures. This is an algorithm that makes it possible to find parameters (weights) so that the function matches given input-output values. The parameters are typically obtained recursively by giving an input value to the function and the desired output value. The weights are then adjusted so that the data is matched. A new input-output pair is then given and the parameters are adjusted again. The procedure is repeated until a good fit has been obtained for a reasonable data set. This procedure is called training a network. A popular method for training a feedforward net-

work is called back propagation. For this reason the feedforward net is sometimes called a back-propagation network.

Control Applications

A feedforward neural network is nothing but a nonlinear function of several variables with a training procedure. The function has many parameters (weights) that can be adjusted by the training procedure so that the function will match given data. Even if this is an extremely simplistic model of a real neuron, it is a very useful system component. In process control we can often make good use of nonlinear functions. Sensor calibration is one case. There are many situations where an instrument has many different sensors, the outputs of which must be combined nonlinearly to obtain the desired measured value. Nonlinear functions can also be used for pattern recognition.

7.7 Fuzzy Control

Fuzzy control is an old control paradigm that has received a lot of attention recently. In this section we will give a brief description of the key ideas. We will start with fuzzy logic, which has inspired the development.

Fuzzy Logic

Ordinary Boolean logic deals with quantities that are either true or false. Fuzzy logic is an attempt to develop a method for logic reasoning that is less sharp. This is achieved by introducing linguistic variables and associating them with *membership functions*, which take values between 0 and 1. In fuzzy control the logical operations `and`, `or`, and `not` are operations on linguistic variables. These operations can be expressed in terms of operations on the membership functions of the linguistic variables. Consider two linguistic variables with the membership functions $f_A(x)$ and $f_B(x)$. The logical operations are defined by the following operations on the membership functions.

$$\begin{aligned} f_{A \texttt{ and } B} &= \min\left(f_A(x), f_B(x)\right) \\ f_{A \texttt{ or } B} &= \max\left(f_A(x), f_B(x)\right) \\ f_{\texttt{not } A} &= 1 - f_A(x) \end{aligned}$$

A linguistic variable, where the membership function is zero everywhere except for one particular value, is called a crisp variable.

Assume for example that we want to reason about temperature. For this purpose we introduce the linguistic variables *cold*, *moderate*,

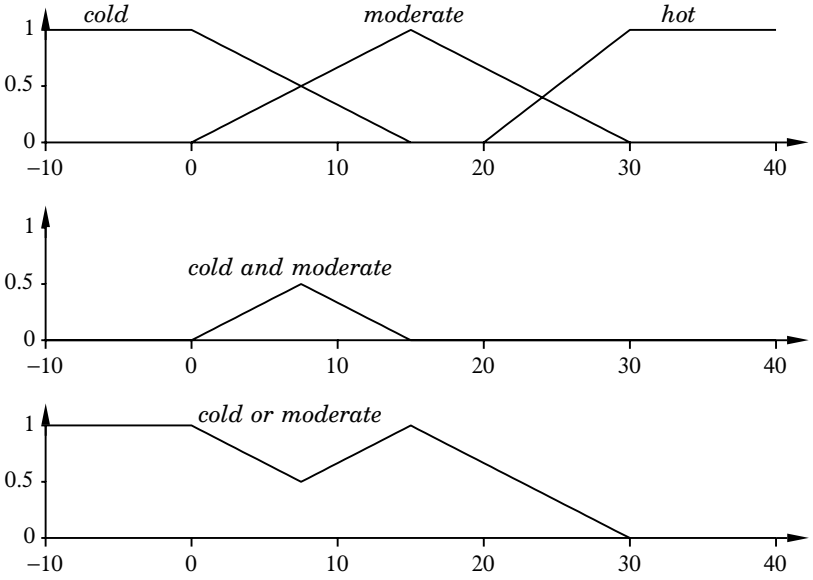

Figure 7.23 Illustration of fuzzy logic. The upper diagram shows the membership functions of *cold*, *moderate*, and *hot*. The middle diagram shows the membership functions for *cold and moderate* the lower diagram shows the membership functions for *cold or moderate*.

and *hot*, and we associate them with the membership functions shown in Figure 7.23. The membership function for the linguistic variables *cold and moderate* and *cold or moderate* are also shown in the figure.

A Fuzzy Controller

A block diagram of a fuzzy PD controller is shown in Figure 7.24. The control error, which is a continuous signal, is fed to a linear system that generates the derivative of the error. The error and its derivative are converted to so-called "linguistic variables" in a process called "fuzzification." This procedure converts continuous variables to a collection of linguistic variables. The number of linguistic variables is typically quite small, for example: negative large (NL), negative medium (NM), negative small (NS), zero (Z), positive small (PS), positive medium (PM), and positive large (PL). The control strategy is expressed in terms of a function that maps linguistic variables to linguistic variables. This function is defined in terms of a set of rules expressed in fuzzy logic. As an illustration we give the rules for a PD controller where the error and its derivative are each characterized by three linguistic variables (N, Z, P) and the control variable is

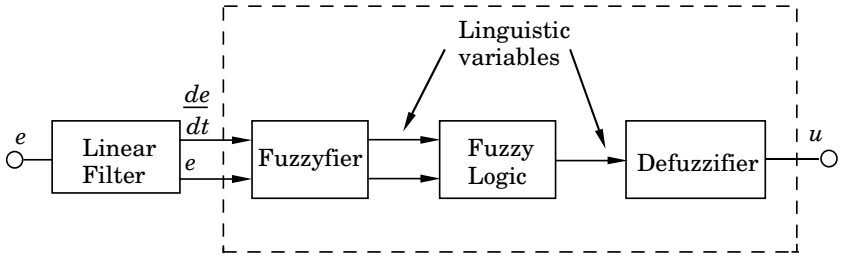

Figure 7.24 A fuzzy PD controller.

characterized by five linguistic variables (*NL*, *NM*, *Z*, *PM*, and *PL*).

```
Rule 1: If e is N and de/dt is P then u is Z
Rule 2: If e is N and de/dt is Z then u is NM
Rule 3: If e is N and de/dt is N then u is NL
Rule 4: If e is Z and de/dt is P then u is PM
Rule 5: If e is Z and de/dt is Z then u is Z
Rule 6: If e is Z and de/dt is N then u is NM
Rule 7: If e is P and de/dt is P then u is PL
Rule 8: If e is P and de/dt is Z then u is PM
Rule 9: If e is P and de/dt is N then u is Z
```

These rules can also be expressed in table form, see Table 7.1.

The membership functions representing the linguistic variables normally overlap (see Figure 7.23). Due to this, several rules contribute to the control signal. The linguistic variable representing the control signal is calculated as a weighted sum of the linguistic variables of the control signal. The linguistic variable representing the control signal is then mapped into a real number by an operation called "defuzzification." More details are given in the following.

Fuzzy Inference

Many different shapes of membership functions can be used. In fuzzy control it is common practice to use overlapping triangular shapes like the ones shown in Figure 7.23 for both inputs and control variables. Typically only a few membership functions are used for the measured variables.

Fuzzy logic is only used to a moderate extent in fuzzy control. A key issue is to interprete logic expressions of the type that appears in the description of the fuzzy controller. Some special methods are used in fuzzy control. To describe these we assume that f_A, f_B, and f_C are the membership functions associated with the linguistic variables A, B, and C. Furthermore let x and y represent measurements. If the values x_0 and y_0 are measured, they are considered as crisp values.

Table 7.1 Representation of the fuzzy PD controller as a table.

		$\frac{de}{dt}$		
		P	Z	N
e	N	Z	NM	NL
	Z	PM	Z	NM
	P	PL	PM	Z

The fuzzy statement

```
If x is A and y is B
```

is then interpreted as the crisp variable

$$z^0 = min(f_A(x^0), f_B(y_0))$$

where and is equivalent to minimization of the membership functions. The linguistic variable u defined by

```
If x is A or y is B then u is C
```

is interpreted as a linguistic variable with the membership function

$$f_u(x) = z^0 f_C(x)$$

If there are several rules, as in the description of the PD controller, each rule is evaluated individually. The results obtained for each rule are combined using the or operator. This corresponds to taking the maximum of the membership functions obtained for each individual rule.

Figure 7.25 is a graphical illustration for the case of the first two rules of the PD controller. The figure shows how the linguistic variable corresponding to each rule is constructed and how the control signal is obtained by taking the maximum of the membership functions obtained from all rules.

The inference procedure described is called “product-max.” This refers to the operations on the membership functions. Other inference procedures are also used in fuzzy control. The and operation is sometimes represented by taking the product of two membership functions and the or operator by taking a saturated sum. Combinations of the schemes are also used. In this way it is possible to obtain “product-max” and “min-sum” inference.

Defuzzification

Fuzzy inference results in a control variable expressed as a linguistic variable and defined by its membership function. To apply a

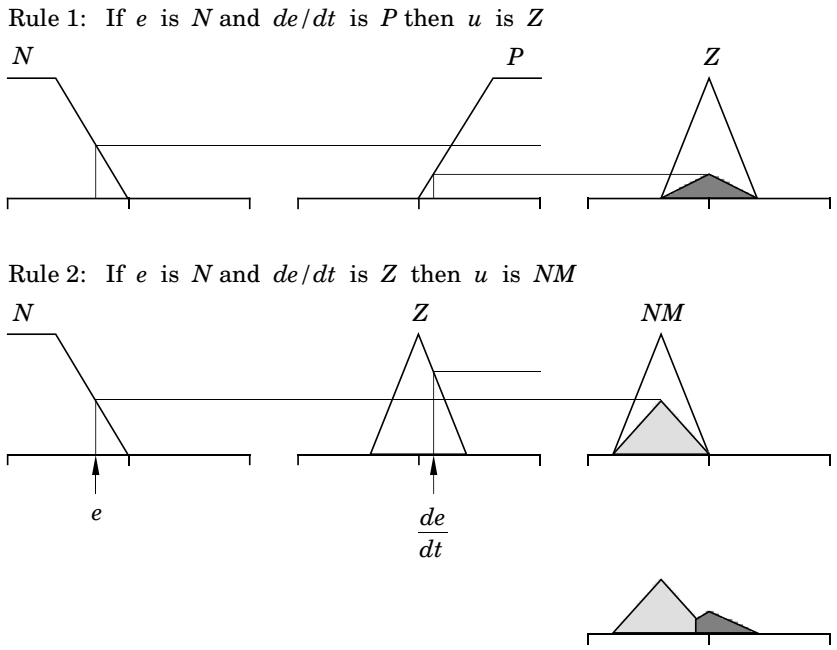

Figure 7.25 Illustration of fuzzy inference with two rules using the min-max rule.

control signal we must have a real variable. Thus, the linguistic variable defining the control signal must be converted to a real number through the operation of "defuzzification." This can be done in several different ways. Consider a linguistic variable A with the membership function $f_A(x)$. Defuzzification by mean values gives the value

$$x_0 = \frac{\int x f_A(x) dx}{\int f_A(x) dx}$$

Defuzzification by the centroid gives a real variable x_0 that satisfies

$$\int_{-\infty}^{x_0} f_A(x) dx = \int_{x_0}^{\infty} f_A(x) dx$$

Nonlinear Control

Having gone through the details, we return to the fuzzy PD controller in Figure 7.24. We first notice that the operations fuzzification, fuzzy logic, and defuzzification can be described in a very simple way. Stripping away the vocabulary and considering the final result, a fuzzy controller is nothing but a nonlinear controller. The system in Figure 7.24

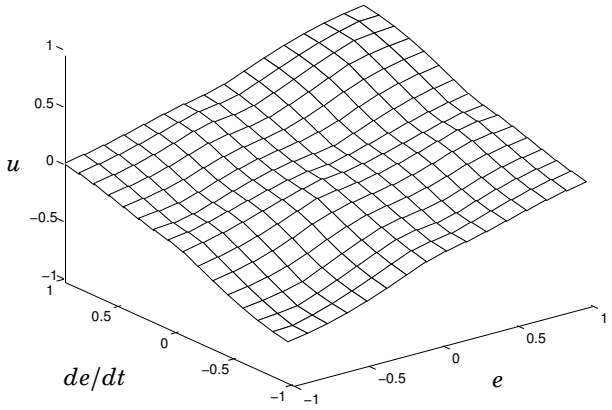

Figure 7.26 Graphic illustration of the nonlinearity of the fuzzy controller showing control signal u as function of control error e and its derivative.

can in fact be expressed as

$$u = F\left(e, \frac{de}{dt}\right)$$

where F is a nonlinear function of two variables. Thus, the fuzzy PD controller is a controller where the output is a nonlinear function of the error e and its derivative de/dt! In Figure 7.26 we give a graphic illustration of the nonlinearity defined by given rules for the PD controller with standard triangular membership functions and product fuzzification. The figure shows that the function is close to linear. In this particular case the fuzzy controller will behave similarly to an ordinary linear PD controller.

Fuzzy control may be considered as a way to represent a nonlinear function. Notice that it is still necessary to deal with generation of derivatives or integrals, integral windup, and all the other matters in the same way as for ordinary PID controllers. We may also inquire as to when it is useful to introduce the nonlinearities and what shape they should have.

Representation of a nonlinearity by fuzzification, fuzzy logic, and defuzzification is not very different from representation of a nonlinear function as a table with an interpolation procedure. Roughly speaking, the function values correspond to the rules; the membership functions and the fuzzification and defuzzification procedures correspond to the interpolation mechanism. To illustrate this we consider a function of two variables. Such a function can be visualized as a surface in two dimensions. A linear function is simply a tilted plane. This

function can be described completely by three points on a plane, i.e., three rules. More complex surfaces or functions are obtained by using more function values. The smoothness of the surface is expressed by the interpolation procedures.

From the point of view of control, the key question is understanding when nonlinearities are useful and what shape they should have. These are matters where much research remains to be done. There are cases where the nonlinearities can be very beneficial but also cases where the nonlinearities cause problems. It is also a nontrivial task to explore what happens. A few simulations of the behavior is not enough because the response of a nonlinear system is strongly amplitude dependent.

Let us also point out that the properties of the controller in Figure 7.24 are strongly influenced by the linear filter used. It is thus necessary to limit the high-frequency gain of the approximation of the derivative. It is also useful to take derivatives of the process output instead of the error, as was discussed in Section 3.4. Other filters can also be used; by adding an integrator to the output of the system in Figure 7.24, we obtain a fuzzy PI controller.

Applications

The representation of the control law as a collection of rules for linguistic variables has a strong intuitive appeal. It is easy to explain heuristically how the control system works. This is useful in communicating control strategies to persons with little formal training. It is one reason why fuzzy control is a good tool for automation of tasks that are normally done by humans. In this approach it is attempted to model the behavior of an operator in terms of linguistic rules. Fuzzy control has been used in a number of simple control tasks for appliances. It has also been used in controllers for processes that are complicated and poorly known. Control of a cement kiln is one example of this type of application. Fuzzy control has also been used for controller tuning.

7.8 Interacting Loops

An advantage to building a complex system from simple components by using a few control principles is that complexity is reduced by decomposition. In normal cases it is also comparatively easy to extrapolate the experience of commissioning and tuning single-loop control. It is also appealing to build up a complex system by gradual refinement. There are, however, also some drawbacks with the approach:

- Since we have not determined the fundamental limitations, it is difficult to decide when further refinements do not give any significant benefits.
- It is easy to get systems that are unnecessarily complicated. We may get systems where several control loops are fighting each other.
- There are cases where it is difficult to arrive at a good overall system by a loop-by-loop approach.

If there are difficulties, it is necessary to use a systematic approach based on mathematical modeling analysis and simulation. This is, however, more demanding than the empirical approach. In this section we illustrate some of the difficulties that may arise.

Parallel Systems

Systems that are connected in parallel are quite common. Typical examples are motors that are driving the same load, power systems and networks for steam distribution. Control of such systems require special consideration. To illustrate the difficulties that may arise we will consider the situation with two motors driving the same load. A schematic diagram of the system is shown in Figure 7.27.

Let ω be the angular velocity of the shaft, J the total moment of inertia, and D the damping coefficient. The system can then be described by the equation

$$J\frac{d\omega}{dt} + D\omega = M_1 + M_2 - M_L \tag{7.12}$$

where M_1 and M_2 are the torques from the motors and M_L is the load torque.

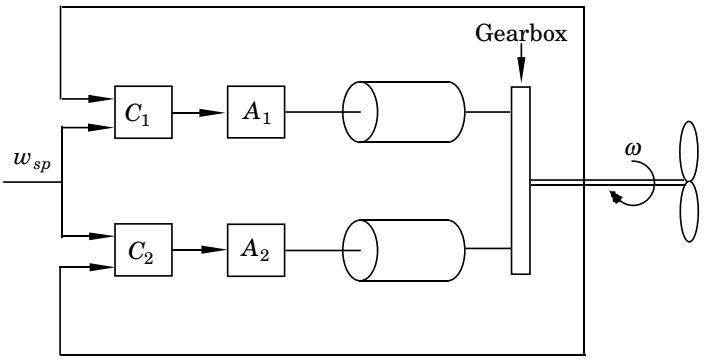

Figure 7.27 Schematic diagram of two motors that drive the same load.

Proportional Control

Assume each motor is provided with a proportional controller. The control strategies are then

$$\begin{aligned} M_1 &= M_{10} + K_1(\omega_{sp} - \omega) \\ M_2 &= M_{20} + K_2(\omega_{sp} - \omega) \end{aligned} \tag{7.13}$$

In these equations the parameters M_{10} and M_{20} give the torques provided by each motor when $\omega = \omega_{sp}$ and K_1 and K_2 are the controller gains. It follows from Equations (7.12) and (7.13) that

$$J\frac{d\omega}{dt} + (D + K_1 + K_2)\omega = M_{10} + M_{20} - M_L + (K_1 + K_2)\omega_{sp}$$

The closed-loop system is, thus, a dynamical system of first order. After perturbations the angular velocity reaches its steady state with a time constant

$$T = \frac{J}{D + K_1 + K_2}$$

The response speed is thus given by the sum of the damping and the controller gains. The stationary value of the angular velocity is given by

$$\omega = \omega_0 = \frac{K_1 + K_2}{D + K_1 + K_2}\,\omega_{sp} + \frac{M_{10} + M_{20} - M_L}{D + K_1 + K_2}$$

This implies that there normally will be a steady state error. Similarly we find from Equation (7.13) that

$$\frac{M_1 - M_{10}}{M_2 - M_{20}} = \frac{K_1}{K_2}$$

The ratio of the controller gains will indicate how the load is shared between the motors.

Proportional and Integral Control

The standard way to eliminate a steady state error is to introduce integral action. In Figure 7.28 we show a simulation of the system in which the motors have identical PI controllers. The setpoint is changed at time 0. A load disturbance in the form of a step in the load torque is introduced at time 10 and a pulse-like measurement disturbance in the second motor controller is introduced at time 20. When the measurement error occurs the balance of the torques is changed so that the first motor takes up much more of the load after the disturbance. In this particular case the second motor is actually breaking. This is highly undesirable, of course.

To understand the phenomena we show the block diagram of the system in Figure 7.29. The figure shows that there are two parallel

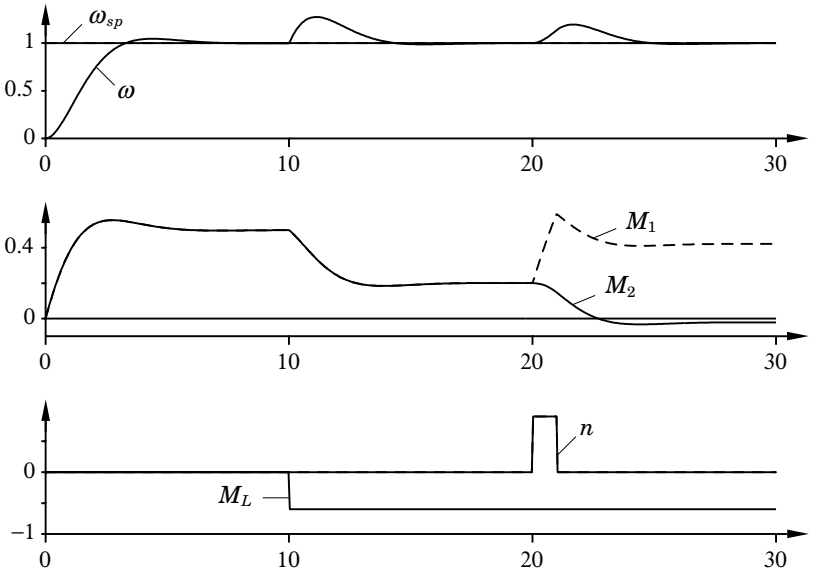

Figure 7.28 Simulation of a system with two motors with PI controllers that drive the same load. The figure shows setpoint ω_{sp}, process output ω, control signals M_1 and M_2, load disturbance M_L, and measurement disturbance n.

paths in the system that contain integration. This is a standard case where observability and controllability is lost. Expressed differently, it is not possible to change the signals M_1 and M_2 individually from the error. Since the uncontrollable state is an integrator, it does not go to zero after disturbance. This means that the torques can take on arbitrary values after disturbance. For example, it may happen that one of the motors takes practically all the load, clearly an undesirable situation.

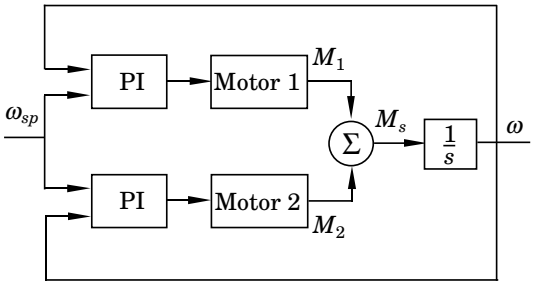

Figure 7.29 Block diagram for the system in Figure 7.28.

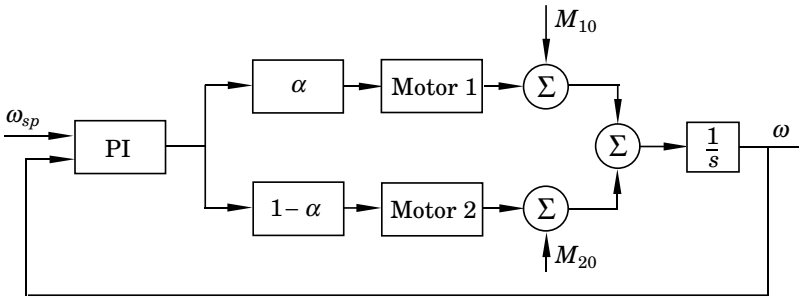

Figure 7.30 Block diagram of an improved control system.

How to Avoid the Difficulties

Having understood the reason for the difficulty, it is easy to modify the controller as shown in Figure 7.30. In this case only one controller with integral action is used. The output of this drives proportional controllers for each motor. A simulation of such a system is shown in Figure 7.31. The difficulties are clearly eliminated.

The difficulties shown in the examples with two motors driving the same load are even more accentuated if there are more motors.

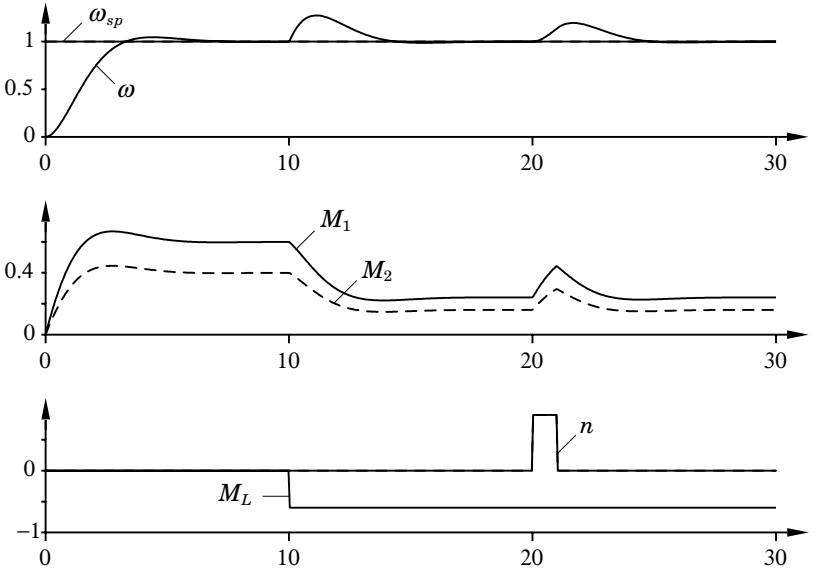

Figure 7.31 Simulation of the system with the modified controller. The figure shows setpoint ω_{sp}, process output ω, control signals M_1 and M_2, load disturbance M_L, and measurement disturbance n.

Good control in this case can be obtained by using one PI controller and distributing the outputs of this PI controller to the different motors, each of which has a proportional controller. An alternative is to provide one motor with a PI controller and let the other have proportional control. To summarize, we have found that there may be difficulties with parallel systems having integral action. The difficulties are caused by the parallel connection of integrators that produce unstable subsystems that are neither controllable nor observable. With disturbances these modes can change in an arbitrary manner. The remedy is to change the control strategies so there is only one integrator.

Interaction of Simple Loops

There are processes that have many control variables and many measured variables. Such systems are called multi-input multi-output (MIMO) systems. Because of the interaction between the signals, it may be very difficult to control such systems by a combination of simple controllers. A reasonably complete treatment of this problem is far outside the scope of this book. Let it suffice to illustrate some difficulties that may arise by considering processes with two inputs and two outputs. A block diagram of such a system is shown in Figure 7.32. A simple approach to control such a system is to use two single-loop controllers, one for each loop. To do this we must first decide how the controllers should be connected, i.e., if y_1 in the figure should be controlled by u_1 or u_2. This is called the pairing problem. This problem is straightforward if there is little interaction among the loops, which can be determined from the responses of all outputs to all inputs (see e.g. Figure 7.33). The single-loop approach will work well if there is small coupling between the loops. The loops can then be tuned separately. There may be difficulties, however, when there is coupling between the loops (as shown in the following example).

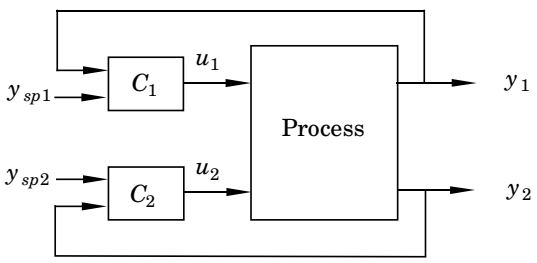

Figure 7.32 Block diagram of a system with two inputs and two outputs.

EXAMPLE 7.8 Rosenbrock's system

Consider a system with two inputs and two outputs. Sych a system can be characterized by giving the transfer functions that relate all inputs and outputs. These transfer functions can be organized as the matrix

$$G(s) = \begin{pmatrix} g_{11}(s) & g_{12}(s) \\ g_{21}(s) & g_{22}(s) \end{pmatrix} = \begin{pmatrix} \frac{1}{s+1} & \frac{2}{s+3} \\ \frac{1}{s+1} & \frac{1}{s+1} \end{pmatrix}$$

The first index refers to the outputs and the second to the inputs. In the matrix above, the transfer function g_{12} denotes the transfer function from the second input to the first output.

The behavior of the system can be illustrated by plotting the step responses from all inputs, as shown in Figure 7.33. From this figure we can see that there are significant interactions between the signals. The dynamics of all responses do, however, appear quite benign. In this case it is not obvious how the signals should be paired. Arbitrarily, we use the pattern 1-1, 2-2. It is very easy to design a controller for the individual loops if there is no interaction. The transfer function of the process is

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

in both cases. With PI control it is possible to obtain arbitrarily high gains, if there are no constraints on measurement noise or process uncertainty. A reasonable choice is to have $K = 19$, $b = 0$, and $T_i = 0.19$. This gives a system with relative damping $\zeta = 0.7$ and an undamped natural frequency of 10 rad/s. The responses obtained with this controller in one loop and the other loop open are shown in

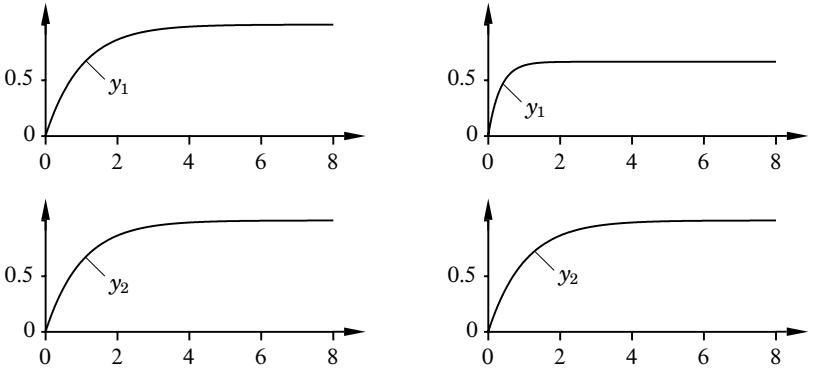

Figure 7.33 Open-loop step responses of the system. The left diagrams show responses to a step change in control signal u_1, and the right diagrams show responses to a step change in control signal u_2.

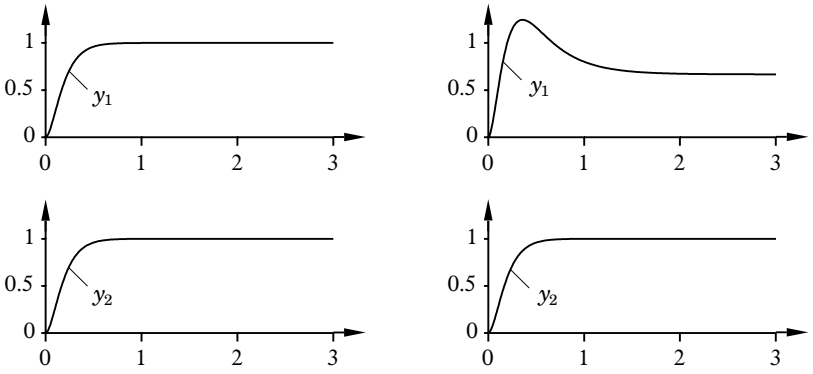

Figure 7.34 Step responses with one loop closed and the other open. The left diagrams show responses to steps in u_1 when controller C_2 is disconnected, and the right diagrams show responses to steps in u_2 when controller C_1 is disconnected.

Figure 7.34. Notice that the desired responses are as expected, but that there also are strong responses in the other signals. If both loops are closed with the controllers obtained, the system will be unstable.

In order to have reasonable responses with both loops closed, it is necessary to detune the loops significantly. In Figure 7.35 we show responses obtained when the controller in the first loop has parameters $K = 2$ and $T_i = 0.5$ and the other controller has $K = 0.8$ and $T_i = 0.7$. The gains are more than an order of magnitude smaller than the ones obtained with one loop open. A comparison with Figure 7.34 shows that the responses are significantly slower. □

The example clearly demonstrates the deficiencies in loop-by loop

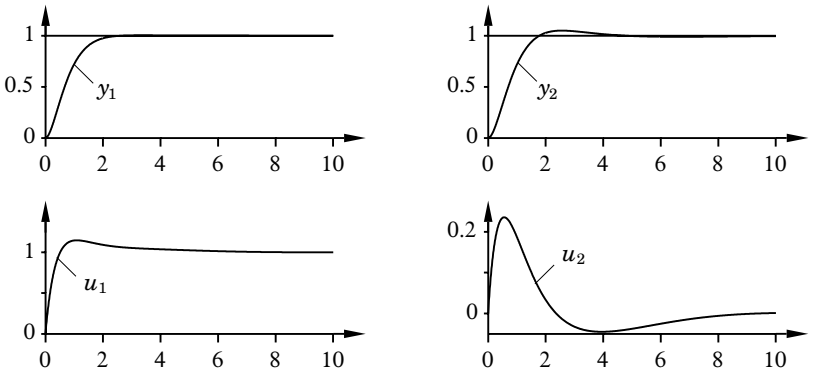

Figure 7.35 Step responses when both loops are closed. The figure shows responses to simultaneous setpoint changes in both loops.

tuning. The example chosen is admittedly somewhat extreme but it clearly indicates that it is necessary to have other techniques for truly multivariable systems. The reason for the difficulty is that the seemingly innocent system is actually a non-minimum phase multivariable system with a zero at $s = -1$.

Interaction Measures

The example given clearly indicates the need to have some way to find out if interactions may cause difficulties. There are no simple universal methods. An indication can be obtained by the relative gain array (RGA). This can be computed from the static gains in all loops in a multivariable system. For the a 2×2 system like the one in Example 7.8 the RGA is

$$R = \begin{pmatrix} \lambda & 1-\lambda \\ 1-\lambda & \lambda \end{pmatrix}$$

where

$$\lambda = \frac{g_{11}(0)g_{22}(0)}{g_{11}(0)g_{22}(0) - g_{12}(0)g_{21}(0)}$$

The number λ has physical interpretation as the ratio of the gain from u_1 to y_1 with the second loop open and with the second loop under very tight feedback ($y_2 = 0$). There is no interaction if $\lambda = 1$. If $\lambda = 0$ there is also no interaction, but the loops should be interchanged. The loops should be interchanged when $\lambda < 0.5$. The interaction is most severe if $\lambda = 0.5$.

For a multivariable system the relative gain is a matrix R in which component r_{ij} is given by

$$r_{ij} = g_{ij}h_{ji}$$

where g_{ij} is ij-th element of the static gain matrix G of the process and h_{ij} is the ij-th element of the the matrix

$$H = G^{-1}$$

Notice that g_{ij} is the static gain from input j to output i.

Bristol's recommendation for controller pairing is that the measured values and control variables should be paired so that the corresponding relative gains are positive, and as close to one as possible. If the gains are outside the interval $0.67 < \lambda < 1.5$, decoupling can improve the control significantly. Since the relative gain is based on the static properties of the system, it does not capture all aspects of the interaction.

7.9 System Structuring

In this section we illustrate how complex control systems can be built from simple components by using the paradigms we have discussed. The problem is quite complex. It involves selection of measured variables and control variables, and it requires significant physical understanding of the process.

The Process

The process to consider is a chemical reactor. A schematic diagram is shown in Figure 7.36. Two substances A and B are mixed in the reactor. They react to form a product. The reaction is exothermic, which means that it will generate heat. The heat is dissipated through water that is circulating in cooling pipes in the reactor. The reaction is very fast; equilibrium is achieved after a time that is much shorter than the residence time of the reactor. The flow q_A of substance A is considerably larger than q_B. Efficiency of the reaction and the heat generation is essentially proportional to the flow q_B.

A static process model is useful in order to understand the control problem. Figure 7.37 shows the efficiency and the heat generation as a function of temperature. In the figure we have drawn a straight line that corresponds to the cooling power. There are equilibria where the power generated by the reaction is equal to the cooling power represented at points P and Q in the figure. The point P corresponds

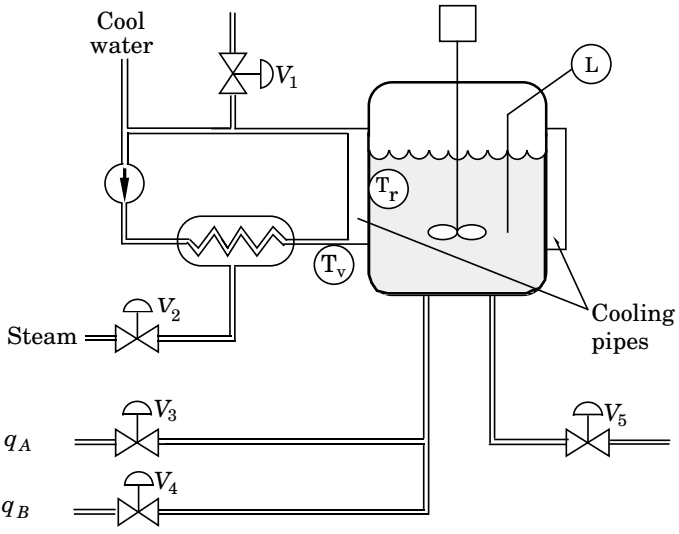

Figure 7.36 Schematic diagram of a chemical reactor.

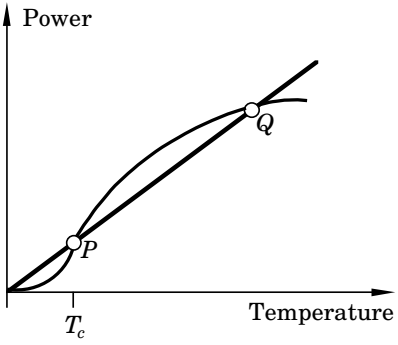

Figure 7.37 Static process model for the exothermic reactor.

to an unstable equilibrium. It follows from Figure 7.37 that if the temperature is increased above P the power generated by the reaction is larger than the cooling power. Temperature will thus increase. The catalyst in the reactor may be damaged if the temperature becomes too high. Similarly if the temperature decreases below point P it will continue to decrease and the reaction stops. This phenomena is called "freezing." Freezing starts at the surface of the cooling tube and will spread rapidly through the reactor. If this happens the reactor must be switched off and restarted again.

Design Requirements

There are considerable risks in running an exothermic reactor. The reactor can explode if the temperature is too high. To reduce the risk of explosion, the reactors are placed in special buildings far away from the operator. Because of the risk of explosion, it is not feasible to experiment with controller tuning. Consequently, it is necessary to compute controller setting beforehand and verify that the settings are correct before starting the reactor. Safety is the overriding requirement of the control system. It is important to guarantee that the reaction temperature will not be too high. It is also important to make sure that process upsets do not lead to loss of coolant flow, and that stirring does not lead to an explosion. It is also desirable to operate the reactor efficiently. This means that freezing must be avoided. Besides it is desirable to keep the efficiency as high as possible. Because of the risks, it is also necessary to automate start and stop as well as normal operation. It is desirable to avoid having to run the reactor under manual control. In this particular case the operator can set two variables, the reactor temperature and the ratio between the flows q_A and q_B. The reaction efficiency and the product quality can be influenced by these two variables.

Controller Structure

The reactor has five valves. Two of them, V_1 and V_2, influence the coolant temperature. The flow of the reactor is controlled by V_3 and V_4, and the product flow is controlled by the valve V_5. In this particular application the valve V_5 is controlled by process steps downstream. (Compare this with the discussion of surge tanks in Section 7.5.)

There are five measured signals: the reactor temperature T_r, the level in the reactor tank L, the cooling temperature T_v, and the flows q_A and q_B. The physical properties of the process gives a natural structuring of the control system. A mass balance for the material in the reactor tank shows that the level is essentially influenced by the flow q_A and the demanded production. It follows from the stochiometry of the reaction that the ratio of the flows q_A and q_B should be kept constant for an efficient reaction. The reactor temperature is strongly influenced by the water temperature, by the temperature of the coolant flow and the flows q_A and q_B. Coolant temperature is influenced by the valve V_1 that controls the amount of flow and by the steam valve V_2.

This simple physical discussion leads to the diagram shown in Figure 7.38, which shows the causality of the variables in the process. The valve V_5 can be regarded as a disturbance because it is set by downstream process units. Figure 7.38 suggests that there are three natural control loops:

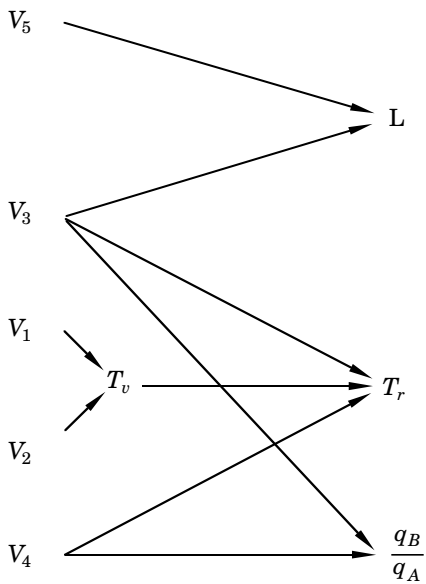

Figure 7.38 Causality diagram for the process variable.

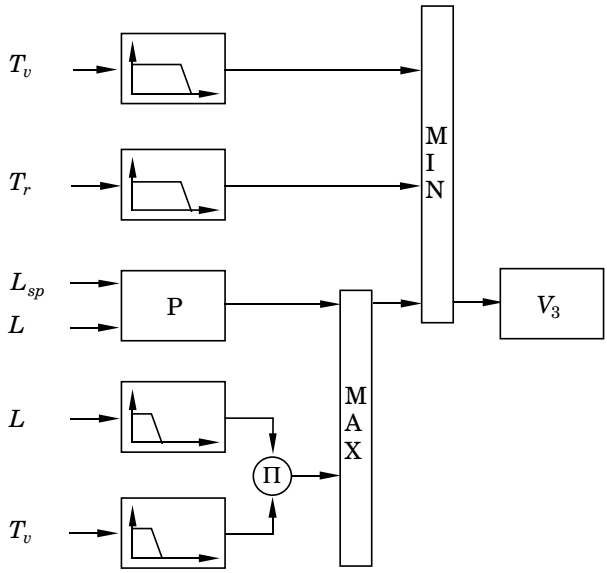

Figure 7.39 Block diagram for the level control through valve V_3.

1. Level control: Controlling the tank level with valve V_3.
2. Temperature control: Control of the reactor temperature with valves V_1 and V_2.
3. Flow ratio control: Control of ratio q_B/q_A with valve V_4.

These control loops are discussed in detail.

Level Control

The block diagram for the level control is shown in Figure 7.39. The primary function is a proportional feedback from the level to the flow q_A, which is controlled by the valve V_3. The reactor is also used as a surge tank to smooth out the difference between actual production and commanded production. The level in the tank will vary during normal operations. Reasonable limits are that the level should be between 50% and 100%. If the proportional band of the controller is chosen as 50%, the control variable will be fully closed when the tank is full and half-open when the tank is half-full. It is important that the reactor temperature remains within given bounds. The flow q_A is constrained, therefore, by two selectors based on measurements of the temperature in the reactor tank (T_r) and the coolant temperature (T_v). When starting the reactor the level is kept at the lower limit until the coolant temperature becomes sufficiently high. This

is achieved by combination of limiters, multipliers, and selectors, as shown in Figure 7.39.

Temperature Control

Figure 7.40 gives a block diagram for controlling the reactor temperature. Since the chemical reaction is fast compared to temperature and flow dynamics, the reactor can be viewed as a heat exchanger from the control point of view. During normal conditions the temperature is controlled by adjusting the coolant flow through the valve V_1. The primary control function is a feedback from temperature to the valves V_1 and V_2. The setpoint in this control loop can be adjusted manually. The parameters of this control loop can be determined as follows. The transfer function from coolant flow to the reactor temperature is approximately given by

$$G(s) = \frac{K_p}{(1 + sT_1)(1 + sT_2)} \tag{7.14}$$

where the time constant typically has values $T_1 = 300$ s and $T_2 = 50$ s. The following rough calculation gives approximate values of the controller parameter. A proportional controller with gain K gives the loop transfer function

$$G_0(s) = \frac{KK_p}{(1 + sT_1)(1 + sT_2)} \tag{7.15}$$

The characteristic equation of the closed loop becomes

$$s^2 + s\left(\frac{1}{T_1} + \frac{1}{T_2}\right) + \frac{1 + KK_p}{T_1T_2} = 0$$

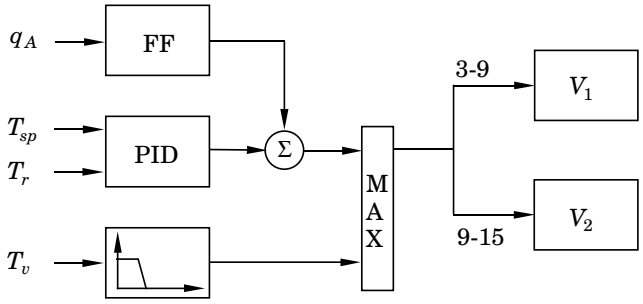

Figure 7.40 Block diagram showing temperature control through valves V_1 and V_2.

The closed system is thus of second order. The relative damping ζ and the the undamped natural frequency ω are given by

$$2\zeta\omega = \frac{1}{T_1} + \frac{1}{T_2} \approx \frac{1}{T_2} \tag{7.16}$$

and

$$2\zeta\omega^2 = \frac{1 + KK_p}{T_1 T_2} \tag{7.17}$$

The approximation in the first expression is motivated by $T_1 \gg T_2$. With a relative damping $\zeta = 0.5$ the Equation (7.16) then gives $\omega \approx 1/T_2$. Furthermore it follows from Equation (7.17) that

$$1 + KK_p = \frac{T_1}{T_2} = \frac{300}{50} = 6$$

The loop gain is thus essentially determined by the ratio of the time constants. The controller gain becomes

$$K = \frac{5}{K_p}$$

and the closed-loop system has the undamped natural frequency.

$$\omega = 1/T_2 = 0.02 \text{ rad/s}$$

If PI control is chosen instead, it is reasonable to choose a value of the integration time.

$$T_1 \approx 5T_2$$

Control can be improved by using derivative action. The achievable improvement depends on the time constant of the temperature sensor. In typical cases this time constant is between 10 s and 40 s. If it is as low as 10 s it is indeed possible to obtain improved control by introducing a derivative action in the controller. The derivative time can be chosen to eliminate the time constant T_2. We then obtain a system with the time constants 300 s and 10 s. The gain can then be increased so that

$$1 + KK_p = \frac{300}{10} = 30$$

and the undamped natural frequency of the system then becomes $\omega \approx 0.1$ rad/s. If the time constant of the temperature sensor is around 40 s, the derivative action gives only marginal improvements.

The heat generated by the chemical reaction is proportional to the flow q_A. To make sure that variations in q_A are compensated rapidly we have also introduced a feedforward from the flow q_A. This feedforward will only operate when the tank level is larger than 50% in order to avoid freezing when the reactor is started.

To start the reaction the reactor must be heated so that the temperature in the reaction vessel is larger than T_c (compare with

Figure 7.37). This is done by using the steam valve V_2. Split range control is used for the steam and water valves (compare Section 7.5). The water valve is open for low signals (3–9 PSI) and the steam valve is open for large pressures (9–15 PSI).

To avoid having the reactor freez, it is necessary to make sure that the reaction temperature is always larger than T_c. This is the reason for the extra feedback from water temperature to T_v through a maximum selector. This feedback makes sure that the steam valve opens if the temperature in the coolant flow becomes too low. Cascade control would be an alternative to this arrangement.

Flow Ratio Control

The ratio of the flows q_A and q_B must be kept constant. Figure 7.41 shows how the efficiency of the reaction depends on q_B when q_A is kept constant. The flow q_B is controlled with a ratio control system (as shown in Figure 7.42), which is the primary control function. The reaction rate depends strongly on q_B. To diminish the risk of explosion, there is a nonlinearity in the feedback that increases the gain when q_B/q_A is large. The flow loop has several selectors. At startup it is desirable that substance B is not added until the water temperature has reached the critical value T_c and the reactor tank is half-full. To achieve this the feedback from the water temperature and tank level have been introduced through limiters and a minimum selector. There are also limiters and a selector that closes valve V_4 rapidly if flow q_A is lost. There is also a direct feedback from q_A through limiters and selectors and a feedback from the reactor temperature that closes valve V_4, if the reactor temperature becomes too high.

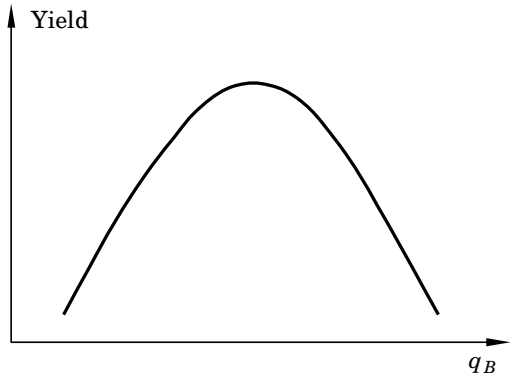

Figure 7.41 Reaction yield as a function of q_B at constant q_A.

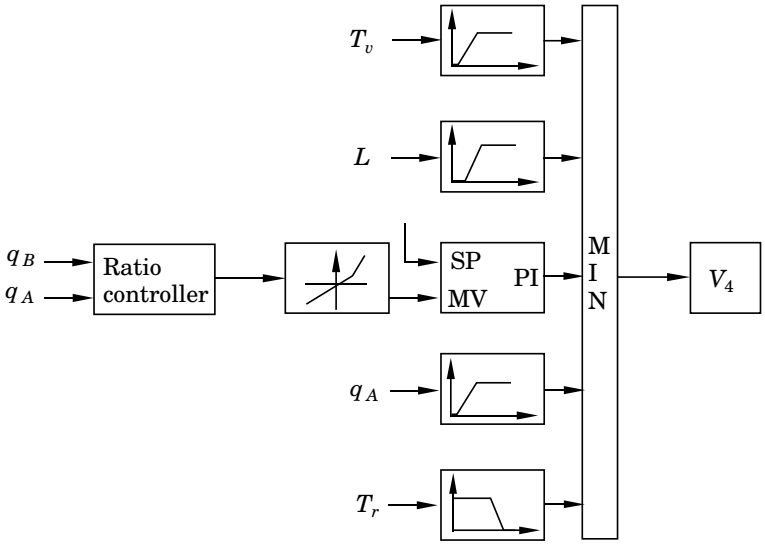

Figure 7.42 Block diagram for controlling the mixing ratio q_B/q_A through valve V_4.

Override Control of the Outlet Valve

The flow out of the reactor is determined by valve V_5. This valve is normally controlled by process steps downstream. The control of the reactor can be improved by introducing an override, which depends on the state of the reactor. When starting the reactor, it is desirable to have the outlet valve closed until the reactor tank is half-full and the reaction has started. This is achieved by introducing the tank level and the tank temperature to the setpoint of the valve controller via limiters and minimum selectors as is shown in Figure 7.43. The valve V_5 is normally controlled by q_{sp}. The minimum selector overrides the command q_{sp} when the level L or the temperature T_r are too low.

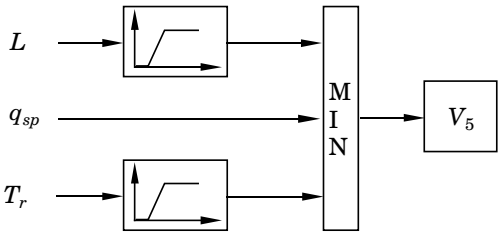

Figure 7.43 Block diagram for controlling the outflow of the reactor through valve V_5.

7.10 Conclusions

In this chapter we have illustrated how complex control systems can be built from simple components such as PID controllers, linear filters, gain schedules, and simple nonlinear functions. A number of control paradigms have been introduced to guide system design.

The primary linear control paradigms are feedback by PID control, and feedforward. Cascade control can be used to enhance control performance through the use of extra measurements. Observers can be used in a related way when measurements are not available. Control by observers and state feedback may be viewed as a natural extension of cascade control.

Smith predictors (discussed in Section 3.9) can be used to improve control of systems with long dead time, and notch filters and other filters with complex poles and zeros are useful when controlling systems with poorly damped oscillatory modes.

We also discussed several nonlinear components and related paradigms. The nonlinearities used are nonlinear functions, gain schedules, limiters, and selectors. In Section 3.4, how simple PID controllers could be enhanced by simple nonlinear functions was discussed. This was used to avoid windup and to provide special control functions like "error squared on integral," etc. Ratio control is a nonlinear strategy that admits control of two process variables so that their ratio is constant. In Section 6.3 we discussed how gain schedules could be used to cope with control of processes with nonlinear characteristics. Gain schedules and nonlinear functions are also useful for control paradigms such as surge tank control, where the goal is not to keep process variables constant but to allow them to vary in prescribed ranges. Selector control is another very important paradigm that is used for constraint control where certain process variables have to be kept within given constraints. We also showed that controllers based on neural and fuzzy techniques could be interpreted as nonlinear controllers.

Parameter estimation, discussed in Section 2.7, can be used to estimate process parameters. Adaptation and tuning are other paradigms that were discussed in Chapter 6.

There are many ways to use the different control paradigms. We have also indicated that there may be difficulties due to interaction of several loops.

7.11 References

Many aspects on the material of this chapter are found in the classical textbooks on process control such as (Buckley, 1964), (Shinskey,

1988), and (Seborg *et al.*, 1989). A more specialized presentation is given in (Hägglund, 1991). The books (Shinskey, 1981) and (Klefenz, 1986) focus on applications to energy systems.

The methods discussed in this chapter can all be characterized as bottom-up procedures in the sense that a complex system is built up by combining simple components. An interesting view of this is presented in (Bristol, 1980). A top-down approach is another possibility. A discussion of this, which is outside the scope of this book, is found in (Seborg *et al.*, 1986).

Cascade and feedforward control are treated in the standard texts on control. A presentation with many practical aspects is found in (Tucker and Wills, 1960). Selector control is widely used in practice. A general presentation is given in (Åström, 1987b). Many applications are given in the books (Shinskey, 1978) and (Klefenz, 1986). Analysis of a simple scheme is given in (Foss, 1981). It is difficult to analyse nonlinear systems. A stability analysis of a system with selectors is given in (Foss, 1981).

Fuzzy control has been around for a long time, see (Mamdani, 1974), (Mamdani and Assilian, 1974), (King and Mamdani, 1977), and (Tong, 1977). It has recently received a lot of attention particularly in Japan: see (Zadeh, 1988), (Tong, 1984), (Sugeno, 1985), (Driankov *et al.*, 1993), and (Wang, 1994). The technique has been used for automation of complicated processes that have previously been controlled manually. Control of cement kilns is a typical example, see (Holmblad and Østergaard, 1981). There has been a similar development in neural networks, see, for example, (Hecht-Nielsen, 1990), (Pao, 1990), and (Åström and McAvoy, 1992). There was a lot of activity in neural networks during the late 1960s, which vanished rapidly. There was a rapid resurgence of interest in the 1980s. There are a lot of exaggerations both in fuzzy and neural techniques, and no balanced view of the relevance of the fields for control has yet emerged. The paper (Willis *et al.*, 1991) gives an overview of possible uses of neural networks for process control, and the paper (Pottman and Seborg, 1993) describes an application to control of pH. The papers (Lee, 1990), (Huang, 1991), and (Swiniarski, 1991) describe applications to PID controllers and their tuning. There have also been attempts to merge fuzzy and neural control, see (Passino and Antsaklis, 1992) and (Brown and Harris, 1994).

Some fundamental issues related to interaction in systems are treated in (Rijnsdorp, 1965a), (Rijnsdorp, 1965b), and (McAvoy, 1983). The relative gain array was described in (Bristol, 1966). Control of systems with strong interaction between many loops require techniques that are very different from those discussed in this chapter, see, for example, (Cutler and Ramaker, 1980) and (Seborg *et al.*, 1986). Section 7.9 is based on (Buckley, 1970).

INDEX

D

E

Bibliography

Anderson, K. L., G. L. Blankenship, and L. G. Lebow (1988): "A rule-based PID controller." In *Proc. IEEE Conference on Decision and Control*, Austin, Texas.

Anderssen, A. S. and E. T. White (1970): "Parameter estimation by the transfer function method." *Chemical Engineering Science*, **25**, pp. 1015–1021.

Anderssen, A. S. and E. T. White (1971): "Parameter estimation by the weighted moments method." *Chemical Engineering Science*, **26**, pp. 1203–1221.

Antsaklis, P. J., K. M. Passino, and S. J. Wang (1991): "An introduction to autonomous control systems." *IEEE Control Systems Magazine*, **11:4**, pp. 5–13.

Åström, K. J. (1987a): "Adaptive feedback control." *Proc. IEEE*, **75**, February, pp. 185–217. Invited paper.

Åström, K. J. (1987b): "Advanced control methods—Survey and assessment of possibilities." In Morris and Williams, Eds., *Advanced Control in Computer Integrated Manufacturing*, Proceedings 13th Annual Advanced Control Conference. Purdue University, West Lafayette, Indiana.

Åström, K. J. (1991): "Assessment of achievable performance of simple feedback loops." *International Journal of Adaptive Control and Signal Processing*, **5**, pp. 3–19.

Åström, K. J. (1992): "Autonomous control." In Bensoussan and Verjus, Eds., *Future Tendencies in Computer Science, Control and Applied Mathematics*, volume 653 of *Lecture Notes in Computer Science*, pp. 267–278. Springer-Verlag.

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

Åström, K. J. and T. Hägglund (1988): *Automatic Tuning of PID Controllers*. Instrument Society of America, Research Triangle Park, North Carolina.

Åström, K. J., T. Hägglund, C. C. Hang, and W. K. Ho (1993): "Automatic tuning and adaptation for PID controllers—A survey." *Control Engineering Practice*, **1:4**, pp. 699–714.

Åström, K. J., C. C. Hang, P. Persson, and W. K. Ho (1992): "Towards intelligent PID control." *Automatica*, **28:1**, pp. 1–9.

Åström, K. J. and T. J. McAvoy (1992): "Intelligent control." *Journal of Process Control*, **2:2**, pp. 1–13.

Åström, K. J. and L. Rundqwist (1989): "Integrator windup and how to avoid it." In *Proceedings of the American Control Conference (ACC '89)*, pp. 1693–1698, Pittsburgh, Pennsylvania.

Åström, K. J. and H. Steingrímsson (1991): "Implementation of a PID controller on a DSP." In Ahmed, Ed., *Digital Control Applications with the TMS 320 Family*, Selected Application Notes, pp. 205–238. Texas Instruments.

Åström, K. J. and B. Wittenmark (1984): *Computer Controlled Systems—Theory and Design*. Prentice-Hall, Englewood Cliffs, New Jersey.

Åström, K. J. and B. Wittenmark (1989): *Adaptive Control*. Addison-Wesley, Reading, Massachusetts.

Åström, K. J. and B. Wittenmark (1990): *Computer Controlled Systems—Theory and Design*. Prentice-Hall, Englewood Cliffs, New Jersey, second edition.

Atherton, D. P. (1975): *Nonlinear Control Engineering—Describing Function Analysis and Design*. Van Nostrand Reinhold Co., London, UK.

Bialkowski, W. L. (1993): "Dreams versus reality: A view from both sides of the gap." *Pulp and Paper Canada*, **94:11**.

Blickley, G. (1990): "Modern control started with ziegler-nichols tuning." *Control Engineering*, October, pp. 11–17.

Blickley, G. J. (1988): "PID tuning made easy with hand-held computer." *Control Engineering*, November, p. 99.

Boyd, S. P. and C. H. Barratt (1991): *Linear Controller Design – Limits of Performance*. Prentice Hall Inc., Englewood Cliffs, New Jersey.

Bristol, E. (1966): "On a new measure of interaction for multivariable process control." *IEEE Transactions on Automatic Control*, **11**, p. 133.

Bristol, E. H. (1967): "A simple adaptive system for industrial control." *Instrumentation Technology*, June.

Bristol, E. H. (1970): "Adaptive control odyssey." In *ISA Silver Jubilee Conference*, Paper 561–570, Philadelphia.

Bristol, E. H. (1977): "Pattern recognition: An alternative to param-

eter identification in adaptive control." *Automatica*, **13**, pp. 197–202.

Bristol, E. H. (1980): "After DDC: Idiomatic (structured) control." In *Proceedings American Institute of Chemical Engineering (AIChE)*, Philadelphia.

Bristol, E. H. (1986): "The EXACT pattern recognition adaptive controller, a user-oriented commercial success." In Narendra, Ed., *Adaptive and Learning Systems*, pp. 149–163, New York. Plenum Press.

Bristol, E. H., G. R. Inaloglu, and J. F. Steadman (1970): "Adaptive process control by pattern recognition." *Instrum. Control Systems*, pp. 101–105.

Bristol, E. H. and T. W. Kraus (1984): "Life with pattern adaptation." In *Proc. 1984 American Control Conference*, San Diego, California.

Brown, M. and C. Harris (1994): *Neurofuzzy Adaptive Modelling and Control*. Prentice Hall.

Buckley, P. S. (1964): *Techniques of Process Control*. John Wiley & Sons, Inc.

Buckley, P. S. (1970): "Protective controls for a chemical reactor." *Chemical Engineering*, April, pp. 145–150.

Callaghan, P. J., P. L. Lee, and R. B. Newell (1986): "Evaluation of foxboro controller." *Process Control Engineering*, **May**, pp. 38–40.

Callender, A., D. R. Hartree, and A. Porter (1936): "Time lag in a control system." *Philos. Trans. A.*, **235**, pp. 415–444.

Cameron, F. and D. E. Seborg (1983): "A self-tuning controller with a PID structure." *Int. J. Control*, **38:2**, pp. 401–417.

Chen, B.-S. and S.-S. Wang (1988): "The stability of feedback control with nonlinear saturating actuator: Time domain approach." *IEEE Transactions on Automatic Control*, **33**, pp. 483–487.

Chen, C.-L. (1989): "A simple method for on-line identification and controller tuning." *AIChE Journal*, **35:12**, pp. 2037–2039.

Chien, I. L. (1988): "IMC-PID controller design—an extension." In *IFAC Symposium, Adaptive Control of Chemical Processes*, pp. 155–160, Copenhagen, Denmark.

Chien, I.-L. and P. S. Fruehauf (1990): "Consider IMC tuning to improve controller performance." *Chemical Engineering Progress*, October, pp. 33–41.

Chien, K. L., J. A. Hrones, and J. B. Reswick (1952): "On the automatic control of generalized passive systems." *Trans. ASME*, **74**, pp. 175–185.

Clarke, D. W. (1984): "PID algorithms and their computer implementation." *Trans. Inst. M. C.*, **6:6**, pp. 305–316.

Close, C. M. and D. K. Frederick (1993): *Modeling and Analysis of Dynamic Systems*. Houghton Mifflin.

Cohen, G. H. and G. A. Coon (1953): "Theoretical consideration of retarded control." *Trans. ASME*, **75**, pp. 827–834.

Coon, G. A. (1956a): "How to find controller settings from process characteristics." *Control Engineering*, **3**, pp. 66–76.

Coon, G. A. (1956b): "How to set three-term controller." *Control Engineering*, **3**, pp. 71–76.

Corripio, A. B. (1990): *Tuning of Industrial Control Systems*. Instrument Society of America.

Cutler, C. R. and B. C. Ramaker (1980): "Dynamic matrix control—A computer control algorithm." In *Proceedings Joint Automatic Control Conference*, Paper WP5-B, San Francisco, California.

Dahlin, E. B. (1968): "Designing and tuning digital controllers." *Instruments and Control Systems*, **42**, June, pp. 77–83.

Deshpande, P. B. and R. H. Ash (1981): *Elements of Computer Process Control with Advanced Control Applications*. Instrument Society of America, Research Triangle Park, North Carolina.

Dreinhofer, L. H. (1988): "Controller tuning for a slow nonlinear process." *IEEE Control Systems Magazine*, **8:2**, pp. 56–60.

Driankov, D., H. Hellendoorn, and M. Reinfrank (1993): *An Introduction to Fuzzy Control*. Springer-Verlag.

Dumont, G. A. (1986): "On the use of adaptive control in the process industries." In Morari and McAvoy, Eds., *Proceedings Third International Conference on Chemical Process Control–CPCIII*, Amsterdam. Elsevier.

Dumont, G. A., J. M. Martin-S nchez, and C. C. Zervos (1989): "Comparison of an auto-tuned PID regulator and an adaptive predictive control system on an industrial bleach plant." *Automatica*, **25**, pp. 33–40.

Elgerd, O. I. and W. C. Stephens (1959): "Effect of closed-loop transfer function pole and zero locations on the transient response of linear control systems." *Applications and Industry*, **42**, pp. 121–127.

Ender, D. B. (1993): "Process control performance: Not as good as you think." *Control Engineering*, **40:10**, pp. 180–190.

Fertik, H. A. (1975): "Tuning controllers for noisy processes." *ISA Transactions*, **14**, pp. 292–304.

Fertik, H. A. and C. W. Ross (1967): "Direct digital control algorithms

with anti-windup feature." *ISA Trans.*, **6:4**, pp. 317–328.

Foss, A. M. (1981): "Criterion to assess stability of a 'lowest wins' control strategy." *IEEE Proc. Pt. D*, **128:1**, pp. 1–8.

Foxboro, Inc. (1979): *Controller Tuning Guide*, PUB 342A.

Frank, P. M. (1990): "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy—A survey and some new results." *Automatica*, **26:3**, pp. 459–474.

Fröhr, F. (1967): "Optimierung von Regelkreisen nach dem Betragsoptimum und dem symmetrischen Optimum." *Automatik*, **12**, January, pp. 9–14.

Fröhr, F. and F. Orttenburger (1982): *Introduction to Electronic Control Engineering*. Siemens Aktiengesellschaft, Heyden & Son Ltd, London.

Gallun, S. E., C. W. Matthews, C. P. Senyard, and B. Slater (1985): "Windup protection and initialization for advanced digital control." *Hydrocarbon Processing*, June, pp. 63–68.

Gawthrop, P. J. (1986): "Self-tuning PID controllers: Algorithms and implementation." *IEEE Transactions on Automatic Control*, **31**, pp. 201–209.

Gelb, A. and W. E. V. Velde (1968): *Multiple-Input Describing Functions and Nonlinear System Design*. McGraw-Hill, New York.

Gerry, J. P. (1987): "A comparison of PID controller algorithms." *Control Engineering*, March, pp. 102–105.

Gille, J. C., M. J. Pelegrin, and P. Decaulne (1959): *Feedback Control Systems*. McGraw-Hill, New York.

Glattfelder, A. H., L. Guzzella, and W. Schaufelberger (1988): "Bumpless transfer, anti-reset-windup, saturating and override controls: A status report on self-tuning regulators." In *Proceedings of IMACS-88, Part 2*, pp. 66–72, Paris, France.

Glattfelder, A. H. and Schaufelberger (1983): "Stability analysis of single loop systems with saturation and antireset-windup circuits." *IEEE Transactions on Automatic Control*, **28**, pp. 1074–1081.

Glattfelder, A. H. and W. Schaufelberger (1986): "Start-up performance of different proportional-integral-anti-wind-up regulators." *International Journal of Control*, **44**, pp. 493–505.

Goff, K. W. (1966a): "Dynamics in direct digital control I—Estimating characteristics and effects of noisy signals." *ISA Journal*, **13**, November, pp. 45–49.

Goff, K. W. (1966b): "Dynamics in direct digital control II—A systematic approach to DDC design." *ISA Journal*, **13**, December,

pp. 44–54.

Graham, D. and R. C. Lathrop (1953): "The synthesis of 'optimum' transient response: Criteria and standard forms." *Transactions of the AIEE*, **72**, November, pp. 273–288.

Grebe, J. J., R. H. Boundy, and R. W. Cermak (1933): "The control of chemical processes." *Trans. of American Institute of Chemical Engineers*, **29**, pp. 211–255.

Haalman, A. (1965): "Adjusting controllers for a deadtime process." *Control Engineering*, **July-65**, pp. 71–73.

Habel, F. (1980): "Ein Verfahren zur Bestimmung der Parametern von PI-, PD- und PID-Reglern." *Regelungstechnik*, **28:6**, pp. 199–205.

Hägglund, T. (1991): *Process Control in Practice*. Chartwell-Bratt Ltd, Bromley, UK.

Hägglund, T. (1992): "A predictive PI controller for processes with long dead times." *IEEE Control Systems Magazine*, **12:1**, pp. 57–60.

Hägglund, T. (1993): "Disturbance supervision in feedback loops." In *Preprints Tooldiag'93, International Conference on Fault Diagnosis*, Toulouse, France.

Hägglund, T. and K. J. Åström (1991): "Industrial adaptive controllers based on frequency response techniques." *Automatica*, **27**, pp. 599–609.

Hang, C. C., K. J. Åström, and W. K. Ho (1991): "Refinements of the Ziegler-Nichols tuning formula." *IEE Proceedings, Part D*, **138:2**, pp. 111–118.

Hang, C. C., K. J. Åström, and W. K. Ho (1993a): "Relay auto-tuning in the presence of static load disturbance." *Automatica*, **29:2**, pp. 563–564.

Hang, C. C., T. H. Lee, and W. K. Ho (1993b): *Adaptive Control*. Instrument Society of America, Research Triangle Park, North Carolina.

Hang, C. C. and K. K. Sin (1991): "An on-line auto-tuning method based on cross-correlation." *IEEE Transactions on Industrial Electronics*, **38:6**, pp. 428–437.

Hanus, R. (1988): "Antiwindup and bumpless transfer: a survey." In *Proceedings of IMACS-88, Part 2*, pp. 59–65, Paris, France.

Hanus, R., M. Kinnaert, and J.-L. Henrotte (1987): "Conditioning technique, a general anti-windup and bumpless transfer method." *Automatica*, **23:729–739**.

Harriott, P. (1964): *Process Control*. McGraw-Hill, New York.

Harris, C. J. and S. A. Billings, Eds. (1981): *Self-tuning and Adaptive Control: Theory and Applications*. Peter Peregrinus, London.

Hartree, D. R., A. Porter, A. Callender, and A. B. Stevenson (1937): "Time-lag in control systems—II." *Proceedings of the Royal Society of London*, **161**, pp. 460–476.

Hawk, Jr., W. M. (1983): "A self-tuning, self-contained PID controller." In *Proc. 1983 American Control Conference*, pp. 838–842, San Francisco, California.

Hazebroek, P. and B. L. van der Waerden (1950): "Theoretical considerations on the optimum adjustment of regulators." *Trans. ASME*, **72**, pp. 309–322.

Hazen, H. L. (1934): "Theory of servomechanisms." *JFI*, **218**, pp. 283–331.

Hecht-Nielsen, R. (1990): *Neurocomputing*. Addison-Wesley.

Hess, P., F. Radke, and R. Schumann (1987): "Industrial applications of a PID self-tuner used for system start-up." In *Preprints 10th IFAC World Congress*, volume 3, pp. 21–26, Munich, Germany.

Higham, E. H. (1985): "A self-tuning controller based on expert systems and artificial intelligence." In *Proceedings of Control 85*, pp. 110–115, England.

Higham, J. D. (1968): "'Single-term' control of first- and second-order processes with dead time." *Control*, February, pp. 2–6.

Holmblad, L. P. and J. ■stergaard (1981): "Control of a cement kiln by fuzzy logic." *F.L. Smidth Review*, **67**, pp. 3–11. Copenhagen, Denmark.

Hoopes, H. S., W. M. Hawk, Jr., and R. C. Lewis (1983): "A self-tuning controller." *ISA Transactions*, **22:3**, pp. 49–58.

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

Howes, G. (1986): "Control of overshoot in plastics-extruder barrel zones." In *EI Technology*, No. 3, pp. 16–17. Eurotherm International, Brighton, UK.

Huang, Z. (1991): "Auto-tuning of PID controllers using neural networks." In *Preprints IFAC International Symposium on Intelligent Tuning and Adaptive Control (ITAC 91)*, Singapore.

Hwang, S.-H. and H.-C. Chang (1987): "A theoretical examination of closed-loop properties and tuning methods of single-loop PI controllers." *Chemical Engineering Science*, **42**, pp. 2395–2415.

Isermann, R. (1980): "Practical aspects of process identification." *Automatica*, **16**, pp. 575–587.

Isermann, R. (1982): "Parameter adaptive control algorithms—A tu-

torial." *Automatica*, **18**, pp. 513–528.

Isermann, R. (1984): "Process fault detection based on modeling and estimation methods—A survey." *Automatica*, **20**, pp. 387–404.

Isermann, R., W. Appel, B. Freyermuth, A. Fuchs, W. Janik, D. Neumann, T. Reiss, and P. Wanke (1990): "Model based fault diagnosis and supervision of machines and drives." In *Preprints 11th IFAC World Congress*, Tallinn, Estonia.

Ivanoff, A. (1934): "Theoretical foundations of the automatic regulation of temperature." *J. Institute of Fuel*, **7**, pp. 117–138.

Johansson, R. (1993): *System Modeling and Identification*. Prentice Hall, Englewood Cliffs, New Jersey.

Kapasouris, P. and M. Athans (1985): "Multivariable control systems with saturating actuators antireset windup strategies." In *Proc. Automatic Control Conference*, pp. 1579–1584, Boston, Massachusetts.

Kaya, A. and S. Titus (1988): "A critical performance evaluation of four single loop self tuning control products." In *Proceedings of the 1988 American Control Conference*, Atlanta, Georgia.

Kessler, C. (1958a): "Das symmetrische Optimum, Teil I." *Regelungstechnik*, **6:11**, pp. 395–400.

Kessler, C. (1958b): "Das symmetrische Optimum, Teil II." *Regelungstechnik*, **6:12**, pp. 432–436.

King, P. J. and E. H. Mamdani (1977): "The application of fuzzy control systems to industrial processes." *Automatica*, **13**, pp. 235–242.

Klefenz, G. (1986): *Automatic Control of Steam Power Plants*. Bibliographisches Institut, third edition.

Klein, M., T. Marczinkowsky, and M. Pandit (1991): "An elementary pattern recognition self-tuning PI-controller." In *Preprints IFAC International Symposium on Intelligent Tuning and Adaptive Control (ITAC 91)*, volume 1, Singapore.

Kramer, L. C. and K. W. Jenkins (1971): "A new technique for preventing direct digital control windup." In *Proc. Joint Automatic Control Conference*, pp. 571–577, St Louis, Missouri.

Kraus, T. W. and T. J. Myron (1984): "Self-tuning PID controller uses pattern recognition approach." *Control Engineering*, June, pp. 106–111.

Krikelis, N. J. (1984): "Design of tracking systems subject to actuators and saturation and integrator windup." *International Journal of Control*, **39:4**, pp. 667–682.

Küpfmüller, K. (1928): "Über die Dynamik der selbststätigen Ver-

stärkungsregler." *ENT*, **5**, pp. 459–467.

Lee, C. C. (1990): "A self-learning rule-based controller with approximate reasoning and neural nets." In *Preprints 11th IFAC World Congress*, Tallinn, Estonia.

Leva, A. (1993): "PID autotuning algorithm based on relay feedback." *IEE Proceedings D*, **140:5**, pp. 328–338.

Ljung, L. (1987): *System Identification—Theory for the User*. Prentice Hall, Englewood Cliffs, New Jersey.

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

Lloyd, S. G. and G. D. Anderson (1971): *Industrial Process Control*. Fisher Controls Co., Marshalltown, Iowa.

L&N (1968): *Leeds & Northrup Technical Journal*. Spring Issue, Number 3.

Lopez, A. M., J. A. Miller, C. L. Smith, and P. W. Murrill (1967): "Tuning controllers with error-integral criteria." *Instrumentation Technology*, November, pp. 57–62.

Lopez, A. M., P. W. Murrill, and C. L. Smith (1969): "Tuning PI and PID digital controllers." *Instruments and Control Systems*, **42**, February, pp. 89–95.

Lukas, M. P. (1986): *Distributed Process Control Systems—Their Evaluation and Design*. Van Nostrand Reinhold, New York.

Luyben, W. L. (1990): *Process Modeling, Simulation and Control for Chemical Engineers*. McGraw-Hill, second edition.

Maciejowski, J. M. (1989): *Multivariable Feedback Design*. Addison-Wesley, Reading, Massachusetts.

Mamdani, E. H. (1974): "Application of fuzzy algorithm for control of simple dynamic plant." *Proc. IEE*, **121**, pp. 1585–1588.

Mamdani, E. H. and S. Assilian (1974): "A case study on the application of fuzzy set theory to automatic control." In *Proceedings IFAC Stochastic Control Symposium*, Budapest, Hungary.

Mantz, R. J. and E. J. Tacconi (1989): "Complementary rules to Ziegler and Nichols' rules for a regulating and tracking controller." *International Journal of Control*, **49**, pp. 1465–1471.

Marsik, J. and V. Strejc (1989): "Application of identification-free algorithms for adaptive control." *Automatica*, **25**, pp. 273–277.

Marsili-Libelli, S. (1981): "Optimal design of PID regulators." *International Journal of Control*, **33:4**, pp. 601–616.

Maxwell, J. C. (1868): "On governors." *Proceedings of the Royal Society of London*, **16**, pp. 270–283. Also published in "Mathematical Trends in Control Theory" edited by R. Bellman and R. Kalaba,

Dover Publications, New York 1964, pp. 3–17.

McAvoy, T. J. (1983): *Interaction Analysis: Principles and Applications*. Instrument Society of America, Research Triangle Park, North Carolina.

McMillan, G. K. (1983): *Tuning and Control Loop Performance*. Instrument Society of America, Research Triangle Park, North Carolina, second edition.

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

McMillan, G. K., W. K. Wojsznis, and G. T. Borders, Jr (1993a): "Flexible gain scheduler." In *Advances in Instrumentation and Control*, volume 48 of *ISA Conference*, pp. 811–818.

McMillan, G. K., W. K. Wojsznis, and K. Meyer (1993b): "Easy tuner for DCS." In *Advances in Instrumentation and Control*, volume 48 of *ISA Conference*, pp. 703–710.

Meyer, C., D. E. Seborg, and R. K. Wood (1976): "A comparison of the Smith predictor and conventional feedback control." *Chemical Engineering Science*, **31**, pp. 775–778.

Miller, J. A., A. M. Lopez, C. L. Smith, and P. W. Murrill (1967): "A comparison of controller tuning techniques." *Control Engineering*, December, pp. 72–75.

Minorsky, N. (1922): "Directional stability of automatically steered bodies." *J. Amer. Soc. of Naval Engineers*, **34:2**, pp. 280–309.

Moore, C. F., C. L. Smith, and P. W. Murrill (1970): "Improved algorithm for direct digital control." *Instruments & Control Systems*, **43**, January, pp. 70–74.

Morari, M. and J. H. Lee (1991): "Model predictive control: The good, the bad, and the ugly." In *Chemical Process Control, CPCIV*, pp. 419–442, Padre Island, Texas.

Morris, H. M. (1987): "How adaptive are adaptive process controllers?" *Control Engineering*, **34-3**, pp. 96–100.

Nachtigal, C. L. (1986a): "Adaptive controller performance evaluation: Foxboro EXACT and ASEA Novatune." In *Proceedings ACC-86*, pp. 1428–1433.

Nachtigal, C. L. (1986b): "Adaptive controller simulated process results: Foxboro EXACT and ASEA Novatune." In *Proceedings ACC-86*, pp. 1434–1439.

Newton, Jr, G. C., L. A. Gould, and J. F. Kaiser (1957): *Analytical Design of Linear Feedback Controls*. John Wiley & Sons.

Nicholson, H., Ed. (1980): *Modelling of Dynamical Systems, Vol. 1*. Peter Peregrinus.

Nicholson, H., Ed. (1981): *Modelling of Dynamical Systems, Vol. 2*. Peter Peregrinus.

Nishikawa, Y., N. Sannomiya, T. Ohta, and H. Tanaka (1984): "A method for auto-tuning of PID control parameters." *Automatica*, **20**, pp. 321–332.

Nyquist, H. (1932): "Regeneration theory." *Bell System Technical Journal*, **11**, pp. 126–147. Also published in "Mathematical Trends in Control Theory" edited by R. Bellman and R. Kalaba, Dover Publications, New York 1964, pp. 83–105.

Oldenburg, R. C. and H. Sartorius (1954): "A uniform approach to the optimum adjustment of control loops." *Transactions of the ASME*, **76**, November, pp. 1265–1279.

Oppelt, W. (1964): *Kleines Handbuch technischer Regelvorgänge*. Verlag Chemie, Weinheim.

Pagano, D. (1991): "Intelligent tuning of PID controllers based on production rules system." In *Preprints IFAC International Symposium on Intelligent Tuning and Adaptive Control (ITAC 91)*, Singapore.

Palmor, Z. J. and R. Shinnar (1979): "Design of sampled data controllers." *Ind. Eng. Chem. Process Design and Development*, **18:1**, pp. 8–30.

Pao, H. H. (1990): "Use of neural-net technology in control: A survey and a perspective." In *Preprints 11th IFAC World Congress*, Tallinn, Estonia.

Passino, K. M. and P. J. Antsaklis, Eds. (1992): *An Introduction to Intelligent and Autonomous Control*. Kluwer Academic Publishers.

Patton, R. J., P. M. Frank, and R. N. Clark (1989): *Fault Diagnosis in Dynamic Systems, Theory and Applications*. Prentice-Hall, Englewood Cliffs, New Jersey.

Patwardhan, A. A., M. N. Karim, and R. Shah (1987): "Controller tuning by a least squares method." *AIChE Journal*, **33**, October, pp. 1735–1737.

Pemberton, T. J. (1972a): "PID: The logical control algorithm." *Control Engineering*, May, pp. 66–67.

Pemberton, T. J. (1972b): "PID: The logical control algorithm–II." *Control Engineering*, July, pp. 61–63.

Persson, P. (1992): *Towards Autonomous PID Control*. PhD thesis ISRN LUTFD2/TFRT--1037--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Persson, P. and K. J. Åström (1992): "Dominant pole design—A unified view of PID controller tuning." In *Preprints 4th IFAC Sym-*

posium on Adaptive Systems in Control and Signal Processing, pp. 127–132, Grenoble, France.

Persson, P. and K. J. Åström (1993): "PID control revisited." In *Preprints IFAC 12th World Congress*, Sydney, Australia.

Pessen, B. W. (1954): "How to 'tune in' a three mode controller." *Instrumentation*, Second Quarter, pp. 29–32.

Polonoyi, M. J. G. (1989): "PID controller tuning using standard form optimization." *Control Engineering*, March, pp. 102–106.

Porter, B., A. H. Jones, and C. B. McKeown (1987): "Real-time expert tuners for PI controllers." *IEE Proceedings Part D*, **134:4**, pp. 260–263.

Pottman, M. and D. E. Seborg (1993): "A radial basis function control strategy and its application to a pH neutralization process." In *Proceedings 2nd European Control Conference, ECC '93*, Groningen, The Netherlands.

Rad, A. B. and P. J. Gawthrop (1991): "Explicit PID self-tuning control for systems with unknown time delay." In *Preprints IFAC International Symposium on Intelligent Tuning and Adaptive Control (ITAC 91)*, volume 5, Singapore.

Radke, F. and R. Isermann (1987): "A parameter-adaptive PID controller with stepwise parameter optimization." *Automatica*, **23**, pp. 449–457.

Rake, H. (1980): "Step response and frequency response methods." *Automatica*, **16**, pp. 519–526.

Rijnsdorp, J. (1965a): "Interaction in two-variable control systems for distillation columns – I." *Automatica*, **1**, p. 15.

Rijnsdorp, J. (1965b): "Interaction in two-variable control systems for distillation columns – II." *Automatica*, **1**, pp. 29–51.

Rivera, D. E., M. Morari, and S. Skogestad (1986): "Internal model control—4. PID controller design." *Ind. Eng. Chem. Proc. Des. Dev.*, **25**, pp. 252–265.

Ross, C. W. (1977): "Evaluation of controllers for deadtime processes." *ISA Transactions*, **16:3**, pp. 25–34.

Rovira, A. A., P. W. Murrill, and C. L. Smith (1969): "Tuning controllers for setpoint changes." *Instruments and Control Systems*, December, pp. 67–69.

Rundqwist, L. (1990): "Anti-reset windup for PID controllers." In *Preprints 11th IFAC World Congress*, Tallinn, Estonia.

Schei, T. S. (1992): "A method for closed loop automatic tuning of PID controllers." *Automatica*, **28:3**, pp. 587–591.

Seborg, D. E., T. F. Edgar, and D. A. Mellichamp (1989): *Process*

Dynamics and Control. Wiley, New York.

Seborg, D. E., T. F. Edgar, and S. L. Shah (1986): "Adaptive control strategies for process control: A survey." *AIChE Journal*, **32**, pp. 881–913.

Shigemasa, T., Y. Iino, and M. Kanda (1987): "Two degrees of freedom PID auto-tuning controller." In *Proceedings of ISA Annual Conference*, pp. 703–711.

Shinskey, F. G. (1978): *Energy Conservation through Control*. Academic Press, New York.

Shinskey, F. G. (1981): *Controlling Multivariable Processes*. Instrument Society of America, Research Triangle Park, North Carolina.

Shinskey, F. G. (1988): *Process-Control Systems. Application, Design, and Tuning*. McGraw-Hill, New York, third edition.

Shinskey, F. G. (1990): "How good are our controllers in absolute performance and robustness?" *Measurement and Control*, **23**, May, pp. 114–121.

Shinskey, F. G. (1991a): "Evaluating feedback controllers challenges users and vendors." *Control Engineering*, September, pp. 75–78.

Shinskey, F. G. (1991b): "Model predictors: The first smart controllers." *Instruments and Control Systems*, September, pp. 49–52.

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

Smith, C. L., A. B. Corripio, and J. J. Martin (1975): "Controller tuning from simple process models." *Instrumentation Technology*, December, pp. 39–44.

Smith, C. L. and P. W. Murrill (1966): "A more precise method for tuning controllers." *ISA Journal*, May, pp. 50–58.

Smith, O. J. M. (1957): "Close control of loops with dead time." *Chemical Engineering Progress*, **53**, May, pp. 217–219.

Söderström, T. and P. Stoica (1988): *System Identification*. Prentice-Hall International, Hemel Hempstead, UK.

Stephanopoulos (1984): *Chemical Process Control. An Introduction to Theory and Practice*. Prentice-Hall.

Stock, J. T. (1987–88): "Pneumatic process controllers: The ancestry of the proportional-integral-derivative controller." *Trans. of the Newcomen Society*, **59**, pp. 15–29.

Stojić, M. R. and T. B. Petrović (1986): "Design of a digital PID standalone single-loop controller." *International Journal of Control*, **43:4**, pp. 1229–1242.

Strejc, V. (1959): "Näherungsverfahren für Aperiodische Übertragscharacteristiken." *Regelungstechnik*, **7:7**, pp. 124–128.

Suda, N. *et al.* (1992): *PID Control*. Asakura Shoten Co., Ltd., Japan.

Sugeno, M., Ed. (1985): *Industrial Applications of Fuzzy Control*. Elsevier Science Publishers BV, The Netherlands.

Swiniarski, R. W. (1991): "Neuromorphic self-tuning PID controller uses pattern recognition approach." In *Preprints IFAC International Symposium on Intelligent Tuning and Adaptive Control (ITAC 91)*, Singapore.

Takahashi, Y., M. J. Rabins, and D. M. Auslander (1972): *Control and Dynamic Systems*. Addison-Wesley, Reading, Massachusetts.

Takatsu, H., T. Kawano, and K. ichi Kitano (1991): "Intelligent self-tuning PID controller." In *Preprints IFAC International Symposium on Intelligent Tuning and Adaptive Control (ITAC 91)*, Singapore.

Tan, L.-Y. and T. W. Weber (1985): "Controller tuning of a third-order process under proportional-integral control." *Industrial & Engineering Chemistry Process Design and Development*, **24**, pp. 1155–1160.

Tong, R. M. (1977): "A control engineering review of fuzzy system." *Automatica*, **13**, pp. 559–569.

Tong, R. M. (1984): "A retrospective view of fuzzy control systems." *Fuzzy Sets and Systems*, **14**, pp. 199–210.

Truxal, J. (1955): *Automatic Feedback Control System Synthesis*. McGraw-Hill, New York.

Tucker, G. K. and D. M. Wills (1960): *A Simplified Technique of Control System Engineering*. Minneapolis-Honeywell Regulator Company, Philadelphia, Pennsylvania.

Tyreus, B. (1987): "TUNEX – an expert system for controller tuning." Technical Report, du Pont.

van der Grinten, P. M. E. M. (1963a): "Determining plant controllability." *Control Engineering*, October, pp. 87–89.

van der Grinten, P. M. E. M. (1963b): "Finding optimum controller settings." *Control Engineering*, December, pp. 51–56.

Voda, A. and I. D. Landau (1995): "A method for the auto-calibration of pid controllers." *Automatica*, **31:2**.

Walgama, K. S. and J. Sternby (1990): "Inherent observer property in a class of anti-windup compensators." *International Journal of Control*, **52:3**, pp. 705–724.

Wang, L.-X. (1994): *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Prentice Hall.

Webb, J. C. (1967): "Representative DDC systems." *Instruments & Control Systems*, **40**, October, pp. 78–83.

Wellstead, P. E. (1979): *Introduction to Physical System Modelling*. Academic Press.

Whatley, M. J. and D. C. Pott (1984): "Adaptive gain improves reactor control." *Hydrocarbon Processing*, May, pp. 75–78.

Willis, M. J., C. Di Massimo, G. A. Montague, M. T. Tham, and A. J. Morris (1991): "Artificial neural networks in process engineering." *IEE Proceedings D*, **138:3**, pp. 256–266.

Wills, D. M. (1962a): "A guide to controller tuning." *Control Engineering*, August, pp. 93–95.

Wills, D. M. (1962b): "Tuning maps for three-mode controllers." *Control Engineering*, April, pp. 104–108.

Wolfe, W. A. (1951): "Controller settings for optimum control." *Transactions of the ASME*, **64**, pp. 413–418.

Wong, S. K. P. and D. E. Seborg (1988): "Control strategy for single-input single-output non-linear systems with time delays." *International Journal of Control*, **48:6**, pp. 2303–2327.

Yamamoto, S. (1991): "Industrial developments in intelligent and adaptive control." In *Preprints IFAC International Symposium on Intelligent Tuning and Adaptive Control (ITAC 91)*, Singapore.

Yamamoto, S. and I. Hashimoto (1991): "Present status and future needs: The view from Japanese industry." In Arkun and Ray, Eds., *Chemical Process Control—CPCIV*. Proceedings of the Fourth Internation Conference on Chemical Process Control, Texas.

Yarber, W. H. (1984a): "Electromax V plus, A logical progression." In *Proceedings, Control Expo 84*.

Yarber, W. H. (1984b): "Single loop, self-tuning algorithm applied." In *Preprints AIChE Anaheim Symposium*.

Yuwana, M. and D. E. Seborg (1982): "A new method for on-line controller tuning." *AIChE Journal*, **28:3**, pp. 434–440.

Zadeh, L. A. (1988): "Fuzzy logic." *IEEE Computer*, April, pp. 83–93.

Zervos, C., P. R. Bélanger, and G. A. Dumont (1988): "On PID controller tuning using orthonormal series identification." *Automatica*, **24:2**, pp. 165–175.

Zhang, C. and R. J. Evans (1988): "Rate constrained adaptive control." *International Journal of Control*, **48:6**, pp. 2179–2187.

Zhuang, M. and D. Atherton (1991): "Tuning PID controllers with integral performance criteria." In *Control '91*, Heriot-Watt University, Edinburgh, UK.

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

Ziegler, J. G., N. B. Nichols, and N. Y. Rochester (1943): "Process lags in automatic-control circuits." *Trans. ASME*, **65**, July, pp. 433–444.