LUND UNIVERSITY

# Towards Autonomous PID Control

Persson, Per

1992

*Document Version:*
Publisher's PDF, also known as Version of record

Link to publication

*Citation for published version (APA):*
Persson, P. (1992). *Towards Autonomous PID Control*. [Doctoral Thesis (monograph), Department of Automatic Control]. Department of Automatic Control, Lund Institute of Technology (LTH).

*Total number of authors:*
1

**General rights**
Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: https://creativecommons.org/licenses/

**Take down policy**
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

# Towards Autonomous PID Control

*Per Persson*

Lund 1992

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

©1992 by Per Persson
Published 1992
Printed in Sweden by Studentlitteratur AB

# Preface

This thesis is a step towards obtaining controllers with an increased functionality. We call these autonomous controllers. The idea is to automate some of the functions that are manually performed by process engineers and control engineers and to include these functions in the controller. This thesis deals with some of the problems which will be encountered in the implementation of such controllers.

This thesis consists of two parts,

A the paper 'An Expert system Interface for an Identification Program'

B the monograph 'PID Controller Design.'

Part A is the Automatica paper [Larsson and Persson, 1991]. The paper describes the use of an expert system to supervise and help a user of the identification program Idpac. The help system is called (ihs), which stands for Intelligent Help System. The logo indicates that the system was written in Lisp. The (ihs) system was written in collaboration with Jan Eric Larsson. Both of us are equally responsible for all parts of the reports and articles produced in the project.

The project resulting in the Automatica paper was also presented at the American Control Conference 1986 (ACC 86) as [Larsson and Persson, 1986b]. The system (ihs) is described in detail in the licentiate thesis [Larsson and Persson, 1987b]. The thesis contains references to reports describing all technical details of the system. The paper [Larsson and Persson, 1988a] was presented at the European Conference on Artificial Intelligence 1988 (ECAI 88) in München. This is a conference aiming at the computer science and AI community. The system was also successfully demonstrated on a workstation at the conference. Different aspects of the use of expert systems in identification have been discussed in papers presented at workshops in Cambridge 1987, see [Larsson and Persson, 1987d], and in Swansea 1988, see [Larsson and Persson, 1988b]. The system has also been presented at the Swedish AI Society's annual meetings 1986 and 1987 (SAIS 86 and 87) as the papers [Larsson and Persson, 1986a] and [Larsson and Persson, 1987a]. The research results are published in a more popular version as [Larsson and Persson, 1987c].

Part B is a monograph describing a method for PID controller design. When the (ihs) project was finished, interest was focussed on other application areas of expert systems in automatic control. One idea was to write a self tuning PID controller based on expert system methods. When expert systems are used there is a need to write down all engineering knowledge of the problem and all rules of thumb very explicitly.

Automatic tuning has been introduced recently and is a useful step. However, it turns out that the design methods based on empirical tuning rules used in present autotuners are too simplistic. In [Åström *et al.*, 1992] it was attempted to improve the heuristic rules. This was not entirely successful. The need for further investigations of simple tuning methods for PID controllers soon became apparent. The Dominant Pole Design principle had been used earlier, see [Åström and Hägglund, 1985] and [Åström, 1988]. A more detailed investigation was now initiated. The results of the investigation are presented in the monograph 'PID Controller Design.'

Results leading to the work presented in this thesis have been presented 1989 at the IFAC Workshop on AI in Real-Time Control in Shenyang, China as [Åström *et al.*, 1989] and in the Automatica paper [Åström *et al.*, 1992]. Some of the results in this thesis will be presented at the International Symposium on Adaptive Systems in Control and Signal Processing (ACASP 92) in Grenoble, see [Persson and Åström, 1992].

### Acknowledgements

Professor Karl Johan Åström has been my supervisor in both of the projects described in this thesis. His enthusiasm for control theory, his support, and encouragement have been vital for the completion of this thesis. Needless to say, I have got many good ideas from discussions with him.

The licentiate thesis and the Automatica paper was written together with Jan Eric Larsson, M. Sc., L. Sc., B. A. *(Sic!)*. Collaboration with Jan Eric can be both rewarding and entertaining. I want to thank Jan Eric for the three years we spent together in the swamps of AI.

Docent Tore Hägglund and Dr. Karl-Erik Årzén have read and given valuable comments on the manuscript. I have very much appreciated Tore's practical experience of control and our enlightening discussions.

Professor Björn Wittenmark has handled all the necessary administration for my last year at the department, for which he is greatly acknowledged.

Civilingenjör Leif Andersson and Anders Blomdell are responsible for the very well managed computer facilities of the department. Leif and Anders have been very helpful with advice on the use of TEX and PostScript. The Perl script in Chapter 8 was written with assistance of Leif. Mrs. Britt-Marie Mårtensson from Eslöv is thanked for the artwork on pages 29 and 60 and for manipulations with the manuscript. Finally, I want to thank all the rest of the members of the department for contributing to the friendly atmosphere in which this work has been carried out.

The (ihs) project was a part of the Computer-Aided Control Engineering, CACE, project at the Department of Automatic Control, supported by STU, under contract no. 85–3042. The PID controller design project was funded by the NUTEK project 'Expertreglering' (Expert Control), project number NUTEK 728-91-1988.

P. P.

# An Expert System Interface for an Identification Program

# PID Controller Design

# 1

# Introduction

There are many approaches to control system design. It is often assumed that a model of the process to be controlled is available as a linear system. If the system is of order $n$ then the controller will be at least of order $n$, if we use the standard methods with observers and state feedback. Hence the complexity of the controller is directly related to the complexity of the process. However, it has been found empirically that complex processes often can be controlled quite well by low order controllers. To apply the standard methods it is then necessary to approximate the system by a model which complexity is compatible with, e.g., that of a PI controller. It is also possible to design a complicated controller and then approximate it with an appropriate PID controller. The design method which will be used here has a different character since it admits design of simple controllers for processes described by complicated models, e.g., partial differential equations.

In this chapter the goals of this thesis will be presented. The design method is presented and a number of considerations in PID controller design will be discussed briefly.

## 1.1 Background

A lot of work has been done in the area of PID control. In spite of this there are few comprehensive presentations. There are also few methods published for designing PID controllers based on general transfer functions. A large amount of literature has been reviewed to determine what has been done before and to find out if methods were available that could solve the problem.

PID controllers have been implemented using many different technologies. The first systems were based on pneumatics and hydraulics, discrete electronics and operational amplifiers were introduced later. Today most implementations are based on micro processors. This means that several short cuts and simplifications that were made previously are no longer necessary. Another important trend is that many PID controllers developed today has some kind of facility for autotuning, see e.g. [Åström and Hägglund, 1988]. To do this well it is necessary to have a deep insight into PID controllers and their tuning.

Despite the development in control theory PID controllers are the most used controllers in industry. Recent estimations from manufacturers in Japan indicate that between 85% and 95% of all control loops in Japanese industry are controlled by PID controllers, see [Yamamoto and Hashimoto, 1991].

In a plant with several hundred subprocesses and control loops there is no way to model each of them and custom-design controllers for each one. There is a need for a single controller structure with few parameters to tune.

A renewed interest in the issue of PID controller tuning has come with the attempts to use AI in control systems, see [Åström *et al.*, 1986] and [Årzén, 1987]. Systems of this kind gradually collects more and more information about the process to be controlled. At any stage the supervisory system tries to select a suitable controller and keep it as well tuned as possible. The application of such systems will be to keep a large number of loops reasonably tuned. The controllers will probably be existing controllers, i.e. PID controllers, and the tuning algorithms will be implemented in the supervisory system, see e.g. [Moore *et al.*, 1990]. In such systems there is a need for a simple design method which can handle a broad spectrum of dynamics uniformly.

## 1.2 The goal of the thesis

The goal of this work has been to develop methods for PID controller tuning, based on the knowledge of the transfer function of the plant. The methods we were looking for should have the property that they could be applied uniformly to all transfer functions. They should not have too many tuning parameters and should not require too much computation. In this work we will assume that the plant transfer function is known. Methods for obtaining these transfer functions will not be discussed.

Some investigations will be made on when a PID controller can be used and when it cannot be used. The benefits of PID controllers compared to PI controllers will also be examined. We will recognize which demands can be put on a PID controlled system.

A number of criteria are important in controller design, e.g.,

- set point response
- load disturbance attenuation
- low sensitivity to measurement noise
- low sensitivity to plant parameter variations.

The methods which will be presented will mainly try to achieve good load disturbance attenuation and a moderate sensitivity to plant parameter variations. It would be nice to take measurement noise into account in the controller design, but since the PI and PID controllers are controllers of very simple structure we cannot fulfill too many requirements.

We intend to present and investigate a method for designing controllers, in particular PI and PID controllers, based on the *Dominant Pole Design* (DPD) principle. The properties of this method will be investigated, with its possibilities and limitations. As a conclusion we will recommend a design method for a class of systems. Dominant Pole Design is a convenient and intuitively appealing way of parameterizing a linear controller. Given a process model on transfer function form we can compute the controller parameters directly, in terms of the dominant poles of the controlled system, without any form of model approximation, Taylor series expansions, etc. In short the problem is to determine the location of the dominant poles such that the closed loop system behaves sensibly.

This thesis has the aim to devise controller design methods, suitable for computer computations. To give a detailed analysis of the methods, other than for the simplest cases, is impossible due to the extreme complexity of the expressions involved. A large number of methods and examples have been explored during the work of this thesis. The possibility of doing this has drawn heavily on easy-to-use software and extensive calculations. The computations involved would not have been feasible on a VAX 11/780, the standard computational tool for a control laboratory ten years ago. Most of the computations have been carried out on a SPARCstation ELC.

We will try to keep as much as possible of the design in the frequency domain. This gives much less computations and the possibility to handle more general transfer functions. Simulations will mainly be used to verify the controller designs.

## 1.3 Drawbacks of existing methods

The review of the results in literature indicated that none of the methods published serve the purpose of a good PID controller tuning method.

- Many methods are too simplistic, they do not give closed loop systems with good performance. In some cases it is because the models used are too simple, in other cases because the models of the plants are not efficiently used.

- Many methods use process pole cancellation by controller zeros as the design principle. This may cause unnecessarily poor load disturbance attenuation.
- Many methods lack a good design parameter.

The DPD method uses a plant model on transfer function form for controller parameter computation. Various complexity of the model can be used. The DPD does not require a rational transfer function. Infinite dimensional systems as time delays, factors of the form $e^{-\sqrt{s}}$ etc., are allowed, and does not complicate the design process. In some cases the DPD may accidentally cancel process poles by controller zeros, but there is a great difference between that and using pole-zero cancellation as a design principle. In the following, design parameters which have the same impact on a large class of systems will be presented.

## 1.4 The contents of the thesis

Chapter 1 contains a short description of the background of the work and some motivations. In Chapter 2 design considerations are discussed and the notations used in the thesis will be introduced. Chapter 3 contains a review of previous work in the field of PID controller tuning. Chapter 4 and Chapter 5 describe the methods, design criteria, and algorithms for controller tuning which have been developed in this thesis. Chapter 6 discusses, mostly by examples, which information is needed to get models suitable for PI and PID controller design with the methods presented in the previous chapters. Chapter 7 gives examples of the tuning of systems and compares the presented method with different conventional methods. In Chapter 8 the computer tools used in the implementation of the methods in the thesis will be presented and discussed. Chapter 9 contains the conclusions and Chapter 10 contains the references.

# 2

# Design Considerations

In this chapter we will introduce the notations which will be used in this thesis. Some design considerations will also be discussed.

## 2.1 Notations

As far as possible the structure of the control system shown in Figure 2.1 will be used. The controller transfer function will be denoted $G_c(s)$ and the plant transfer function will be denoted $G(s)$. The loop transfer function is $L(s) = G(s)G_c(s)$. The signals have the following meaning:

$y_r(t)$ is the set point signal, which is supplied by the operator or by another controller.

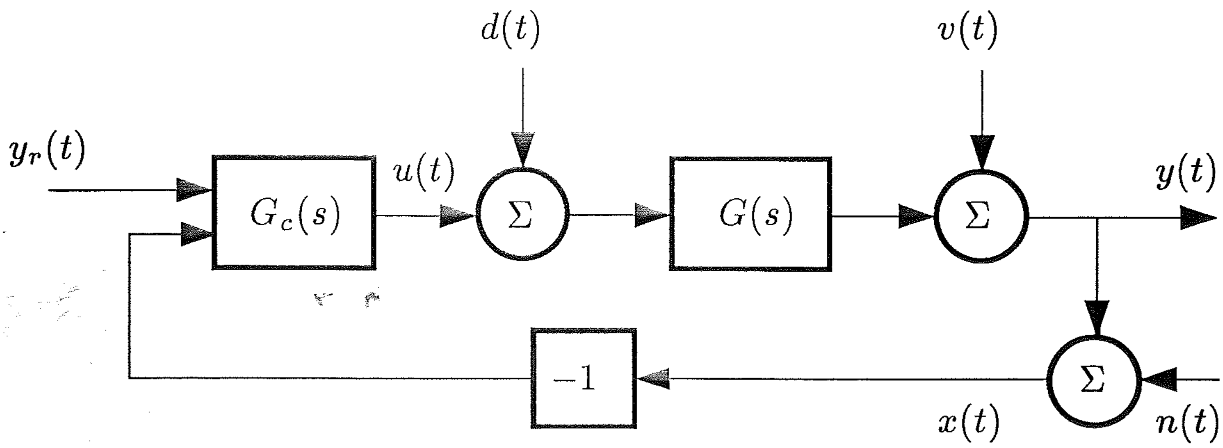



**Figure 2.1** The control system.

$d(t)$ is a disturbance on the input of the process. This is the standard model of plant disturbances. Physically it may, e.g., be an extra inflow to a level controlled tank.

$v(t)$ is a model of the process noise. Process noise can, of course enter the system in many places, e.g. inside $G(s)$, but this is easy for computations.

$y(t)$ is the actual process output, the signal we want to control.

$n(t)$ is the measurement noise. Notice the difference between measurement noise and process noise. The measurement noise does only exist in the controller. The process noise affects the output signal of the plant. Of course the measurement noise can also affect the process output, but only indirectly, via the controller.

$x(t)$ is the measurement signal, the signal we feed into the controller.

Laplace transforms are denoted by uppercase letters. The control error signal is defined as

$$E(s) = Y_r(s) - Y(s). \tag{2.1}$$

From the formula

$$Y(s) = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)} Y_r(s) + \frac{G(s)}{1 + G_c(s)G(s)} D(s) + \\ + \frac{1}{1 + G_c(s)G(s)} V(s) \tag{2.2}$$

we get

$$E(s) = \frac{1}{1 + G_c(s)G(s)} Y_r(s) \text{ if } D(s) = 0 \text{ and } V(s) = 0. \tag{2.3}$$

It is seen that $V(s)$ affects $Y(s)$ exactly as $Y_r(s)$ affects $E(s)$. It is very common in the literature to regard a step disturbance at $V(s)$ only, see, e.g., [Seborg *et al.*, 1989]. If the response to set point changes is also considered a load disturbance at $V(s)$ does not add anything to the controller evaluation. External disturbances may of course act on other points of the controlled system. In [Weber and Bhalodia, 1979] there is an investigation of the influence of disturbances acting at different places inside the process, but $D(s)$ and $V(s)$ represent the extreme cases, and nothing else will be considered. The sensitivity function of the closed loop system will be denoted

$$S(s) = \frac{1}{1 + L(s)}. \tag{2.4}$$

The maximum of the sensitivity function is defined as

$$M_s = \max_{\omega} |S(i\omega)| = |S(i\omega_s)|. \tag{2.5}$$

The general form of the PID controller we will consider is

$$U(s) = k_c(\beta Y_r(s) - Y(s) + \frac{1}{sT_i}(Y_r(s) - Y(s)) - \frac{sT_d}{1 + s\frac{T_d}{N}} Y(s)). \tag{2.6}$$

This is usually called the PID controller on parallel form. The parameter $\beta$ is the set point weighting factor. It can be used to change the overshoots of responses to set point changes. This is a two degree of freedom system since the set point and the measurement signal are treated differently. Some variants of the PID controller will be discussed in Chapter 3. The conventional and compact way of writing PID-controllers is

$$G_{\mathrm{PI}}(s) = k_c(1 + \frac{1}{sT_i}) = k + \frac{k_i}{s} \tag{2.7}$$

$$G_{\mathrm{PID}}(s) = k_c(1 + \frac{1}{sT_i} + sT_d) = k + \frac{k_i}{s} + sk_d \tag{2.8}$$

$$G_{\mathrm{PIDF}}(s) = k_c(1 + \frac{1}{sT_i} + \frac{sT_d}{1 + \frac{sT_d}{N}}). \tag{2.9}$$

The controller $G_{\mathrm{PIDF}}(s)$ has a filter in the derivative part to make the controller physically realisable. This does not cover all aspects of PID control, since these formulas indicate that measurement and set point values are treated equally, which is not the case for the normal implementation. In some cases (e.g., $\beta = 1$ and $T_d = 0$) the compact form $G_{\mathrm{PI}}(s)$ is the same as the "true" form, (2.9). The controllers $G_{\mathrm{PI}}$, $G_{\mathrm{PID}}$, and $G_{\mathrm{PIDF}}$ will still be used frequently since the poles of the closed loop system will be in the same places as with (2.9). Only a zero of the closed loop system will be affected. Furthermore, $y_r$ and $\beta$ do not affect the response to load disturbances.

## 2.2 The problem setting

Our goal has been to give a method to tune a *standard* PID controller, given the transfer function of the system. When we design controllers we must consider

- set point changes
- load disturbances
- noise

- sensitivity to process variations.

We will handle load disturbances by minimizing the error integral when the system is subject to a step load disturbance. To cope with process variations we will require certain maximum value of the process sensitivity function. Finally, set point changes will be shaped by set point weighting. The only thing we do not try to handle explicitly is the noise. However, the noise comes into the picture when the filter in the derivative part is chosen. As far as we are concerned, load disturbance attenuation and insensitivity to process variations will be our main objectives in controller design. Load disturbances are considered to be the most common disturbance in process industry. This implies that if we do not get the response that we want for a change in the set point signal we have to make another design and get a worse response to load disturbances and plant variations.

This method agrees well with the principles in, e.g., [Horowitz, 1963], where the feedback is used to handle load disturbances and process uncertainty, and feed forward is used to shape the response to the set point signal.

A PID controller is a simple controller structure and this implies that there is a limit of what we can achieve with it. This rules out attempts to get high performance from systems with very low damping or from systems with multiple resonances.

The systems we can handle successfylly with PID controllers are mainly plants from process industry, i.e., systems which are dominated by time delays and non-resonant lags. Furthermore, only single-input-single-output (SISO) systems will be considered.

### Load disturbance criteria

A number of penalty functions have been suggested in optimization of PID controllers. The most used are

Integrated Error:

$$\mathrm{IE} = \int_0^\infty e(t)\,dt = \frac{1}{k_i} \tag{2.10}$$

Integrated Absolute Error:

$$\mathrm{IAE} = \int_0^\infty |e(t)|\,dt \tag{2.11}$$

Integrated Square Error:

$$\mathrm{ISE} = \int_0^\infty e(t)^2\,dt \tag{2.12}$$

Integrated Time multiplied Square Error:

$$\mathrm{IT^2SE} = \int_0^\infty (te(t))^2\,dt \tag{2.13}$$

Integrated Time multiplied Absolute Error:

$$\mathrm{ITAE} = \int_0^\infty t|e(t)|\,dt. \tag{2.14}$$

The only cost function of these which, in the general case, can be evaluated without numerical solution of differential equations is IE. The transfer function from load disturbance to process output is

$$Y(s) = \frac{G(s)}{1 + G_c(s)G(s)} D(s). \tag{2.15}$$

Let $G_c(s) = k_d s + k + k_i/s$ and $D(s) = 1/s$, a step load disturbance. Then we get from the final value theorem

$$\mathrm{IE} = \lim_{t\to\infty} \int_0^t y(\tau)\,d\tau = \lim_{s\to 0} s \frac{G(s)}{1 + (k_d s + k + k_i/s)G(s)} \frac{1}{s^2} = \lim_{s\to 0} \frac{1}{sG^{-1}(s) + ks + k_i + k_d s^2} = \frac{1}{k_i}. \tag{2.16}$$

If

$$\lim_{s\to 0} \frac{s}{G(s)} = 0, \tag{2.17}$$

i.e., if $G$ does not contain any differentiators. This important relation makes it possible to compute a cost functional by using only the controller parameters. This relation will be used in the following as an optimization criterion.

# 3

# Previous work

In this chapter we will review some of the work which has been done in the area of PID controller tuning. A number of different approaches to the tuning of PID controllers have been taken. They have here been divided into the heuristic, the analytical, the frequency domain, and the optimization approach.

Most of this work was done between 1940 and 1970. Lately there has been a renewed interest in PID controller tuning due to the use of expert systems and smart controllers. When these systems are used there is a need to automate the tuning of simple controllers. Rules of thumb and control engineers' intuition must be formalized and written down as algorithms and rules. This requires better insight into the design methods, and methods which can cope with a greater variety of process dynamics.

There has also been a renewed interest in tuning of simple loops due to new autotuning techniques, see [Åström and Hägglund, 1984]. In these methods a simple model is estimated by a simple and robust method, the model is then the base for design of a PID controller.

Since the PID controller is the most common controller in process industry there has been research in methods for automatic tuning of such controllers. In [Yuwana and Seborg, 1982], [Lee, 1989], and [Lee *et al.*, 1990] a poorly controlled loop is given a step in the set point signal and an improved controller is computed from the shape of the response. In [Åström and Hägglund, 1988] certain points on the Nyquist curve of a system are computed from the oscillations obtained when the system is controlled by relay feedback. The estimated frequency response is then used for controller design.

## 3.1 Heuristic methods

In 1942 Ziegler and Nichols (ZN) published the classic paper "Optimum Settings for Automatic Controllers," [Ziegler and Nichols, 1942]. For the first time rules of thumb for tuning a PID controller were published. Some interesting historical notes from an interview with Mr. Ziegler are given in [Blickley, 1990]. By simple experiments with the process dynamics Ziegler and Nichols determined suitable values for the parameters of a PID controller for good attenuation of load disturbances. Ziegler and Nichols made two kinds of experiments. In one experiment they tried to determine the time constant and time delay from a step response of the open system. The second experiment was carried out under proportional control and aimed at finding the ultimate gain and frequency of the system. Quarter amplitude damping of the response to a step disturbance on the output of the controller was considered to be the good behaviour of the closed system. The controller settings were found from experiments on physical processes.

The key idea behind the ZN methods is to characterize the process dynamics with a few parameters. Controller parameters are then given as functions of these parameters. Only two parameters are used in the ZN methods.

In the open loop method the parameters are the apparent dead-time and the maximum slope. In the closed loop method (the self oscillation method) the parameters are the ultimate gain and the ultimate period. Since a PID controller has three parameters the additional condition $T_i = 4T_d$ is also used. The ZN method is very simple and surprisingly effective. It is the basis of many practical tuning procedures currently used in industry.

The ZN methods give only approximate values of the controller parameters. Often it is necessary to adjust the controllers manually for fine tuning. For this reason there have been many attempts to improve the ZN tuning methods. To do this it is necessary to introduce more features. It is also common to distinguish between stable processes and processes with integration. Some recent tuning rules of this type are given in [Hang *et al.*, 1991].

## 3.2 Analytical methods

Most of the tuning methods presented are based on analytical methods. The process is then described by a transfer function. In this section some of the tuning methods based on analytical methods will be described.

**Pole placement methods**

An early attempt to analyze a process controlled by a PID controller can be found in [Callender *et al.*, 1936] and [Hartree *et al.*, 1937]. In [Hartree *et al.*, 1937] the process

$$G(s) = K\frac{e^{-sL}}{1+sT} \tag{3.1}$$

is studied. A PID controller with the structure

$$U(s) = \frac{b_1 s + b_2}{s + b_1} \cdot \frac{n_1 + n_2 s + n_3 s^2}{s} E(s), \tag{3.2}$$

is used. By specifying the roots of the characteristic equation of the closed system, the parameters of the controller can be determined by solving a nonlinear equation. The poles of the closed loop system were placed in $\alpha \pm i\beta$. The parameters $\alpha$ and $\beta$ are chosen so that the load response would go to zero "as rapid as possible." We quote from [Hartree *et al.*, 1937]: (the parameters $\nu_1 = n_1 T^2$ and $\nu_2 = n_2 T$ have been introduced)

> "The best values for the control parameters are those for which $\alpha$ is fairly high both for the fundamental and for the subharmonic, and $\beta$ for the subharmonic is not too small. But the neighborhood of the saddle point should be avoided, as the values for $\alpha$ and $\beta$ are there very sensitive to the values of $\nu_1$ and $\nu_2$ so that the control would not be flexible (in the sense of giving good control over a range of values of the control parameters)."

It is interesting to note that Hartree *et al.* realized the importance of choosing the controller parameters with respect to the robustness of the closed system.

Hartree gives a diagram with $\alpha$ and $\beta$ on the axes and with $\nu_1$ and $\nu_2$ contours in it. The $\nu_1$ and $\nu_2$ parameters are the controller parameters. Changing the controller parameters corresponds to changing $\alpha$ and $\beta$, i.e., the position of the closed loop dominant poles. Changing the process parameters also corresponds to changing the $\nu_1$ and $\nu_2$ contours. The only thing we can say for certain from the diagram is how the change of the controller parameters affects the closed loop dominant poles. This was of course an important issue when the parameters of the controller could vary with time.

In [Cohen and Coon, 1953] tuning formulas for P, PD, PI, and PID controllers are derived based on the process model (3.1). Cohen and Coon consider disturbances on the process input. In the case of a P controller the solutions are required to have "quarter amplitude damping" (QAD). The PD controllers are specified by QAD and minimum offset for a step change in the reference value. PI controllers are specified by QAD and minimization of the IE criterion. PID controllers are determined by three poles, two poles are placed to get a QAD mode, one real pole is placed on the negative real axis at the same distance as the two complex ones and the distance of the poles from the origin is obtained by minimizing IE. As a result of their investigations Cohen and Coon present a number of formulas for the tuning of PID controllers. The tuning rules of Cohen and Coon come close to the ZN rules when $L/T$ is small. The method presented in [Smith and Murrill, 1966] follows the ideas of [Cohen and Coon, 1953].

A number of design methods are based on algebraic manipulations of a transfer function. The approach taken in [van der Grinten, 1963] is: compute an ideal controller from the desired response and a given model, then approximate the controller so that it fits into the PID-structure. This often boils down to cancellation of process poles by controller zeros. Similar ideas can be found in [Haalman, 1965], where it is assumed that

- controllers and processes can be matched in such a way that the open-loop transfer function is the same for all combinations
- a purely integral controller yields good control for a purely dead-time process.

Process poles are then cancelled by controller zeros, so that the resulting loop-transfer function will be $L(s) = ke^{-sT}/s$, with $k = 2/(3T)$. This method has recently been used in [Rad and Gawthrop, 1991] for autotuning purposes. The design methods in [Pemberton, 1972] and [Smith *et al.*, 1975] use roughly the same principles.

The danger of cancelling process poles by controller zeros is that controllability is lost which implies that the controller does not act on the modes corresponding to cancelled poles. The approach is particularly bad if the process has slow poles. The response to load disturbances is then quite sluggish. The drawback of cancellation of process poles has been pointed out many times [Shinskey, 1988], [Hang, 1989], and [Clark, 1988] but the approach keeps reappearing in the literature. According to Hang problems may arise when slow dynamics is cancelled and we have a disturbance acting on the input of the process. Clark claims that we may get large transient errors when we have limitations on the control signal. In [Truxal, 1955] pp. 303–305 a number of reasons are given why one should not cancel process poles with controller zeros.

The 'Dominant Pole Design' (DPD) is also a method based on pole placement. The method was first suggested in [Åström and Hägglund, 1985] and further explored in [Åström, 1988]. The idea have been used earlier for special plants, e.g., in [Cohen and Coon, 1953] and [Hartree *et al.*, 1937], but not as a general method. The method will be further explored in this thesis.

### The internal model principle

The internal model principle, see [Rivera *et al.*, 1986] has also been applied to the design of PID controllers. Controllers obtained by this method have the transfer function

$$G_c = \frac{G_f G_p^\dagger}{1 - G_f G_p^\dagger G}, \tag{3.3}$$

where $G_p^\dagger$ is an approximate inverse of the process transfer function and $G_f$ is a low pass filter, typically $G_f(s) = 1/(1 + sT_f)$. It follows from (3.3) that controllers of this type will always attempt to cancel poles of the plant. This is illustrated by an example.

EXAMPLE 3.1—Internal model principle
For a process with the transfer function

$$G(s) = \frac{k_p}{1 + sT} e^{-sL} \tag{3.4}$$

an approximate inverse is given by

$$G_p^\dagger(s) = \frac{1 + sT}{k_p}. \tag{3.5}$$

With the low pass filter $G_f(s) = 1/(1 + sT_f)$ and the approximation $e^{-sL} \approx 1 - sL$ the controller becomes

$$G_c(s) = \frac{1 + sT}{k_p s(L + T_f)}, \tag{3.6}$$

which is a PI controller. With the first order Padé approximation

$$e^{-sL} = \frac{1 - sL/2}{1 + sL/2} \tag{3.7}$$

we get the PID controller

$$G_c(s) = \frac{(1 + sL/2)(1 + sT)}{k_p s(L + T_f + sT_f L/2)} \approx \frac{(1 + sL/2)(1 + sT)}{k_p s(L + T_f)}. \tag{3.8}$$

This way of approximating transfer functions and controllers is typical for IMC and the design methods which are based on algebraic manipulations.

□

### Modulus and symmetrical optimum

Modulus Optimum (BO) and Symmetrical Optimum (SO) are two methods for selecting and tuning controllers. The methods are extensively used by Siemens. The acronyms BO and SO are derived from the German words Betragsoptimum and Symmetrische Optimum, see [Kessler, 1958a], [Kessler, 1958b], [Fröhr, 1967], and [Fröhr and Orttenburger, 1982]. In spite of their names these methods are algebraic pole placement methods. The methods are based on simple analytical calculations and some heuristic rules. The methods are interesting because they form a methodology to design simple controllers.

The key idea is to shape the loop transfer function $L(s) = G_c(s)G(s)$ to approximate either

$$L_2 = \frac{\omega^2}{s(s + 2\zeta\omega)} \tag{3.9}$$

for BO or

$$L_3 = \frac{\omega^2(2s + \omega)}{s^2(s + 2\omega)} \tag{3.10}$$

for SO. The name symmetrical optimum derives from the fact that the Bode diagram is symmetrical around $s = i\omega$ for $L_3(s)$. The motivation for these choices is to make the closed loop transfer function $L/(1 + L)$ maximally flat at the origin. The design method corresponds to a pole placement method with the characteristic polynomial $s^2 + \sqrt{2}\omega s + \omega^2$ for BO and $(s + \omega)(s^2 + \omega s + \omega^2)$ for SO. The SO method gives a command response with quite a large overshoot. In this case a prefilter is introduced to reduce the overshoot.

The design method consists of two steps. In the first step the process transfer function is simplified to one of the following forms

$$G_1(s) = \frac{1}{1 + sT} \tag{3.11}$$

$$G_2(s) = \frac{1}{(1 + sT_1)(1 + sT_2)}, \quad T_1 > T_2 \tag{3.12}$$

$$G_3(s) = \frac{1}{s(1 + sT)} \tag{3.13}$$

$$G_4(s) = \frac{1}{s(1 + sT_1)(1 + sT_2)}, \quad T_1 > T_2. \tag{3.14}$$

Process poles are cancelled by controller zeros to obtain the desired loop transfer function. A slow pole may be approximated by an integrator and fast poles may be lumped together. The response to load disturbances is sometimes poor because process poles are cancelled. A summary of some of the design rules are given in Table 3.1.

**Table 3.1**

| Process | Controller | Remark | L | Method |
|---|---|---|---|---|
| 1 | I | | $L_2$ | BO |
| 2 | PI | Cancel $T_1$ | $L_2$ | BO |
| 2 | P | $1+sT_1 \approx sT_1$ | $L_2$ | BO |
| 2 | PI | $1+sT_1 \approx sT_1$ | $L_3$ | SO |
| 3 | P | | $L_2$ | BO |
| 3 | PI | | $L_3$ | SO |
| 4 | PD | Cancel $T_1$ | $L_2$ | BO |
| 4 | PID | Cancel $T_1$ | $L_3$ | SO |
| 4 | PD | $1+sT_1 \approx sT_1$ | $L_2$ | SO |

These design principles can be applied to other processes as well. If we want to make the frequency response curve of

$$|G_{\mathrm{CL}}(i\omega)| = |\frac{G_c(i\omega)G(i\omega)}{1+G_c(i\omega)G(i\omega)}| \tag{3.15}$$

as flat as possible in the origin one can proceed as in Example 3.2.

EXAMPLE 3.2—Time delay and first order lag
For a BO design, cancel as many poles as possible with controller zeros, then use the rest of the parameters to zero out as many coefficients of $\omega$ as possible in (3.15). For a SO design do not cancel any poles directly, but zero out as many coefficients of $\omega$ as possible in the Taylor series expansion of the numerator and the denominator of (3.15). This involves much more complicated computations. Consider the process

$$G(s) = \frac{e^{-sL}}{1+sT} \tag{3.16}$$

controlled by

$$G_c(s) = k_c(1+\frac{1}{sT_i}). \tag{3.17}$$

***BO for PI control*** We immediately get $T_i = T$ from process pole cancelling by the controller zero, and $k_c = T/(2L)$ from setting the coefficient of $\omega^2$ in the denominator of $|G_{\mathrm{CL}}(i\omega)|$ to zero.

***SO for PI control*** The parameters $k_c$ and $T_i$ are obtained from setting the coefficient of $\omega^2$ and $\omega^4$ in the denominator of $|G_{\mathrm{CL}}(i\omega)|$ to zero. This leads to the equations

$$\begin{cases} (k_c^2+1+2k_c)T_i + (-2k_cT - 2Lk_c) = 0 \\ (-L^2k_c + T^2 - 2Lk_cT)T_i + (L^3k_c/3 + TL^2k_c) = 0. \end{cases} \tag{3.18}$$

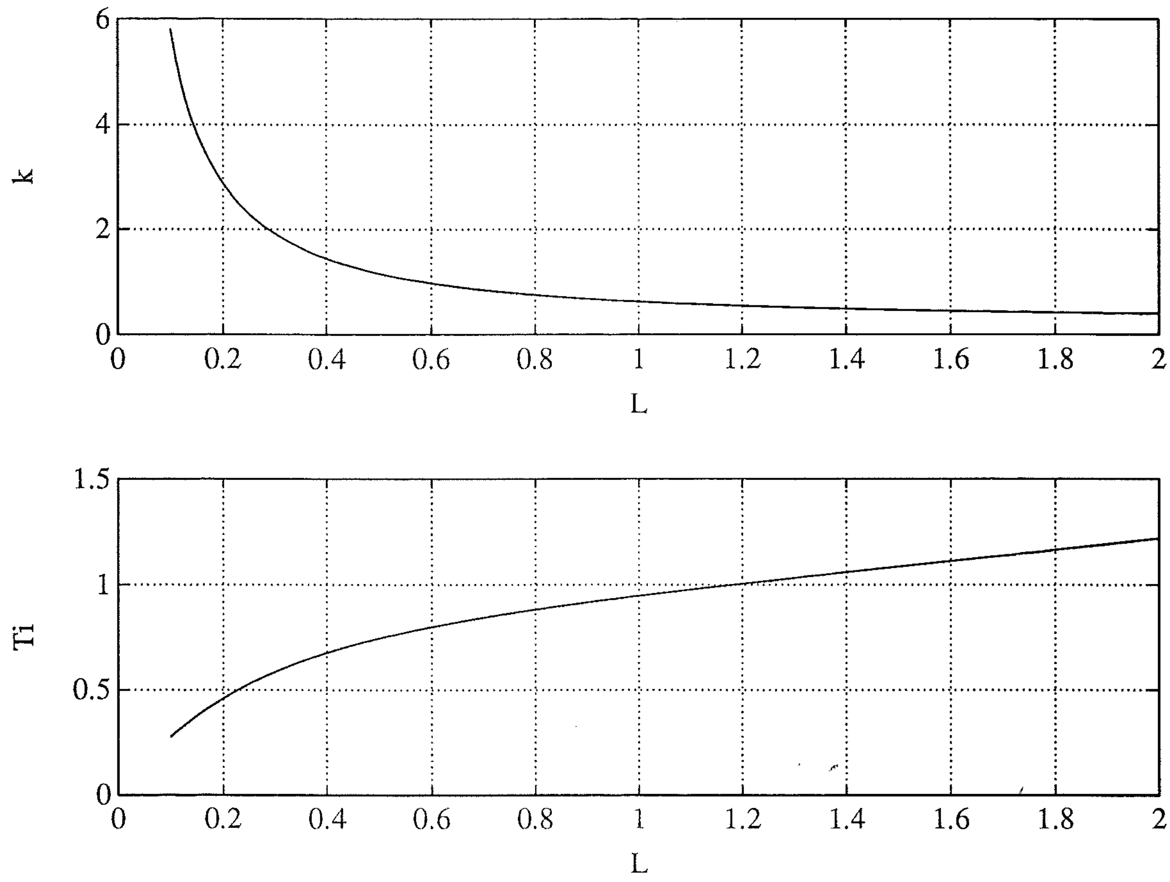



**Figure 3.1** Controller parameters for $T = 1$ as function of $L$. Upper figure: $k_c$, lower figure: $T_i$

The parameters $k_c$ and $T_i$ are plotted in Figure 3.1 for the case $T = 1$ and $L$ varying from 0.1 to 2. This method gives reasonable time responses of the closed loop system. □

## 3.3 Non parametric frequency domain methods

Most standard text books on control give methods for computing PID controllers from information in Bode or Nyquist diagrams, e.g., amplitude and phase margins. See, e.g., [Truxal, 1955], [Åström, 1976], and [Seborg *et al.*, 1989]. Recently [Lee *et al.*, 1990] has published a method which uses $M_p$ as design specification.

Another way of designing PI controllers can be found in [Buckley, 1964]. The method is quite simple and based on a Nichols diagram: choose controller gain $k_c$ to get a reasonable $M_p$, then choose $T_i$ to get an additional phase lag of $5° - 10°$ at the frequency corresponding to the $M_p$ peak.

In a recent thesis, [Thomas, 1990], the properties of Nyquist and Bode curves are used to design PID controllers. Thomas has also made some analysis on the Ziegler-Nichols design methods.

## 3.4 Optimization methods

Many attempts have been made to tune PID controllers with optimization techniques. A number of different optimization criteria have been used. The most important are IE, IAE, ISE, ITAE, and IT$^2$SE, as introduced in Chapter 2.

As a part of specifying the optimization criteria it is also important to specify the disturbances we are optimizing for. Good discussions of the physical significance of different optimization criteria can be found in [Shinskey, 1988] and [Shinskey, 1990]. Most of the interpretations come from applications in the process industry, where PID controllers have been widely used. According to Shinskey the key concerns in PID control are

- efficient load attenuation, this means that disturbances on the process input should be considered, not responses to set point changes
- minimization of IAE
- minimization of peak deviation, $e_p$, at a load disturbance.

Shinskey gives octane control in gasoline production as a typical example of when the IAE criterion is relevant. The IE criterion is relevant, e.g., when the output of the controlled process leads into a large storage tank (an integrator) with good mixing and the concentration of the product in the tank is to be controlled to get a product of a specified quality.

The desire to minimize the total control energy may lead to the use of ISE criterion. Similar arguments have been used in the LQG literature, see [Kalman, 1960]. An interpretation of PI design in terms of LQG can be found in [Athans, 1971] and in [Marsili-Libelli, 1981].

The criteria ITAE and IT$^2$SE have in general no physical significance.

Almost all optimization has been done numerically. It is possible to find analytical expressions for the cost functional ISE, see [Åström, 1970] and [Walton and Marshall, 1984] and the functional IT$^2$SE, see [Zhuang and Atherton, 1991] and [Zhuang, 1991]. This simplifies the computation of the integrals, but the actual optimization must, in the general case, be carried out numerically. The IAE cannot be computed analytically, the cost functional must therefore be computed by numerical solution of the underlying differential equations. Analytical expressions for IE are easily obtained, and the minimization of IE can be carried out analytically in certain cases.

Each optimization is carried out for an external disturbance of the process. The most common disturbance is a change in the set point signal or a load disturbance on the process input. Investigations of random loads have also been carried out. Different results are obtained depending on where in the process a disturbance acts, see [Weber and Bhalodia, 1979].

In [Ziegler and Nichols, 1943] PID controller settings are made based on the IAE criterion. Ziegler and Nichols observe that IAE of a load disturbance is approximately $\text{IAE} \approx \text{IE} = \int_0^\infty e(t)\,dt = 1/k_i$ for a well damped system.

In [Hazebroek and van der Waerden, 1950b] and [Hazebroek and van der Waerden, 1950a] controllers are designed by minimizing an ISE criterion numerically. Formulas for $k_c$ and $T_i$ are obtained as a result. The formulas are similar to Ziegler-Nichol's formulas, but the $k$ and $T_i$ are more complicated functions of $L$ and $T$.

In [Wolfe, 1951] the system

$$G(s) = k\frac{e^{-sL}}{s} \tag{3.19}$$

controlled by a PI controller was investigated. The same the same pole placement technique was used as in [Hartree *et al.*, 1937]. However, Wolfe chose the parameters of the controller (or the locations of the poles) by minimizing the IE criterion, when the system is subject to a disturbance at the input of the process.

In [Oldenburg and Sartorius, 1954] a number of different optimization criteria are investigated, e.g., IE, ISE, and $\int_0^\infty J(\omega^2)\,d\omega$, i.e., an integral of some kind of frequency spectrum.

In [Lopez *et al.*, 1967] computations can be found of standard PI and PID controllers when a system with the structure of (3.1) is subject to step load disturbances at the process input, the IAE, ISE, and the ITAE methods are used in the optimization. A good comparison of these methods can be found in [Miller *et al.*, 1967]. In [Rovira *et al.*, 1969] similar investigation are made, but optimized for a step disturbance at the set point input. In [Sood and Huddleston, 1973] PI controller settings are computed with the IAE criterion for the case when random disturbances act on the process input. Their conclusions are

> "It is perhaps best not to generalize the results however, other than to say that if load changes appear in the system at intervals less than the response time of the system, some of the integral action of the controller should be substituted by proportional action."

Various tuning formulas have been represented as 'tuning maps' to aid the process operator in tuning controllers, see, e.g., [Wills, 1962] and material produced by Foxboro, e.g., [Fox, 1979]. Many of the optimization results were computed in the sixties with early digital, analog, or hybrid equipment. The minima of the integral criteria are rather flat, so there may be difficulties in obtaining accurate results, if high precision is not used in the numerical computations. Results from different sources differ slightly, approximately 1 – 5%. The formulas presented for controller parameter settings are least square approximations of the optimization results. The formulas are only valid for relatively short time delays, $L < T$, and should be used with caution.

In [Harris and Mellichamp, 1985] the optimization of an 'Index of Performance' as a design method is proposed. As an example they suggest

$$\mathrm{IP} = w_m \frac{|M_p - M_{pd}|}{M_{pd}} + w_f/\omega_r + \frac{|\varphi_m - \varphi_{md}|}{\varphi_{md}}, \tag{3.20}$$

where $M_p$ is the resonant peak ratio, and $M_{pd}$ its desired value, $\omega_r$ is the resonant frequency, $\varphi_m$ is the phase margin and $\varphi_{md}$ is its desired value. The parameters $w_m$ and $w_f$ are weighting factors. This is, of course, a very general method. The results are obtained from numerical optimization, and requires reliable and efficient software. It is impossible to make any statements about the existence of optima, stability of solutions etc., in the general case.

## 3.5 Structures of PID controllers

Several different PID controller structures have been proposed, in addition to the PID controller presented in Chapter 2. Some are mere reparametrizations of (2.6). The standard controller can also be written

$$G'_{\mathrm{PID}}(s) = k'_c(1 + \frac{1}{sT'_i})(1 + \frac{sT'_d}{1 + \frac{sT'_d}{N'}}). \tag{3.21}$$

This is called the series form of a PID controller. The standard controller, (2.6) may have complex zeros in the controller, which cannot happen with (3.21). Thus (2.6) is more general than (3.21). The relation between the parameters of the two controllers are

$$k_c = k'_c \frac{T'_d + T'_i}{T'_i} \tag{3.22}$$

$$T_i = T'_i + T'_d \tag{3.23}$$

$$T_d = \frac{T'_d(T'_i N' - T'_d)}{N'(T'_i + T'_d)} \tag{3.24}$$

$$N = \frac{T'_i N' - T'_d}{T'_d + T'_i}. \tag{3.25}$$

The inverse transformation is

$$k'_c = Z \tag{3.26}$$

$$T'_i = \frac{Z T_i}{k_c} \tag{3.27}$$

$$T'_d = \frac{T_i(k_c - Z)}{k_c} \tag{3.28}$$

$$N' = \frac{k_c(1 + N) - Z}{Z} \tag{3.29}$$

where $Z$ is

$$Z = k_c \frac{T_d + T_i N \pm \sqrt{(T_d + T_i N)^2 - 4T_i T_d N(1+N)}}{2T_i N}. \tag{3.30}$$

Hence for certain values of $k$, $T_i$, $T_d$, and $N$, the parameter $Z$, may become complex and there is no physically realizable solution on the series form. The controller structure

$$G_c = k \frac{(1 + \frac{1}{sT_i})(1 + sT_d)}{(1 + \frac{sT_d}{N})} \tag{3.31}$$

has also been suggested.

In [Frank, 1968] the PIS$_m$ controller is suggested for dealing with systems with time-delays. The PIS$_m$ controller has the structure

$$R_m(s) = k_c + \frac{k_i}{s} + \sum_{\nu=1}^{m} \frac{k_{s_\nu} s}{s^2 + \omega_\nu^2}, \tag{3.32}$$

which can be interpreted as an approximation of a Smith predictor.

The modified PID controller

$$G_c(s) = k_c(\beta_p Y_r(s) - Y(s) + \frac{Y_r(s) - Y(s)}{sT_i} + \frac{s(\beta_d Y_r(s) - Y(s))}{1 + s\frac{T_d}{N}}) \tag{3.33}$$

is examined in [Eitelberg, 1987], [Hippe *et al.*, 1987], and in [Wurmthaler and Hippe, 1974]. In [Mantz and Tacconi, 1989] these additional degrees of freedom are used to obtain a both tracking and regulating controller. Based on calculations with the Bode integrals Mantz and Tacconi suggest $\beta_p = 0.17$ and $\beta_d = 0.654$ as appropriate values. However, there will be a large 'derivative kick' in the control signal, due to the presence of $\beta_d$. In all realistic controllers we should have $\beta_d = 0$. This method can also be interpreted as cancellation of closed loop oscillating poles with controller zeros. The advantage with the alternative PID structures is that we can get better performance from the system. For example, we may get good disturbance attenuation and good set point following at the same time. The disadvantages are that there are more parameters to tune and that these control algorithms are not usually implemented in commercially available controllers. For a good compilation of properties of commercial controllers, see [Åström and Hägglund, 1993].

The alternative PID controllers may have some theoretical interest, but if one wants to get more performance out of a controller one should not restrict oneself to PID controllers, but look for more general controllers.

## 3.6 Summary of tuning formulas

In this section we present a compilation of many common tuning formulas. Most of the tuning formulas apply to the process

$$G(s) = k\frac{e^{-sL}}{1+sT}. \tag{3.34}$$

We assume that the controller is written as

$$G_c(s) = k_c(1 + \frac{1}{sT_i} + sT_d). \tag{3.35}$$

The ultimate period and the ultimate gain of a process is denoted $T_u$ and $k_u$, and are defined by $\arg G(2\pi i/T_u) = -\pi$ and $k_u = 1/|G(2\pi i/T_u)|$. The normalized dead time is defined as $\theta = L/T$ and the normalized process gain as $\kappa = kk_u$.

### P control

| Method | $k_c k$ |
|---|---|
| ZN step | $\left(\frac{L}{T}\right)^{-1}$ |
| ZN osc. | $0.5k_u$ |
| Cohen-Coon | $\left(\frac{L}{T}\right)^{-1} + 0.35$ |
| ISE | $1.411(\frac{L}{T})^{-0.917}$ |
| IAE | $0.902(\frac{L}{T})^{-0.985}$ |
| ITAE | $0.490(\frac{L}{T})^{-1.084}$ |

**PI control**

| Method | $k_c k$ | $T_i$ |
|---|---|---|
| ZN step | $0.9(\frac{L}{T})^{-1}$ | $L/0.3$ |
| ZN osc. | $0.45k_u$ | $T_u/1.2$ |
| Cohen-Coon | $0.9(\frac{L}{T})^{-1}+0.083$ | $\frac{T(0.9\frac{L}{T}+0.083(\frac{L}{T})^2)}{0.27+0.6(\frac{L}{T})}$ |
| ISE load | $1.305(\frac{L}{T})^{-0.959}$ | $2.033T(\frac{L}{T})^{0.739}$ |
| IAE load | $0.984(\frac{L}{T})^{-0.986}$ | $1.644T(\frac{L}{T})^{0.707}$ |
| ITAE load | $0.859(\frac{L}{T})^{-0.977}$ | $1.484T(\frac{L}{T})^{0.680}$ |
| ISE set point | $0.980(\frac{L}{T})^{-0.892}$ | $\frac{T}{0.690-0.155(\frac{L}{T})}$ |
| IAE set point | $0.758(\frac{L}{T})^{-0.861}$ | $\frac{T}{1.02-0.323(\frac{L}{T})}$ |
| ITAE set point | $0.586(\frac{L}{T})^{-0.916}$ | $\frac{T}{1.03-0.165(\frac{L}{T})}$ |
| Haalman | $\frac{2L}{3T}$ | $T$ |
| Refined ZN | $k_u\frac{5}{6}(\frac{12+\kappa}{15+14\kappa})$ | $T_u\frac{1}{5}(\frac{4}{15}\kappa + 1)$ |

**PID control**

| Method | $k_c k$ | $T_i$ | $T_d$ |
|---|---|---|---|
| ZN step | $1.2(\frac{L}{T})^{-1}$ | $2L$ | $0.5L$ |
| ZN osc. | $0.6k_u$ | $T_u/2$ | $T_u/8$ |
| Cohen-Coon | $1.35(\frac{L}{T})^{-1}+0.25$ | $L\frac{1.35+0.25\frac{L}{T}}{0.54+0.33\frac{L}{T}}$ | $L\frac{0.5}{1.35+0.25\frac{L}{T}}$ |
| ISE load | $1.495(\frac{L}{T})^{-0.945}$ | $0.917T(\frac{L}{T})^{0.771}$ | $0.560T(\frac{L}{T})^{1.006}$ |
| IAE load | $1.435(\frac{L}{T})^{-0.921}$ | $1.139T(\frac{L}{T})^{0.749}$ | $0.482T(\frac{L}{T})^{1.137}$ |
| ITAE load | $1.357(\frac{L}{T})^{-0.947}$ | $1.176T(\frac{L}{T})^{0.738}$ | $0.381T(\frac{L}{T})^{0.995}$ |
| ISE set point | $1.048(\frac{L}{T})^{-0.897}$ | $\frac{T}{1.195-0.368(\frac{L}{T})}$ | $0.489T(\frac{L}{T})^{0.888}$ |
| IAE set point | $1.086(\frac{L}{T})^{-0.869}$ | $\frac{T}{0.74-0.13(\frac{L}{T})}$ | $0.348T(\frac{L}{T})^{0.914}$ |
| ITAE set point | $0.965(\frac{L}{T})^{-0.85}$ | $\frac{T}{0.796-0.1465(\frac{L}{T})}$ | $0.308T(\frac{L}{T})^{0.929}$ |
| Pessen 1 | $k_u/5$ | $T_u/3$ | $T_u/2$ |
| Pessen 2 | $k_u/3$ | $T_u/2$ | $T_u/3$ |

These formulas have been compiled from [Ziegler and Nichols, 1942], [Hazebroek and van der Waerden, 1950b], [Cohen and Coon, 1953], [Haalman, 1965], [Miller *et al.*, 1967], [Rovira *et al.*, 1969], and [Hang *et al.*, 1991]. Most of these formulas are valid only for $0.1 < L/T < 1$. A number of these tuning formulas are plotted in Figure 3.2. As can be seen the maih difference between the different tuning rules are their values for the integration time. Comparisons and interpretations of different tuning rules will be discussed further in Chapter 7.

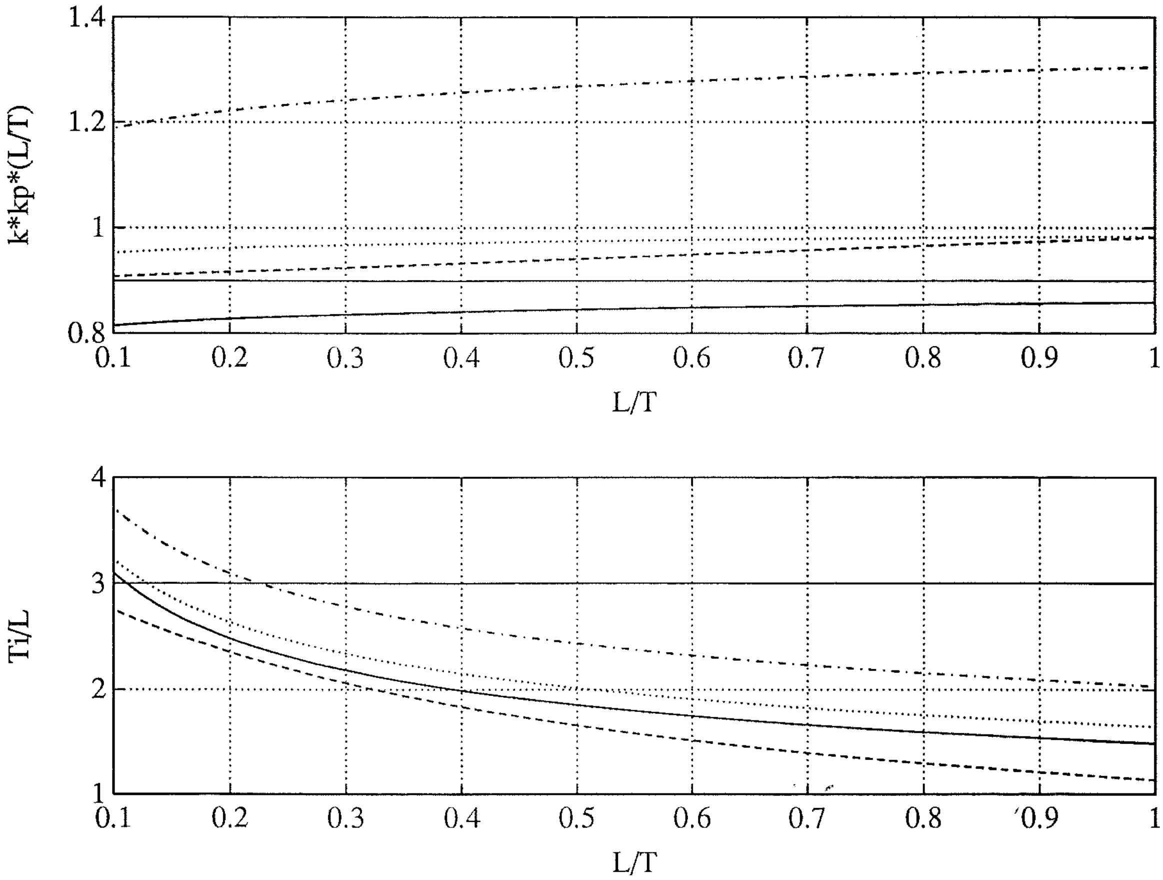



**Figure 3.2** The controller parameters for of $G(s) = e^{-sL}/(1 + sT)$, with ZN (solid straight line), Cohen-Coon (dashed line), IAE (dotted line), ISE (dashed-dotted line), and ITAE (solid line).

# 4

# PI and PD control

The most common control algorithm is the PI controller. When a PID controller is used the derivative action is often switched off, and it becomes a PI controller. In this chapter methods for tuning PI controllers will be developed. Since a PI controller has two parameters it is natural to try to specify the controller by placing two poles of the closed loop system. PD controllers are not as commonly used as PI controllers, but since PD controllers also have two parameters they can be tuned in a similar way.

## 4.1 Limitations of PI control

In this section we will give some examples of the limitations of PI control.

EXAMPLE 4.1—First order system
The closed loop system obtained when a first order system is controlled by a PI controller is of second order. Its poles can be located arbitrarily. Let

$$G(s) = \frac{a}{s+b} \tag{4.1}$$

and

$$G_c(s) = k(1 + \frac{1}{sT_i}) \tag{4.2}$$

then the characteristic equation becomes

$$s^2 + (b + ak)s + \frac{ak}{T_i} = 0. \tag{4.3}$$

From this it is clear that any poles can be obtained by chosing $k$ and $T_i$ properly. Hence for first order systems dynamics is no limitation. Tuning is instead limited by other factors. □

EXAMPLE 4.2—Systems without integrators
Systems without integrators can always be stably controlled by a PI controller. This can easily be realized from a root-locus argument. Of course the system may be very slow. The key point is that there is only one integrator in the origin. □

EXAMPLE 4.3—System with double integrator
A PI controller has a phase lag between 0° and 90°. This means that any process with a phase lag of 180° or larger cannot be controlled by a PI controller. PI control will not work for a double integrator, as can be seen from the following argument. Suppose we have

$$G(s) = \frac{1}{s^2}\frac{1}{a(s)} \tag{4.4}$$

with $\deg a(s) = n$. The characteristic equation for this system controlled by a PI controller, $G_c(s) = k + k_i/s$, is

$$s^3 a(s) + (ks + k_i) = 0. \tag{4.5}$$

Since there is no $s^2$ term we know for sure that the system cannot be stable when controlled by a PI controller. □

## 4.2 Dominant Pole Design

A systematic method for tuning controllers will now be developed, given a system with transfer function $G(s)$. The idea with Dominant Pole Design (DPD) is to find a controller such that the dominant poles of the closed loop system are in specified locations. In general it is possible to place as many poles as there are free parameters in the controller. Problems may arise when there are more poles in the closed loop system than the number of free parameters in the controller. The poles we have specified will of course be in the right positions, but we have little control of the other poles. Whether or not the system will behave sensibly will depend on where we place the "dominant" poles. E.g., if we specify what we think are the dominant poles too far out in the complex plane, i.e., require too high bandwidth of the closed loop system, it is very likely that a system of high order will be unstable in closed loop.

If the system is of such low order that the order of the system plus the order of the controller is less or equal to the number of free parameters in the controller, of course these problems do not arise. Then we may place all the poles in suitable locations. Similar ideas for tuning simple controllers can be found in [Hartree *et al.*, 1937], [Cohen and Coon, 1953] and [Wolfe, 1951]. The methods presented here can be seen as developments and generalizations of their ideas.

The formulation of DPD used in this thesis makes no assumptions of $G(s)$. We do not require knowledge of poles or zeros of $G(s)$, i.e., we do not require that it should be possible to write $G(s)$ as a system of ordinary differential equations. Some problems including flow and heat transfer give rise to transfer functions including $e^{-\sqrt{sT}}$ see, e.g., [Sundaresan and Krishnaswamy, 1978]. Unfortunately such systems are difficult to simulate in the time domain.

### Dominant Poles

Intuitively it seems reasonable to say that some closed loop poles are dominating if the closed loop response is well approximated by the contribution from these poles. Many control systems have a closed loop pole-zero pattern as indicated in Figure 4.1, see [Truxal, 1955]. Sometimes the complex poles with largest real parts are called the dominating poles, see [Takahashi *et al.*, 1972]. In Figure 4.1, A and B are the dominating poles according to this definition. The situation in general is not so simple because the closed loop response also depends critically on the existence of closed loop zeros, close to certain poles, like the pole-zero pair C in Figure 4.1. This pole-zero pair gives rise to a small residual. This pole gives a small contribution to

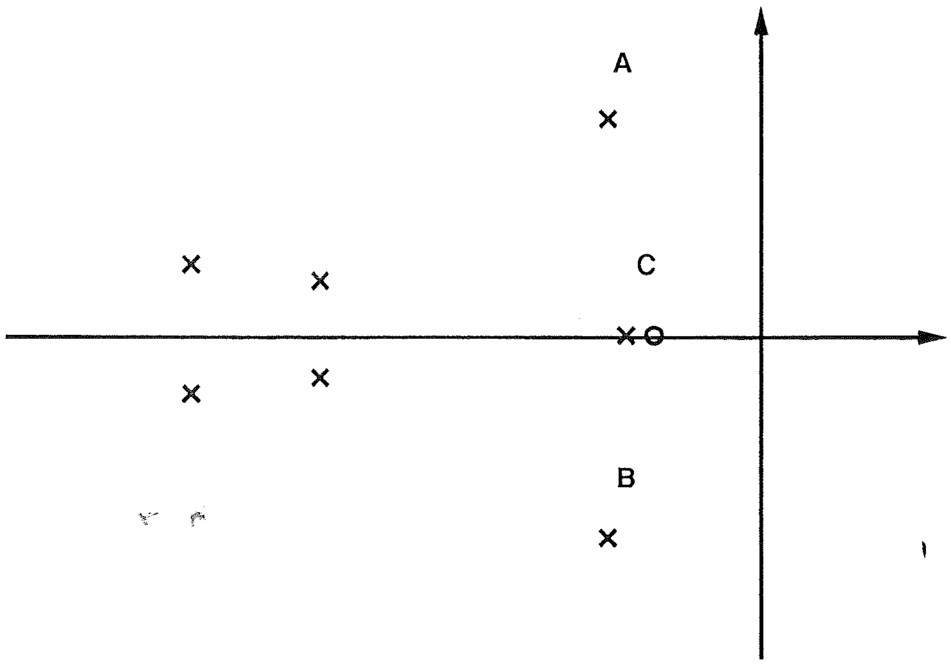



**Figure 4.1** A typical pole-zero pattern of a closed loop system.

a time response. Similarly, oscillating poles with small real parts and large imaginary parts may also be poorly exited by typical commands.

### Dominant Pole Placement

Suppose that a plant with transfer function $G(s)$ is given. We want to control the plant with a controller which transfer function is $G_c(s)$. Suppose that the controller has the structure

$$G_c(s) = \frac{b_0 s^m + b_1 s^{m-1} + \ldots + b_m}{s^\ell (s^n + a_1 s^{n-1} + \ldots + a_n)} = \frac{B(s)}{s^\ell A(s)}. \tag{4.6}$$

Where the constants $m, n$, and $\ell$ are given. There are then $m+n+1$ free constants in the controller. The closed-loop transfer function is

$$G_{\mathrm{CL}}(s) = \frac{B(s)G(s)}{s^\ell A(s) + B(s)G(s)}. \tag{4.7}$$

To determine the $m+n+1$ parameters we require that the closed loop system must have poles in specified locations $p_1, \ldots, p_{m+n+1}$. We must solve

$$p_i^\ell (p_i^n + a_1 p_i^{n-1} + \ldots + a_n) + G(p_i)(b_0 p_i^m + b_1 p_i^{m-1} + \ldots + b_m) = 0, \tag{4.8}$$

for $i = 1, \ldots, m+n+1$. Let

$$R = \begin{bmatrix} p_1^{\ell+n-1} & \ldots & p_1^\ell & G(p_1)p_1^m & \ldots & G(p_1) \\ \vdots & & \vdots & \vdots & & \vdots \\ p_{m+n+1}^{\ell+n-1} & \ldots & p_{m+n+1}^\ell & G(p_{m+n+1})p_{m+n+1}^m & \ldots & G(p_{m+n+1}) \end{bmatrix}. \tag{4.9}$$

Introduce

$$v = \begin{pmatrix} a_1 & \ldots & a_n & b_0 & \ldots & b_m \end{pmatrix}^T \tag{4.10}$$

$$P = \begin{pmatrix} -p_1^{\ell+n} & \ldots & -p_{m+n+1}^{\ell+n} \end{pmatrix}^T. \tag{4.11}$$

The controller parameters are now obtained as solutions of the equation

$$Rv = P. \tag{4.12}$$

This requires $R$ to be invertible. If $p_i = p_j$, $i \neq j$ then two rows in the $R$ matrix will be equal and thus $R$ will not be invertible. Problems may also arise when we try to place a closed loop pole on the location of an open loop pole, where $G(p_i) = \infty$, and the equations are not solvable. As special cases of this method we may get different forms of PID controllers.

***PI controllers*** are obtained if $m = 1$, $n = 0$ and $\ell = 1$. If we use the standard form for a PI-controller

$$G_{\mathrm{PI}}(s) = k(1 + \frac{1}{sT_i}), \tag{4.13}$$

we get

$$k = b_0 \tag{4.14}$$
$$T_i = b_0/b_1. \tag{4.15}$$

***PID controllers*** are obtained if $m = 2$, $n = 0$ and $\ell = 1$. With the standard parametrization

$$G_{\mathrm{PID}}(s) = k(1 + \frac{1}{sT_i} + sT_d), \tag{4.16}$$

we get

$$k = b_1 \tag{4.17}$$
$$T_i = b_1/b_2 \tag{4.18}$$
$$T_d = b_0/b_1. \tag{4.19}$$

***PID controllers with derivative filter*** are obtained if $m = 2$, $n = 1$, and $\ell = 1$. If we use the standard form for a PID-controller with a derivative filter

$$G_{\mathrm{PIDF}}(s) = k(1 + \frac{1}{sT_i} + \frac{sT_d}{1 + s\frac{T_d}{N}}), \tag{4.20}$$

we get

$$T_i = \frac{b_1}{b_2} - \frac{1}{a_1} \tag{4.21}$$
$$k = \frac{b_2 T_i}{a_1} \tag{4.22}$$
$$N = \frac{b_0}{k} - 1 \tag{4.23}$$
$$T_d = \frac{N}{a_1 - 1}. \tag{4.24}$$

An advantage with this method is that we do not need to have a state-space representation of $G(s)$. The choice of the poles $p_i$ is of course critical. It may be very difficult to choose many closed loop poles for a high order system, and get a sensible controller. The choice of only a few poles may be handled successfully. To place two poles, as in the case of the PI controller, is almost always possible. To specify a PID controller by placing three poles may be problematic

For these reasons it must be stressed that pole placement is not the way to choose the parameter $N$ in a PID controller. An attempt to do so will in most cases give very strange values of $N$. The parameter $N$ is a parameter to reduce the noise sensitivity of the controller and more or less an implementation issue.

## 4.3 Two Dominant Poles

In case of a PI or PD controller there are two free parameters, thus two poles must be placed. In this section PI and PD controllers will be derived from this principle.

### PI Control

It is convenient to parametrize the PI controller as

$$G_c(s) = k + \frac{k_i}{s} = k(1 + \frac{1}{sT_i}). \tag{4.25}$$

The characteristic equation of a system with transfer function $G(s)$ controlled by a PI controller is then

$$1 + (k + \frac{k_i}{s})G(s) = 0. \tag{4.26}$$

We require the closed loop system to have poles in

$$p_{1,2} = \omega_0(-\zeta_0 \pm i\sqrt{1-\zeta_0^2}) = -\sigma \pm i\omega, \quad 0 < \zeta_0 < 1. \tag{4.27}$$

It is the task of the designer of the controller to choose $\zeta_0$ and $\omega_0$. Thus we must solve the equation

$$1 + (k + \frac{k_i}{p_1})G(p_1) = 0 \tag{4.28}$$

for $k$ and $k_i$. If we let

$$A = \operatorname{Re} G(p_1) \tag{4.29}$$
$$B = \operatorname{Im} G(p_1) \tag{4.30}$$

we get the equation

$$1+(k+\frac{k_i}{-\sigma+i\omega})G(-\sigma+i\omega)=$$
$$=1+(k+\frac{k_i}{-\sigma+i\omega})(A+iB)=0. \tag{4.31}$$

By setting real and imaginary parts to zero we can solve (4.31) for $k$ and $k_i$, and we get

$$k=-\frac{\omega A+\sigma B}{\omega(A^2+B^2)}=-\frac{\sqrt{1-\zeta_0^2}A+\zeta_0 B}{\sqrt{1-\zeta_0^2}(A^2+B^2)} \tag{4.32}$$
$$k_i=-\frac{(\omega^2+\sigma^2)B}{\omega(A^2+B^2)}=-\frac{\omega_0 B}{\sqrt{1-\zeta_0^2}(A^2+B^2)}. \tag{4.33}$$

If the controller is parametrized as

$$G_c(s)=k(1+\frac{1}{sT_i}), \tag{4.34}$$

the integration time, $T_i$, is given by

$$T_i=\frac{k}{k_i}=\frac{\sqrt{1-\zeta_0^2}A+\zeta_0 B}{\omega_0 B}. \tag{4.35}$$

If we want to have $\zeta_0=1$, i.e., a double pole in $-\omega_0$, both the characteristic equation and its derivative must have a zero in $s=-\omega_0$. We get

$$1+G_c(-\omega_0)G(-\omega_0)=0 \tag{4.36}$$
$$G_c'(-\omega_0)G(-\omega_0)+G_c(-\omega_0)G'(-\omega_0)=0. \tag{4.37}$$

Let $A=G(-\omega_0)$ and $A'=G'(-\omega_0)$, then we get the equations

$$1+(k-\frac{k_i}{\omega_0})A=0 \tag{4.38}$$
$$-\frac{k_i}{\omega_0^2}A+(k-\frac{k_i}{\omega_0})A'=0, \tag{4.39}$$

which have the solution

$$k=\frac{A+\omega_0 A'}{A^2} \tag{4.40}$$
$$k_i=-\frac{\omega_0^2 A'}{A^2}. \tag{4.41}$$

These equations require us to know both $G$ and $G'$, but $G'$ may be difficult to estimate accurately from measurement data.

We may also get a double pole by letting $\zeta_0 \to 1$. In this case $G(\omega_0(-\zeta_0 + i\sqrt{1-\zeta_0^2})) \to A$, and $B \to 0$ and we get

$$k = -\frac{1}{A} - \frac{1}{A^2} \lim_{\zeta_0 \to 1} \frac{B}{\sqrt{1-\zeta_0^2}} \tag{4.42}$$

$$k_i = -\frac{\omega_0}{A^2} \lim_{\zeta_0 \to 1} \frac{B}{\sqrt{1-\zeta_0^2}}. \tag{4.43}$$

In numerical computations this is the easiest way to get a double pole.

***A different parametrization*** To get further insight in the nature of $k$ and $k_i$ we parametrize the equations in another way. This parametrization will be used later in the analysis of PI design.

Define $a(\omega_0)$ and $\phi(\omega_0)$ by $G(\omega_0 e^{i(\pi-\gamma)}) = a(\omega_0)e^{i\phi(\omega_0)}$, where $\gamma$ is regarded as a constant. However, it must be remembered that $a(\omega_0)$ and $\phi(\omega_0)$ depend on both $\omega_0$ and $\gamma$. For the poles $p_1$ we get

$$\omega_0(-\zeta_0 + i\sqrt{1-\zeta_0^2}) = \omega_0 e^{i(\pi-\gamma)} = \omega_0(-\cos\gamma + i\sin\gamma). \tag{4.44}$$

Hence $\gamma = \arccos\zeta_0$. Using previously introduced notations we get

$$A = a(\omega_0)\cos\phi(\omega_0) \tag{4.45}$$

$$B = a(\omega_0)\sin\phi(\omega_0). \tag{4.46}$$

Insert these expressions for $A$ and $B$ in (4.33) and we get

$$k = -\frac{\sin(\phi(\omega_0)+\gamma)}{a(\omega_0)\sin\gamma} \tag{4.47}$$

$$k_i = -\frac{\omega_0 \sin\phi(\omega_0)}{a(\omega_0)\sin\gamma} \tag{4.48}$$

and

$$T_i = \frac{k}{k_i} = \frac{\sin(\phi(\omega_0)+\gamma)}{\omega_0 \sin\phi(\omega_0)}. \tag{4.49}$$

Notice that $T_i$ is independent of $a(\omega_0)$. From these formulas we see that the phase $\phi(\omega_0)$ is important if we want to maximize some of the controller parameters.

When $\gamma = \pi/2$ then $G(\omega_0 e^{i(\pi-\gamma)}) = G(i\omega)$ which is the normal frequency response. If $\gamma < \pi/2$ then the 'Nyquist curve' of $G(\omega_0 e^{i(\pi-\gamma)})$ will look different from the normal one. In many cases the difference will be very significant.

EXAMPLE 4.4
Suppose

$$G(s) = \frac{e^{-sL}}{1+sT}, \tag{4.50}$$

from the formulas for $k$ and $k_i$, (4.32) and (4.33), we get

$$k = e^{-\omega_0 L\zeta_0}((2\omega_0 T\zeta_0 - 1)\cos\omega_0 L\sqrt{1-\zeta_0^2} + \\ + (\omega_0 T\sqrt{1-\zeta_0^2} + \frac{\zeta_0}{\sqrt{1-\zeta_0^2}} - \omega_0 T\frac{\zeta_0^2}{\sqrt{1-\zeta_0^2}})\sin\omega_0 L\sqrt{1-\zeta_0^2} \tag{4.51}$$

$$k_i = \omega_0 e^{-\omega_0 L\zeta_0}\cdot \\ \cdot(\omega_0 T\cos\omega_0 L\sqrt{1-\zeta_0^2} + \frac{1-\omega_0 T\zeta_0}{\sqrt{1-\zeta_0^2}}\sin\omega_0 L\sqrt{1-\zeta_0^2}). \tag{4.52}$$

□

EXAMPLE 4.5—Poles with a PI controller
Suppose we have $G(s) = 1/(s+1)^4$, controlled by a PI controller. It is the task of the designer of the controller to choose suitable $k$ and $T_i$, or in our formulation $\omega_0$ and $\zeta_0$. In Figure 4.2 the root locus of the equation $1 + G_{\mathrm{PI}}(s)G(s) = 0$ is shown for $\zeta_0 = 0.7$ and $\omega_0 = 0.1\ldots 2.0$.

For small values of $\omega_0$ the two complex poles closest to the origin dominate the system, and as the complex poles are moved further from the origin other poles move into the right half plane and causes instability. This is a very typical situation when designing controllers with this method. In the following we will suggest methods for choosing $\omega_0$ and $\zeta_0$ such that the system will behave well. □

## PD Control

With the same notations as in the PI case we get the parameters of the PD controller $G_{\mathrm{PD}}(s) = k + k_d s$ when we place two zeros of $1 + G(s)G_{\mathrm{PD}}(s) = 0$ in $p_{1,2} = \omega_0(-\zeta_0 \pm i\sqrt{1-\zeta_0^2})$. Introducing $A = \mathrm{Re}\, G(p_1)$ and $B = \mathrm{Im}\, G(p_1)$ we get

$$k = \frac{-\sqrt{1-\zeta_0^2}A + \zeta_0 B}{\sqrt{1-\zeta_0^2}(A^2+B^2)} \tag{4.53}$$

$$k_d = \frac{B}{\omega_0\sqrt{1-\zeta_0^2}(A^2+B^2)}. \tag{4.54}$$

Define the functions $a(\omega_0)$ and $\phi(\omega_0)$ like in the case of PI control, then

$$A = a(\omega_0)\cos\phi(\omega_0) \tag{4.55}$$

$$B = a(\omega_0)\sin\phi(\omega_0), \tag{4.56}$$

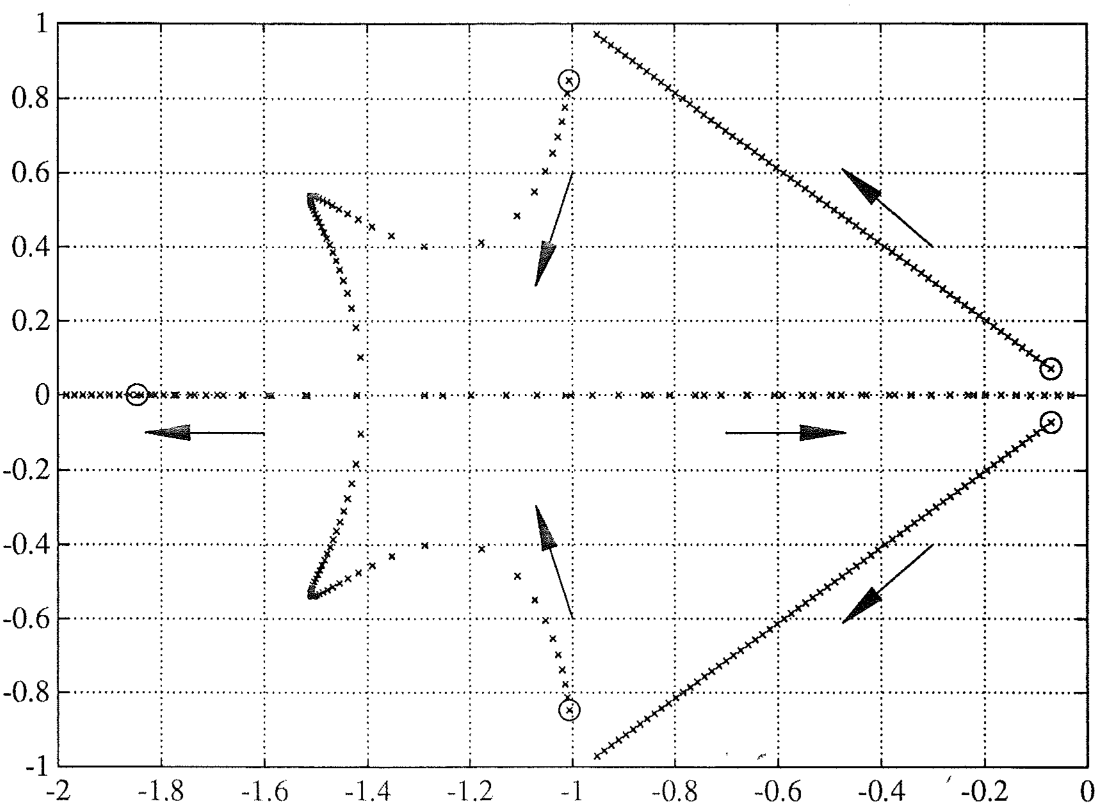



**Figure 4.2** Root locus of $1 + G_{\rm PI}(s)G(s) = 0$ with respect to $\omega_0$ for $\zeta_0 = 0.7$. Roots for $\omega_0 = 0.1$ are marked with 'o'.

and the formulas for $k$ and $k_d$ in this parametrization become,

$$k = \frac{\sin(\phi(\omega_0) - \gamma)}{a(\omega_0)\sin\gamma} \tag{4.57}$$

$$k_d = \frac{\sin\phi(\omega_0)}{\omega_0 a(\omega_0)\sin\gamma}. \tag{4.58}$$

Note that the expressions of $k$ and $k_d$ for PD controllers are similar to those of PI controllers.

## Maximizing $k_i$

In Chapter 2 it was shown that $\text{IE} = 1/k_i$. Since IE can be used as a criterion for tuning PI controllers it is interesting to study the properties of $k_i$ with respect to $\omega_0$ and $\zeta_0$. From the formula (4.48)

$$k_i = -\frac{\omega_0 \sin\phi(\omega_0)}{a(\omega_0)\sin\gamma}, \tag{4.59}$$

it is clear that $k_i$ has a maximum if $G(0) = 1$ and the phase, $\phi$, of the system reaches $-\pi$, because $\sin\phi = 0$ for $\phi = 0$ and $\phi = \pi$, and $\omega_0/a(\omega_0) > 0$, $\forall\omega_0$. Therefore it is necessary to study the phase function $\phi(\omega_0)$.

### Monotonicity of the functions $a(\omega)$ and $\phi(\omega)$

Some insight about the controller can be obtained from the functions $a(\omega)$ and $\phi(\omega)$ which describe the amplitude and phase of the transfer function along the ray $\omega_0 e^{i(\pi-\gamma)}$. These functions are defined by

$$G(s) = G(\omega_0 e^{i(\pi-\gamma)}) = G(\omega_0(-\zeta_0 + i\sqrt{1-\zeta_0^2})) = a(\omega_0)e^{i\phi(\omega_0)}. \qquad (4.60)$$

We say that $G(s)$ has monotonic amplitude and phase on the ray $\omega_0 e^{i(\pi-\gamma)}$ if the functions $a(\omega_0)$ and $\phi(\omega_0)$ are monotonic. In the following we assume that $0 < \gamma \le \pi/2$. For $\gamma = \pi/2$ we have $G(\omega_0 e^{i(\pi-\gamma)}) = G(i\omega_0)$, which is the frequency response of the system. Systems with monotonic phase and amplitude for $\gamma = \pi/2$ need not have monotonic phase and amplitude for $\gamma < \pi/2$. It is obvious that if $G_1(s)$ and $G_2(s)$ have monotonicly decreasing phase or amplitude on a ray the composite system $G_1(s)G_2(s) = a_1(\omega_0)a_2(\omega_0)e^{i(\phi_1(\omega_0)+\phi_2(\omega_0))}$ also has monotonicly decreasing phase or amplitude on the ray. Because of the equations (4.48) and (4.57) it is of particular interest to know if the functions are monotonic at least for some frequency interval.

EXAMPLE 4.6—$a(\omega)$ and $\phi(\omega)$ for some simple functions

- $G(s) = 1/(s+b)$, $b > 0$. In this case we immediately get

$$a(\omega_0) = \frac{1}{\sqrt{(b-\omega_0\cos\gamma)^2 + \omega_0^2\sin^2\gamma}} \qquad (4.61)$$

and

$$\phi(\omega_0) = \begin{cases} -\arctan\frac{\omega_0\sin\gamma}{b-\omega_0\cos\gamma}, & \text{if } \omega_0\cos\gamma \le b; \\ -\pi - \arctan\frac{\omega_0\sin\gamma}{b-\omega_0\cos\gamma}, & \text{if } \omega_0\cos\gamma > b. \end{cases} \qquad (4.62)$$

From this we see that $a(\omega_0)$ has a maximum for $\omega_0 = b\cos\gamma$ and $\phi(\omega_0)$ is negative and monotonicly decreasing. The phase $\phi \to -\pi + \gamma$ as $\omega_0 \to \infty$.

- $G(s) = s + b$, $b > 0$. In this case we immediately get

$$a(\omega_0) = \sqrt{(b-\omega_0\cos\gamma)^2 + \omega_0^2\sin^2\gamma} \qquad (4.63)$$

and

$$\phi(\omega_0) = \begin{cases} \arctan\frac{\omega_0\sin\gamma}{b-\omega_0\cos\gamma}, & \text{if } \omega_0\cos\gamma \le b; \\ \pi + \arctan\frac{\omega_0\sin\gamma}{b-\omega_0\cos\gamma}, & \text{if } \omega_0\cos\gamma > b. \end{cases} \qquad (4.64)$$

The phase $\phi$ is positive, monotonic, and $\phi \to \pi - \gamma$ as $\omega_0 \to \infty$.

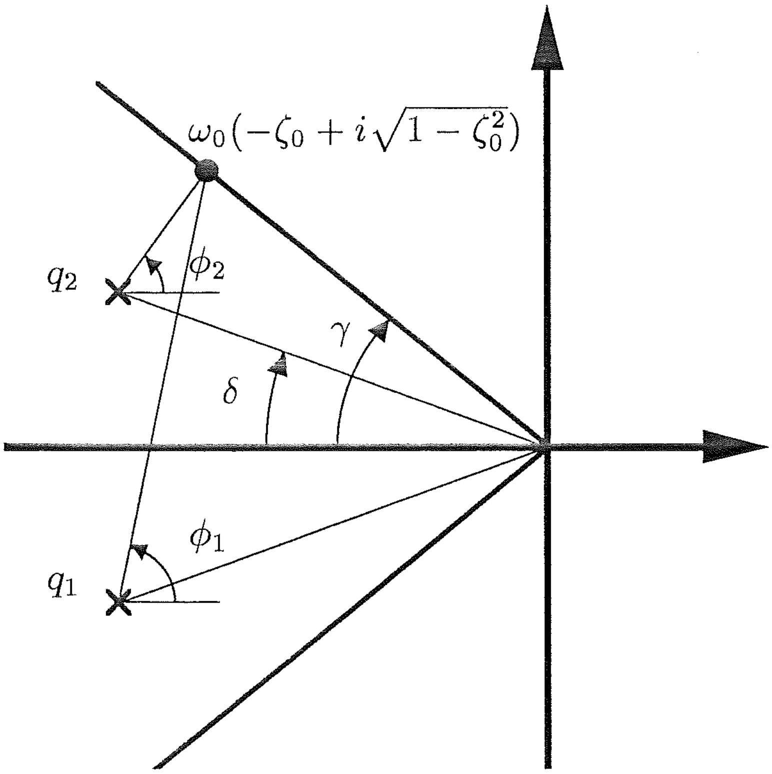



**Figure 4.3** Notations for Example 4.7.

- $G(s) = e^{-sL}$. From this we get $G(\omega_0 e^{i(\pi-\gamma)}) = e^{\omega_0 L \cos\gamma} e^{-iL\omega_0 \sin\gamma}$. Thus $a(\omega_0) = e^{\omega_0 L\cos\gamma}$ and $\phi(\omega_0) = -L\omega_0 \sin\gamma$. We notice that $a(\omega_0)$ is an exponentially increasing function, while $\phi(\omega_0)$ is linear in $\omega_0$.

- $G(s) = 1/s$. $G(\omega_0 e^{i(\pi-\gamma)}) = e^{-i(\pi-\gamma)}/\omega_0$, thus $a(\omega_0) = 1/\omega_0$ and $\phi(\omega_0) = -\pi + \gamma$

- $G(s) = \omega_p^2/(s^2 + 2\zeta_p\omega_p s + \omega_p^2)$, $0 < \zeta_p < 1$. The poles of this transfer function are located at $q_{1,2} = \omega_p e^{i(\pi\pm\delta)}$ where $\cos\delta = \zeta_p$. Let $\phi_{1,2}$ be defined as in Figure 4.3, then $\arg G(\omega_0 e^{i(\pi-\gamma)}) = -(\phi_1 + \phi_2)$.
  Three cases may be identified:
  1. If $\delta < \gamma$ then $-(\phi_1+\phi_2)$ is a monotonicly decreasing function starting in 0 and ending in $-2(\pi-\gamma)$ for $\omega_0 = \infty$.
  2. If $\delta = \gamma$ then $-(\phi_1+\phi_2)$ is a monotonicly decreasing function starting in 0 and ending in $-2(\pi-\gamma)$ for $\omega_0 = \infty$, with a discontinuity of $-\pi$ in $\omega_0 = \omega_p$.
  3. If $\delta > \gamma$, the phase starts in 0, has a local negative minimum in the interval $[0, \omega_p\cos\delta/\cos\gamma]$, goes through 0 for $\omega_0 = \omega_p\cos\delta/\cos\gamma$ and finally ends in $2\gamma$ for $\omega_0 = \infty$. This is realized with some simple geometry. The phase in this case is *not* monotonic.

Thus the sine of the phase in this case always has the property of being zero at two points, if $\gamma < \pi/2$. □

From these investigations we see that if a system has all its poles inside the sector $re^{i\alpha}$, where $0 < r < \infty$ and $\pi - \gamma < \alpha < \pi + \gamma$ then the phase of $G(-\omega_0 e^{-i\gamma})$ is decreasing monotonicly. It is very difficult to make any statement about the monotonicity of the amplitude. Fortunately, for a given transfer function it is very easy to compute the functions $a(\omega)$ and $\phi(\omega)$ and plot them.

EXAMPLE 4.7
Let $G(s) = 1/(1+s)^n$. In this case we can determine $a(\omega_0)$ and $\phi(\omega_0)$,

$$G(-\omega_0 e^{-i\gamma}) = \frac{1}{(1-\omega_0 e^{-i\gamma})^n} = \\ = \frac{1}{((1-\omega_0\cos\gamma)^2 + \omega_0^2\sin^2\gamma)^{\frac{n}{2}}} e^{-in\psi}, \tag{4.65}$$

where

$$\psi(\omega_0) = \begin{cases} \arctan\frac{\omega_0\sin\gamma}{1-\omega_0\cos\gamma}, & \text{if } \omega_0\cos\gamma \le 1; \\ \pi + \arctan\frac{\omega_0\sin\gamma}{1-\omega_0\cos\gamma}, & \text{if } \omega_0\cos\gamma > 1. \end{cases} \tag{4.66}$$

To examine $a(\omega_0)$ and $\phi(\omega_0)$, differentiate the functions, and we get

$$\frac{da(\omega_0)}{d\omega_0} = \frac{n(-\omega_0 + \cos\gamma)}{((1-\omega_0\cos\gamma)^2 + \omega_0^2\sin^2\gamma)^{\frac{n}{2}+1}} \tag{4.67}$$

and

$$\frac{d\phi(\omega_0)}{d\omega_0} = -\frac{n\sin\gamma}{1-2\omega_0\cos\gamma+\omega_0^2}. \tag{4.68}$$

From this example we see that the amplitude for $G(-\omega_0 e^{-i\gamma})$ has a maximum for $\omega_0 = \cos\gamma$, and that the phase decreases monotonicly, since $1-2\omega_0\cos\gamma + \omega_0^2 > 0, \forall\,\omega_0$. The 'normal' Nyquist curve has a monotonicly decreasing amplitude and phase for all $\omega_0 > 0$. □

The following example will show the behaviour of $G(s)$ on a ray when the system has a time delay.

EXAMPLE 4.8
Let $G(s) = e^{-sL}/(1+sT)$. From the definition $G(s) = G(\omega_0 e^{-i(\pi-\gamma)}) = a(\omega_0)e^{i\phi(\omega_0)}$ we get

$$a(\omega_0) = \frac{e^{\omega_0\zeta_0 L}}{\sqrt{1-2\omega_0\cos\gamma T + \omega_0^2 T^2}} \tag{4.69}$$

$$\frac{da(\omega_0)}{d\omega_0} = e^{\omega_0\zeta_0 L}\frac{\omega_0^2\zeta_0 LT^2 - \omega_0 T(2\zeta_0^2 L + T) + \zeta_0(L+T)}{(1-2\omega_0\zeta_0 T + \omega_0^2 T^2)^{\frac{3}{2}}}. \tag{4.70}$$

and

$$\phi(\omega_0) = -\omega_0 L\sin\gamma - \psi(\omega_0 T) \tag{4.71}$$

where $\psi(\omega_0)$ is defined in Example 4.7. Clearly $\phi(\omega_0 T)$ is monotonicly decreasing $\forall\, \omega_0, L, T > 0$, and $\phi(\omega_0) \to -\infty$ as $\omega_0 \to \infty$.

On the other hand, the behaviour of $a(\omega_0)$ is more complex. As $\omega_0 \to \infty$, $a(\omega_0) \to \infty$, $\forall L, T > 0$. From the expression for $da(\omega_0)/d\omega_0$ we see that if $T/L < 2\sin\gamma$ then $a(\omega_0)$ is monotonicly increasing, otherwise $a(\omega_0)$ first increases, reaches a maximum, decreases, reaches a minimum, and finally goes to infinity as $\omega_0 \to \infty$. □

## Properties of $k$ and $k_i$

One possible criterion for design of a PI controller is to minimize the integrated error (IE) for a load disturbance at the process input. It was shown in Chapter 2 that this is equivalent to maximizing $k_i$. Hence there is an interest in more understanding of the functions $k = k(\omega_0, \zeta_0)$ and $k_i = k_i(\omega_0, \zeta_0)$.

Rolle's theorem says that if for a continuous function $f$, $f(a) = 0$ and $f(b) = 0$, $\exists \xi : a < \xi < b$ with $f'(\xi) = 0$. From the formula

$$k_i = -\frac{\omega_0 \sin\phi(\omega_0)}{a(\omega_0)\sin\gamma} \tag{4.72}$$

it is clear that $k_i$ has a maximum if $0 < G(0) < \infty$ and the phase, $\phi$ of the system reaches $-\pi$, because $\sin\phi = 0$ for $\phi = 0$ and for $\phi = \pi$, and $\omega_0/a(\omega_0) > 0$, $\forall \omega_0$.

Write the formula for $k_i$ as $k_i(\omega_0) = g(\omega_0)\sin\phi(\omega_0)$ where we have introduced $g(\omega_0) = \omega_0/(a(\omega_0)\sin\gamma)$. Differentiating then gives $k_i'(\omega_0) = g'(\omega_0)\sin\phi(\omega_0) + g(\omega_0)\phi'(\omega_0)\cos\phi(\omega_0)$. Setting $k_i' = 0$ we get $\tan\phi(\omega_0) = -g(\omega_0)\phi'(\omega_0)/g'(\omega_0)$. A reasonable assumption on the plant is that $\phi'(\omega_0) < 0$ and $g'(\omega_0) > 0$, which corresponds to monotonicly decreasing phase and amplitude. From this we see that there is a solution $\omega_x$ such that $-\pi < \phi(\omega_x) < -\pi/2$. The interpretation is that the point on the curve $G(\omega_0 e^{i(\pi-\gamma)})$ corresponding to maximal $k_i$ lies in the third quadrant.

***A condition on*** $G(s)$ If $G(s)$ is a rational, stable function with relative degree $n$, the phase of $G(s) = G(\omega_0 e^{i(\pi-\gamma)})$, approaches $-n(\pi-\gamma)$ as $\omega_0$ approaches infinity if all the poles and zeros of the system lies in the sector $re^{i\alpha}$ $0 < r < \infty$ and $\pi - \gamma < \alpha < \pi + \gamma$. In that case, if $-n(\pi-\gamma) < -\pi$, i.e., $n > \pi/(\pi-\gamma)$, we are certain that $k_i$ has a maximum. Since $0 < \gamma < \pi/2$, we see that a relative degree of 2 of the plant and the condition on the damping of the plant poles guarantees that $k_i$ has a maximum.

***Composite systems*** If $G_{12}(s) = G_1(s)G_2(s)$ and we parametrize the transfer functions according to (4.48), then we get

$$k_{i12} = -\frac{\omega_0 \sin(\phi_1 + \phi_2)}{a_1 a_2 \sin\gamma}. \tag{4.73}$$

From this formula it is seen that a composite system has a maximum of $k_i$ if the sum of the phases of the components adds to $-\pi$ for some $\omega_0$.

For a second order system we can compute the maximum value of $\omega_0$ which give stable closed system. Suppose we have a transfer function $G(s) = G_2(s)G_a(s)$, where $G_2$ is a second order system and $G_a$ denotes some additional dynamics then $k_i = -\omega_0 \sin(\phi_2 + \phi_a)/a_2 a_a \sin\gamma$. The stability limit is then the $\omega_0$ for which $\phi_2(\omega_0) + \phi_a(\omega_0) = -\pi$. The stability limit is $\omega_p\zeta_p/\zeta_0$ for a pure second-order system. From the previous formula we see that the stability limit is decreased when additional dynamics is introduced if $\phi_a(\omega_0)$ is monotonicly decreasing.

***Consequences of*** $T_i > 0$ If the transfer function of the process is $G(s)$ and the transfer function of the controller is $G_c(s)$, the closed loop system will be

$$G_{\mathrm{CL}} = \frac{G(s)G_c(s)}{1 + G(s)G_c(s)}. \tag{4.74}$$

Hence the zeros of $G_{\mathrm{CL}}$ are the zeros of the process and the zeros of the controller. If $G_c(s)$ is a PI regulator we are introducing a zero at $-1/T_i$. From the formulas for $k$ and $k_i$ we get

$$T_i = \frac{k}{k_i} = -\frac{\sin(\phi(\omega_0) + \gamma)}{\omega_0 \sin\phi(\omega_0)}. \tag{4.75}$$

A common requirement in PID controller tuning is that $T_i$ should be positive. In that case we must have

$$\frac{\sin(\phi(\omega_0) + \gamma)}{\omega_0 \sin\phi(\omega_0)} > 0. \tag{4.76}$$

If we assume that $\phi(\omega_0) < 0$ then we get

$$-\pi < \phi(\omega_0) < -\gamma. \tag{4.77}$$

For most transfer functions $\phi(\omega_0)$ is monotonic and negative. The assumption that $T_i$ should be positive and the inequality (4.77) puts limits on the possible values of $\omega_0$.

***The*** $k_i(k)$ ***curves*** The stability regions of a system controlled by a PI controller is $k_i > 0$ (provided $G(0) > 0$) and the $k_i(k)$ curve obtained with $\zeta_0 = 0$, which corresponds to placing two poles on the imaginary axis. We get

$$k = -\frac{\mathrm{Re}\, G(i\omega_0)}{|G(i\omega_0)|^2} \tag{4.78}$$

$$k_i = -\frac{\omega_0 \mathrm{Im}\, G(i\omega_0)}{|G(i\omega_0)|^2}. \tag{4.79}$$

Motivations and discussions of this can be found in [Hwang and Chang, 1987].

EXAMPLE 4.9—$k_i(k)$ curves for $G(s) = e^{sL}/(1 + sT)$
Consider

$$G(s) = \frac{e^{-sL}}{1 + sT}. \tag{4.80}$$

In Figure 4.4 the $k_i(k)$ curves are plotted as functions of $\omega_0$ and $\zeta_0$. The solid lines correspond to constant $\zeta_0 = 0, 0.1, \ldots 1.0$ and the dotted ones to constant $\omega_0$.

In the figure some of the standard PI controller settings are marked. Settings according to Ziegler-Nichols are marked with 1, Cohen-Coon with 2, IAE with 3, ISE with 4, and ITAE with 5. Note that all these settings occur for an $\omega_0$ which is greater than the $\omega_0$ corresponding to a maximal $k_i$ using the $\zeta_0$ which is obtained at the different optimal settings. This fact will later be used for design procedure recommendations. □

EXAMPLE 4.10—$k_i(k)$ curves for $G(s) = e^{-sL}/s(1 + sT)$
Consider

$$G(s) = \frac{e^{-sL}}{s(1 + sT)}. \tag{4.81}$$

The curves in Figure 4.5 resemble the curves from Example 4.9. □

EXAMPLE 4.11—$k_i(k)$ curves for a resonant system
Consider

$$G(s) = \frac{0.5}{(s + 0.5)(s^2 + 2 \cdot 0.2s + 1)}. \tag{4.82}$$

Its $k_i(k)$ plot can be seen in Figure 4.6.

When curves for higher $\zeta_0$ crosses curves for lower $\zeta_0$ we know for sure that there are less damped poles than the poles corresponding to controllers on the curve with high value of $\zeta_0$.

In this example we see that poorly damped poles in the plant implies that $k_i$ may not have a maximum if we require too well damped poles in the closed loop system. This also puts a limit on the bandwidth of the closed loop system if we want well damped poles. □

Systems may also have non-convex $k_i(k)$ curves. An example is

$$G(s) = \frac{e^{-s}}{1 + s} + \frac{e^{-2s}}{1 + 2s} \tag{4.83}$$

for large values of $\zeta_0$. Systems with several transport delays may be described with such models.

Using this parametrization of the controller it is easily seen from the $k_i(k)$ diagrams what can be done with a PI controller. The diagrams indicate upper bounds on the bandwidth of the systems, and which settings that correspond to unstable or poorly damped systems. The rest of our investigations will be based on such diagrams.

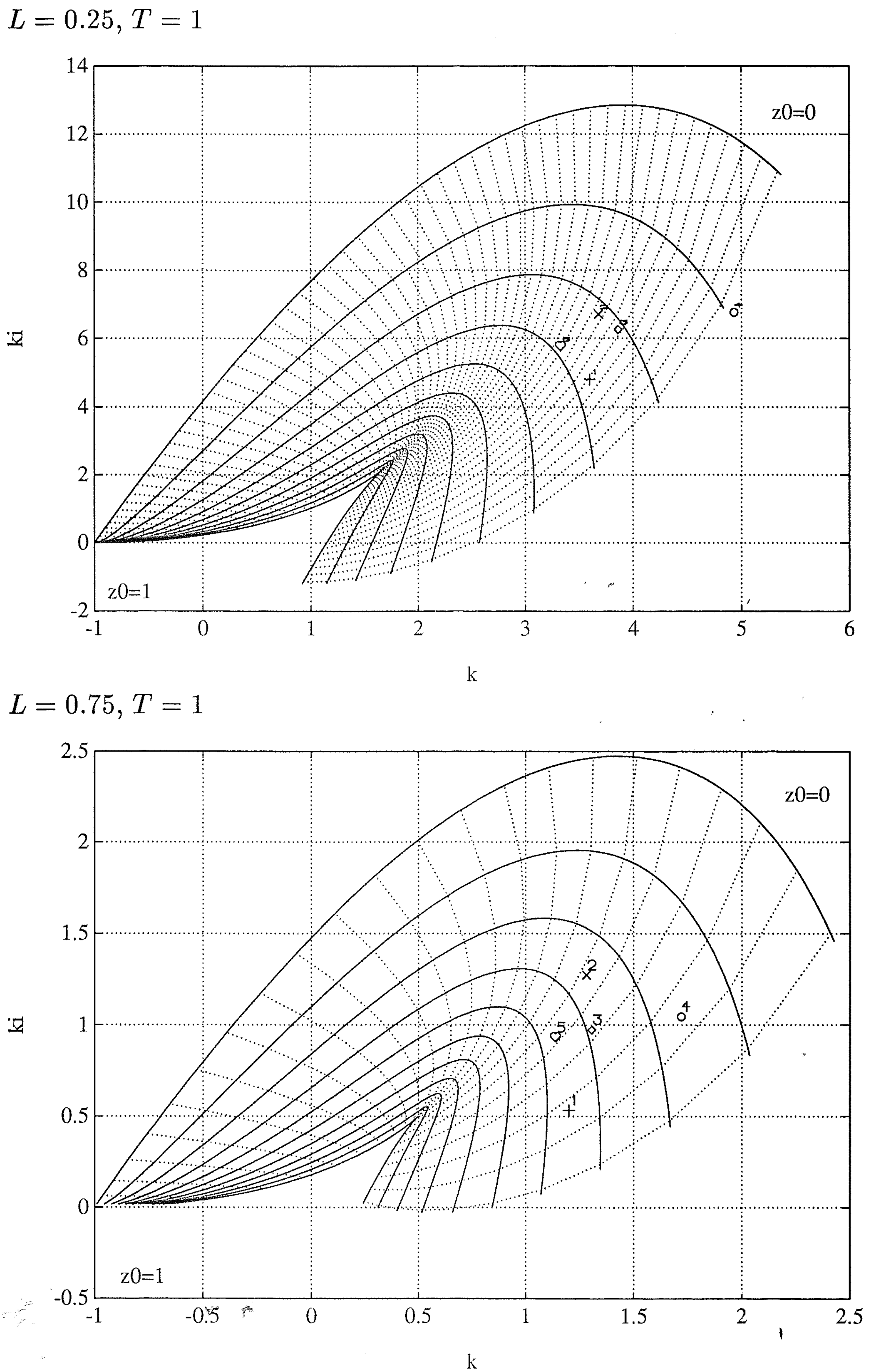



**Figure 4.4** The $k_i(k)$ curves for $G(s) = e^{-sL}/(1+sT)$ for $L = 0.25$, $T = 1$ and $L = 0.75$, $T = 1$. Settings from different tuning rules are marked. Ziegler-Nichols: 1, Cohen-Coon: 2, IAE: 3, ISE: 4, and ITAE: 5.

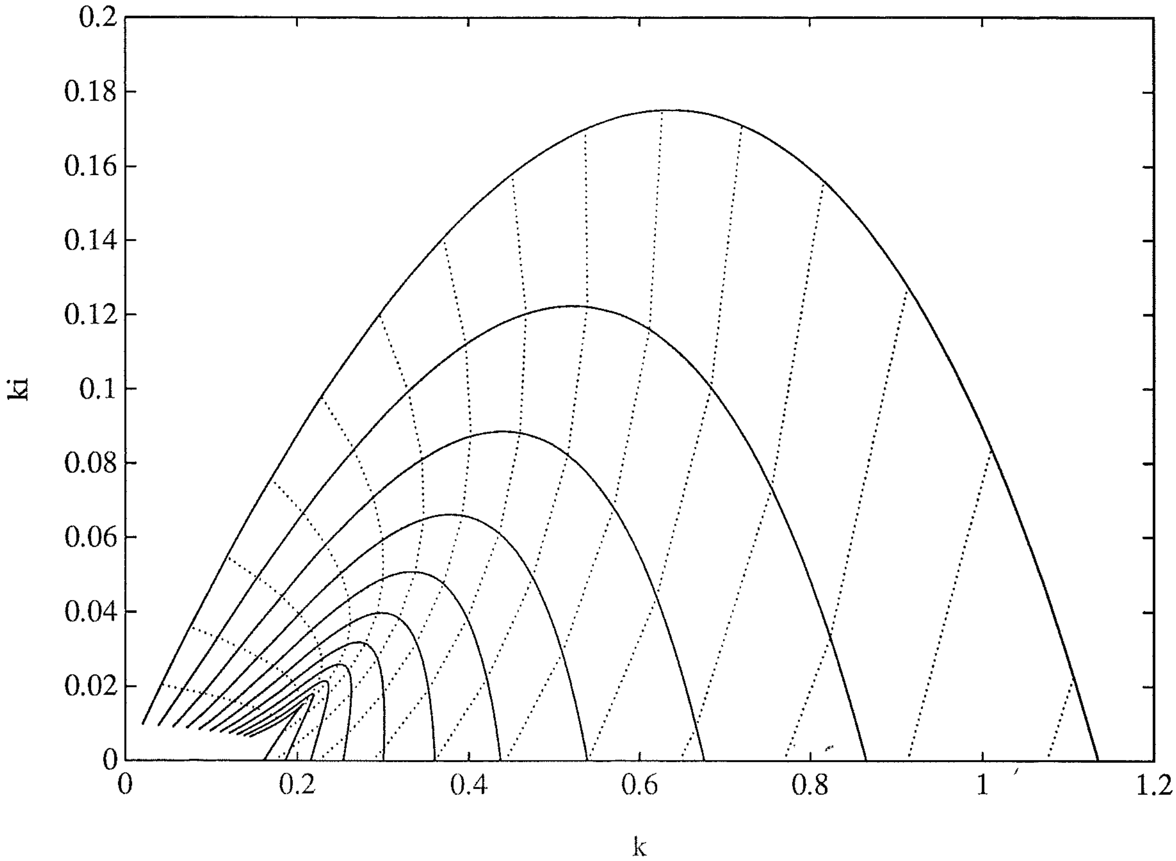



**Figure 4.5** The $k_i(k)$ curves for $G(s) = e^{-s}/s(1+s)$.

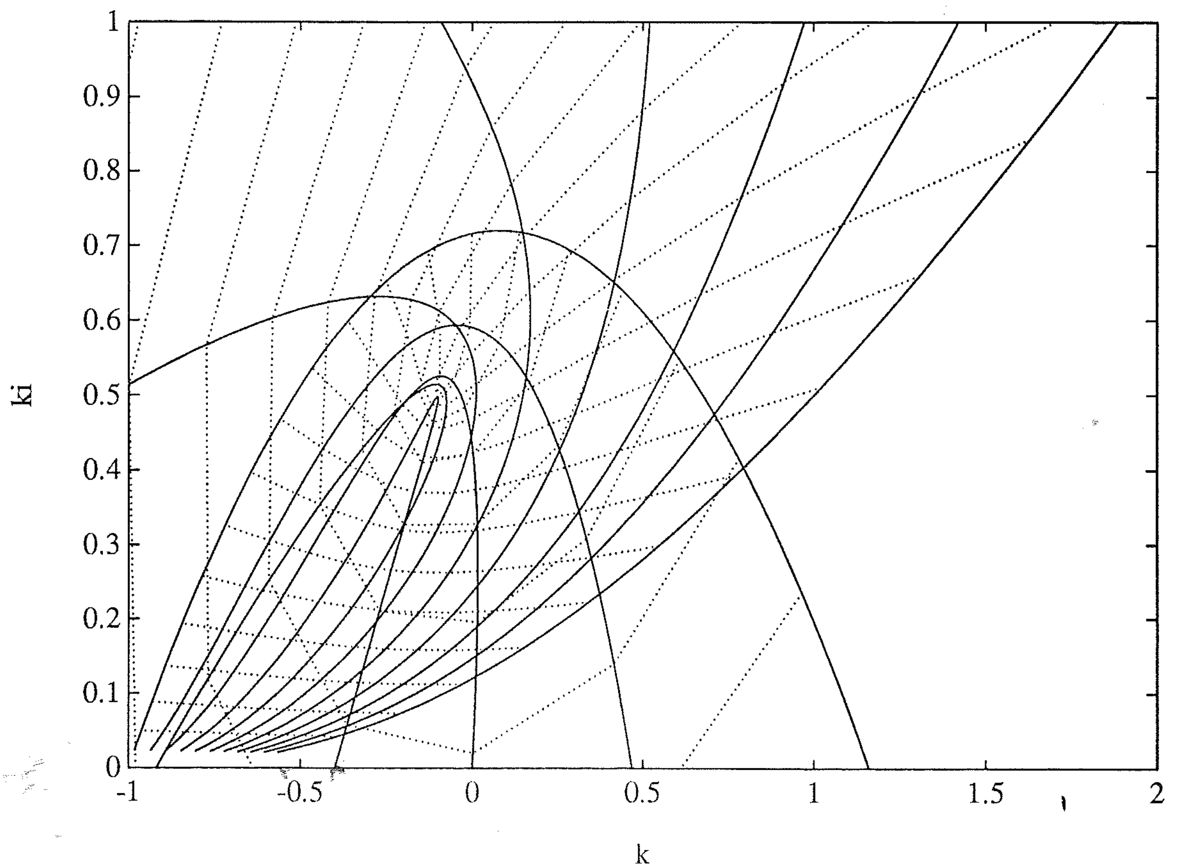



**Figure 4.6** The $k_i(k)$ curves for $G(s) = 0.5/(s+0.5)(s^2 + 2 \cdot 0.2s + 1)$.

## 4.4 Design choices

For a PI and PD controller we have to choose $\omega_0$ and $\zeta_0$. These basic parameters can be chosen in many different ways. In this section a number of methods will be presented and discussed.

When designing a PI or PD controller our task is to choose suitable places for two of the closed loop poles. Different strategies for doing this can be used. The simplest strategy is to compute the closed loop poles and zeros for different values of $\omega_0$ and $\zeta_0$, and choose the set of parameters which give the desired closed loop pole-zero configuration.

The point of using these parameters is that they have an intuitive appeal to the control engineer. The parameter $\zeta_0$ says something about the damping of the system and $\omega_0$ something about the bandwidth. In this section principles for choosing $\omega_0$ and $\zeta_0$ will be discussed.

### Direct tuning of $\omega_0$ and $\zeta_0$

One way of determining $\omega_0$ and $\zeta_0$ is to compute the poles of the closed system for a fixed $\zeta_0$, choosing $\omega_0$ such that the relative location of the poles is according to some specification.

As $\omega_0$ is increased normally one or several poles will move into the right half plane, while two poles will move further into the left half plane. One specification of the relative location of the poles could be 'Choose $\omega_0$ such that the real part of the three poles closest to the origin is equal.' (If a pair of complex poles are moving to the right we must require the real parts of two pairs of poles to be equal.) This method assumes that we can solve the characteristic equation of the closed system numerically, and find all relevant roots. This can be difficult numerically for processes described by non-rational transfer functions. Specifications as the one mentioned here can also give rise to poorly damped poles. This way of selecting $\omega_0$ and $\zeta_0$ will not be investigated further.

### Performance issues

Of the parameters $\omega_0$ and $\zeta_0$ it is $\omega_0$ which is most important for the system's performance. To get high bandwidth of the system it is desirable to move the dominant poles as far out as possible, without causing poor stability due to other poles. Due to this fact, it feels natural to make the choice of $\omega_0$ by some kind of optimization. In this section a number of optimization criteria will be discussed. The criteria has been chosen to be suitable for a frequency domain optimization.

***Maximize*** $k_i$ In the case of PI and PID control there is an intuitively appealing, and often good, way of choosing $\omega_0$, namely the maximization of

$k_i$. In Chapter 2 it has been stated that

$$\mathrm{IE} = \int_0^\infty e(t)\,dt = \frac{1}{k_i}, \tag{4.84}$$

i.e., maximizing $k_i$ is equivalent to minimizing IE. From (4.33) we know that $k_i$ is a function of $\omega_0$ and $\zeta_0$. If we choose $\omega_0$ such that $k_i$ is maximal for a fixed $\zeta_0$ and the system still is stable, we see that the integrated error for a load disturbance becomes as small as possible, with this prescribed damping. This method of optimizing a controller can also be found in [Ziegler and Nichols, 1943], [Cohen and Coon, 1953], and [Wolfe, 1951].

In [Shinskey, 1988] it is pointed out that the IE criterion may be a physically relevant if the quality of the product flowing into a storage tank is to be controlled. If the concentration of the product is controlled then the integrated error $\int^\infty e(t)\,dt$ should be kept as close to zero as possible.

It has also been claimed that the IE criterion is not adequate, e.g., in [Shinskey, 1990] and [Oldenburg and Sartorius, 1954], because the minimization of IE wold give an oscillating system and thus 0 as its cost. In [Shinskey, 1991] we can read:

> "Although the cost function is proportional to integrated error, we can't tune a controller on the basis of the minimum integrated error. The loop could be cycling uniformly and still produce the minimum integrated error, because the positive and negative deviations would cancel each other. Therefore, minimizing the IAE produces an almost identical integrated error, as nearly all of the response curve lies on one side of set point. Most academics seem to prefer minimizing the integral square error. But, that seems more of a mathematical convenience, as it has no recognizable relationship to operating cost. Later it will be shown that minimizing IAE also minimizes peak deviation."

This is certainly true if we allow the poles to be placed on the imaginary axis. However, if we put a restriction on the location of the poles with $\zeta_0$ and minimizes over $\omega_0$ very good results are obtained for most systems. If the controller is optimized with respect to both $\omega_0$ and $\zeta_0$ the IE criterion will give $\zeta_0 = 0$, and a useless design.

Due to the extremely complicated expressions for $k$ and $k_i$ it is not possible to carry out an analytical analysis for this method of choosing $\omega_0$ for other processes than second and third order systems. On the other hand, the real strength of this method for choosing $\omega_0$ is that it is very easy to handle numerically. The functions $k(\omega_0, \zeta_0)$ and $k_i(\omega_0, \zeta_0)$ can easily and efficiently be computed for any arguments. It is also very easy to optimize $k_i$ with respect to $\omega_0$ for a given $\zeta_0$, using, e.g., a golden section algorithm, see

[Luenberger, 1984]. This is how the routines for the numerical optimization of $k_i$ as been implemented.

The existence of a maximum of $k_i$ does not imply stability. For systems with time-delays and poles on the negative real axis there does not seem to be any problems, i.e., all $k_i(k)$ curves lie inside the curve corresponding to $\zeta_0 = 0$ and a maximal $k_i$ corresponds to a stable system. In Example 4.12 we show a system and a design case where the maximal $k_i$ corresponds to an unstable closed system. By proper ways of selecting $\zeta_0$ such cases can also be handled, within the natural limitations of a PI controller.

From experience we have noticed that when we design a PI controller and maximize $k_i$, and thus specify two poles, the pole on the negative real axis is at such a distance from the origin that the bandwidth will still be reasonable, if a large enough $\zeta_0$ is chosen.

EXAMPLE 4.12—Instability from maximal $k_i$

Consider

$$G(s) = \frac{\alpha_p}{(s+\alpha_p)(s^2+2\zeta_p s+1)} \tag{4.85}$$

with $\alpha_p = 0.2$ and $\zeta_p = 0.2$. Maximizing $k_i$ for $\zeta_0 = 0.4$ give an unstable system. In Figure 4.7 the relevant $k_i(k)$ curves can be found. Phenomena like this occur for resonant systems when poles are specified with larger relative damping than the resonant poles of the plant. □

From experience we see that systems designed with $\omega_0$ chosen by maximizing $k_i$ give a little too much overshoot in the load responses for a $\zeta_0$ compatible with those obtained from the integral criteria. Controllers designed with IAE, ISE, or ITAE behave better in this respect. These controllers correspond to $\omega_0$ which are greater than the $k_i$-optimal for the corresponding $\zeta_0$. The parameters $\omega_0$ and $\zeta_0$ that correspond to the $k$ and $k_i$ of the IAE optimal controller have been computed. The $\zeta_0$ values were used to design a $k_i$ optimal controller. The quantity $k_{i,\text{IAE}}/k_{i,\text{IE}}$ was computed for $0.1 < L/T < 1$ and was found to be between 0.68 and 0.81. This leads to a simple rule for modifying a $k_i$-optimal PI controller; compute $\omega_0$ to maximize $k_i$, then increase $\omega_0$ for constant $\zeta_0$, until $k_i = 0.8k_{i,\max}$. This is very easy to implement numerically. This modification works well for processes of high order or processes with time delays. For processes of low order (2 or 3) the response is unnecessarily slow.

The point of using IE is that the design computations can be carried out completely in the frequency domain, without time domain optimizations.

***Maximize the bandwidth of the closed system*** The bandwidth $\omega_b$ of the system $G(s)$ is defined as the smallest solution of

$$\left|\frac{G(i\omega_b)}{G(0)}\right| = \frac{1}{\sqrt{2}}. \tag{4.86}$$

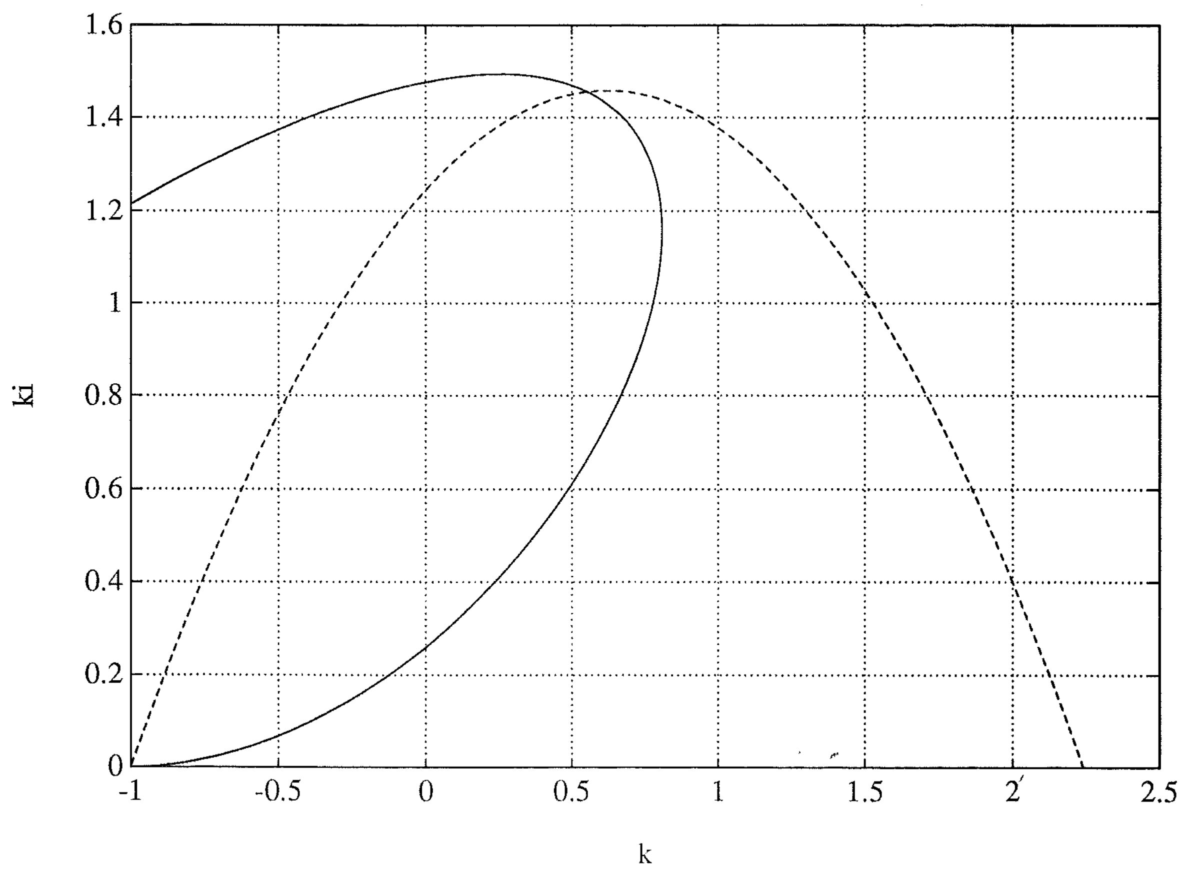



**Figure 4.7** An example of a system with a maximal $k_i$ corresponding to an unstable system. The solid line is $k_i(k)$ for $\zeta_0 = 0.4$, and the dashed line is the stability limit.

This is the definition used in [Truxal, 1955], [Horowitz, 1963], and [Åström, 1976]. This definition of bandwidth does not require stability of the system $G(s)$. For example, both $G(s) = 1/(1+s)$ and $G(s) = 1/(1-s)$ have $\omega_b = 1$ according to (4.86). If we want to use this definition for numerical computations the stability of the result must always be checked. In [Rivera *et al.*, 1986] the bandwidth of a closed loop system is defined as the first solution of

$$\left|\frac{1}{1+L(i\omega_b)}\right| = \frac{1}{\sqrt{2}} \tag{4.87}$$

where $L(s)$ is the loop transfer function

A very common configuration of the poles of the closed loop system is when we have two complex poles moving away from the origin as $\omega_0$ is increased, and one pole moving towards the origin at the same time. These three poles and possibly zeros from the controller have the greatest impact on the closed loop bandwidth. In certain cases the bandwidth has a maximum as $\omega_0$ increases. Thus, one way of choosing $\omega_0$ is to determine the maximum bandwidth of the controlled system with respect to $\omega_0$. However, this method of choosing $\omega_0$ is not altogether unproblematic, as Example 4.13 will show.

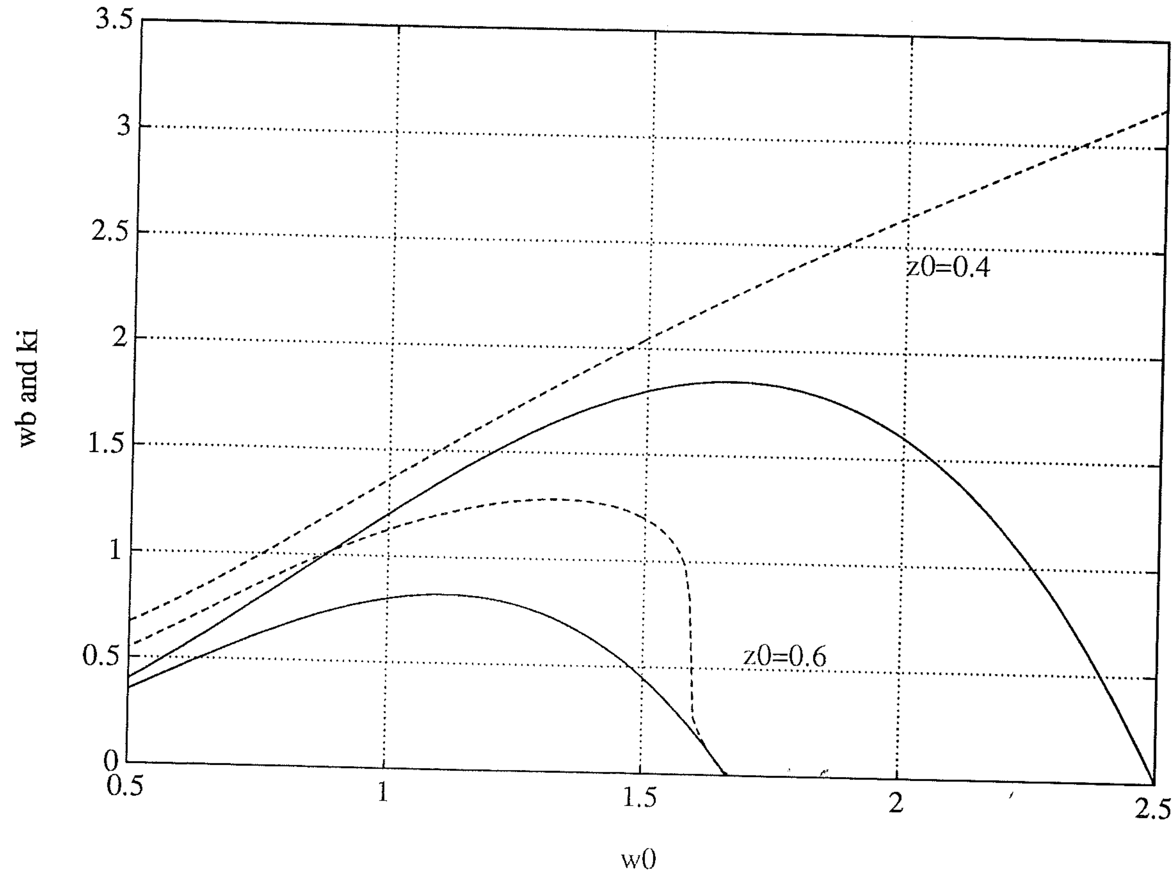



**Figure 4.8** The $k_i$ (solid) and $\omega_b$ (dashed) for $\zeta_0 = 0.4$ and $\zeta_0 = 0.6$.

EXAMPLE 4.13—The danger of using $\omega_b$
Consider the process

$$G(s) = \frac{1}{(s+1)^2}. \tag{4.88}$$

Let the process be controlled by a PI controller such that two poles will be placed in $\omega_0(-\zeta_0 \pm i\sqrt{1-\zeta_0^2})$. According to Example 4.21 the additional pole of the closed system will be located in

$$p_3 = 2(1 - \omega_0\zeta_0) \tag{4.89}$$

and the controller zero in

$$z_1 = 2\frac{(\omega_0\zeta_0 - 1)\omega_0^2}{4\omega_0^2\zeta_0^2 - \omega_0^2 - 4\omega_0\zeta_0 + 1}. \tag{4.90}$$

In Figure 4.8 $k_i$ and $\omega_b$ are plotted as functions of $\omega_0$. In the case where $\zeta_0 = 0.6$ both $k_i$ and $\omega_b$ has a maximum for an $\omega_0$ which corresponds to a stable system ($k_i > 0$). In the case where $\zeta_0 = 0.4$ there is no maximum of $\omega_b$ for a stable system. The reason is that in the case of $\zeta_0 = 0.4$ the controller zero and the pole which is moving to the right are very close in a much larger interval of $\omega_0$ than in the case of $\zeta_0 = 0.6$. The poles and

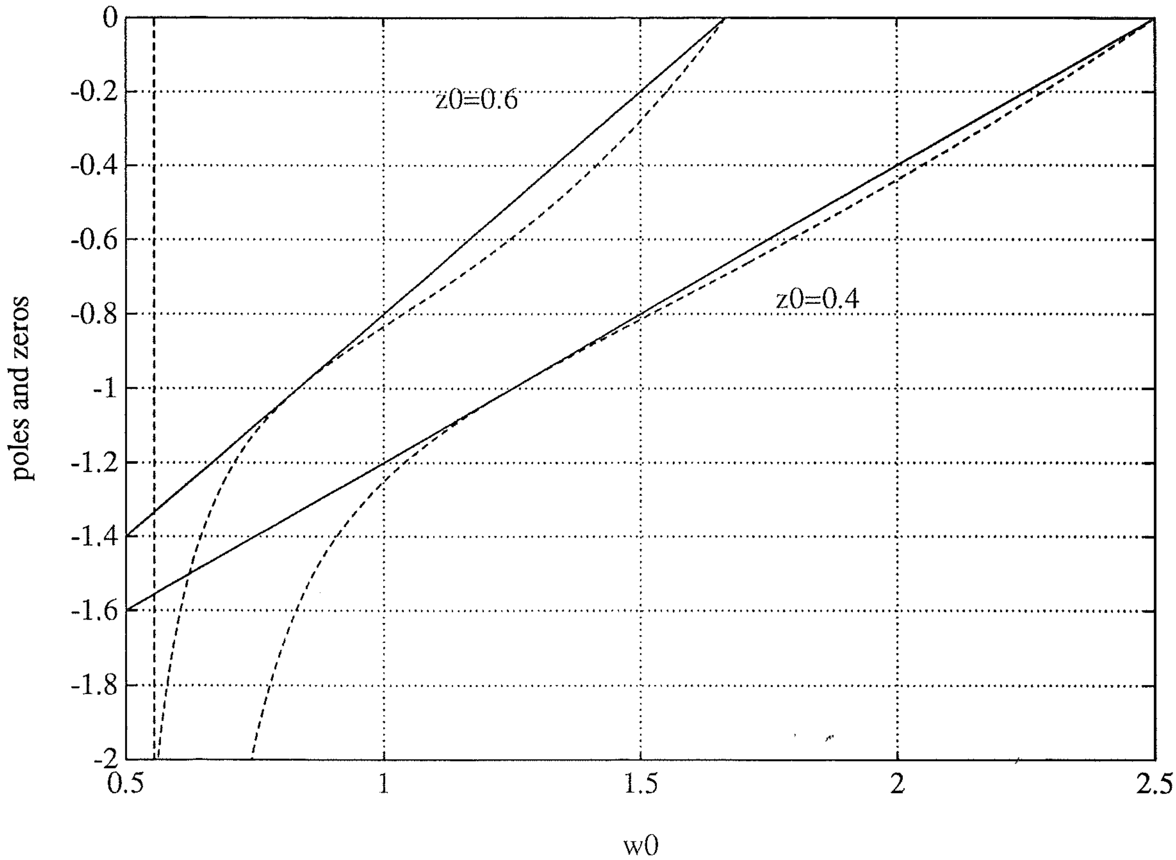



**Figure 4.9** The pole on the real axis (solid) and the controller zero (dashed) for $\zeta_0 = 0.4$ and $\zeta_0 = 0.6$.

zeros on the real axis are shown in Figure 4.9. In the case of $\zeta_0 = 0.4$ the behaviour of the closed system will be completely determined by the complex dominant poles, and the bandwidth criterion will not detect that the system is becoming unstable.

This example has shown that the criterion of maximizing the bandwidth, although intuitively appealing, cannot always be trusted due to pole-zero cancellations. □

From other examples it has been seen that the optimization of bandwidth and the optimization of $k_i$ give approximately the same controllers if $\zeta_0 > 0.5$. The optimization of bandwidth is 10 to 100 times more computationally demanding than the $k_i$ optimization.

When $\omega_b$ is used in a criterion for determining $\omega_0$ we use the bandwidth of

$$G_{\text{CL}} = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)}. \tag{4.91}$$

When we use a PID controller in practice we often make a set point weighting ('$\beta$ modification') and the closed loop transfer function will be

$$G_{\text{CL}} = \frac{k(\beta + \frac{1}{sT_i})G(s)}{1 + k(1 + \frac{1}{sT_i})G(s)}. \tag{4.92}$$

Making the $\beta$ modification *after* a bandwidth optimization with respect to $\omega_0$ is meaningless, because $\beta$ affects the bandwidth. This problem could of course be solved by an optimization in two variables ($\omega_0$ and $\beta$), but would be very computationally demanding and complicated.

For a system with many resonance peaks in its Bode diagram, a common situation when a time delay is in the loop, there can be doubts of which solution of (4.88) to choose. For this reason another definition of bandwidth is proposed. A sensible definition of bandwidth could be the $\omega_b$ which satisfies the equation

$$x = \frac{\int_0^{\omega_b} |G(i\omega)|^2 \, d\omega}{\int_0^{\infty} |G(i\omega)|^2 \, d\omega}, \tag{4.93}$$

for an $x$ in the interval 0 to 1. For $x = 1/2$ we get the same bandwidth for $G(s) = 1/(1+s)$ as with (4.88). This method involves more computations than the standard equation, but can easily be implemented numerically. If there are reasons for computing bandwidth, this is probably a more robust method than (4.88).

***Minimize*** $e_p$ The $e_p$ is defined as the peak value of the control error after a load disturbance at the process input. According to [Shinskey, 1990] the minimization of $e_p$ is of prime concern. This minimization requires numerical solution of the equations describing the process, which we try to avoid. Therefore this criterion will not be considered further.

***Conclusions*** For reasons presented in this section we recommend the maximization of $k_i$ with respect to $\omega_0$ as a good performance criterion. In certain cases further improvement of the control can be obtained if $\omega_0$ is modified as indicated earlier.

## Robustness issues

While the parameter $\omega_0$ was chosen to get good performance, $\zeta_0$ will be chosen to get good robustness of the closed system. A number of principles of chosing $\zeta_0$ will be discussed in this section.

***Direct choice of*** $\zeta_0$ The step response of a second order system looks reasonable if the relative damping is about 0.7. This may lead us to routinely choose $\zeta_0 = 0.7$, since we are designing systems by specifying two poles. In Example 4.14 it is shown that this may lead to unnecessarily poor performance of the closed loop system. The parameter $\zeta_0$ is not a good design parameter in itself.

EXAMPLE 4.14—Design with $\zeta_0$
When servo systems are designed the required behaviour of the closed system is usually that of a system with two poles with relative damping 0.7.

The two systems

$$G_1 = \frac{e^{-sL}}{1+sT} \tag{4.94}$$

$$G_2 = \frac{\omega_p^3\alpha_p}{(s+\omega_p\alpha_p)(s^2+2\omega_p\zeta_p s+\omega_p^2)} \tag{4.95}$$

with $L = T = 1$ for $G_1$ and $\omega_p = 1, \alpha_p = 3, \zeta_p = 0.7$ for $G_2$ are given.

If PI controllers are designed with $\zeta_0 = 0.7$ and the $\omega_0$ which maximizes $k_i$ the set point and load responses of the controlled system will be those shown in Figure 4.10.

As can be seen two system designed with the same $\zeta_0$ may behave quite differently. PI controllers may instead be designed with $k_i$ maximized with respect to $\omega_0$ and $\zeta_0$ chosen such that the $M_s$ value of the compensated system is 1.6. The step and load responses are also shown in Figure 4.10. Data of the two designs are presented in Table 4.1, where the design parameters provided by the user are written in boldface.

Table 4.1

| System | $\omega_0$ | $\zeta_0$ | $M_s$ | $k$ | $T_i$ | IE | IAE |
|---|---|---|---|---|---|---|---|
| $G_1$ | 1.05 | **0.70** | 1.61 | 0.50 | 0.96 | 1.94 | 2.03 |
| $G_2$ | 0.70 | **0.70** | 1.37 | 0.16 | 0.57 | 3.47 | 3.52 |
| $G_1$ | 1.05 | 0.71 | **1.60** | 0.49 | 0.97 | 1.97 | 2.04 |
| $G_2$ | 0.90 | 0.43 | **1.60** | 0.55 | 1.02 | 1.86 | 2.14 |

In this case the two controlled systems behave reasonably similar if they are designed with a prescribed $M_s$ value, but differently when they are designed with the same $\zeta_0$. Different systems may react very differently to the $\zeta_0$ parameter. The $\zeta_0$ parameter should not be regarded as the relative damping of the system, but rather as a parameter in the controller by which we may control the stability and robustness of the closed system. □

***Design with specification on*** $M_s$ Define $M_s$ as the maximum of the sensitivity function $S(s)$,

$$M_s = \max_{\omega>0}|S(i\omega)| = \max_{\omega>0}\left|\frac{1}{1+L(i\omega)}\right|, \tag{4.96}$$

where $L(s) = G_c(s)G(s)$ is the loop transfer function. Another interpretation of this is to say that $M_s$ is the inverted value of the shortest distance from $-1$ to the Nyquist curve of $L(s)$.

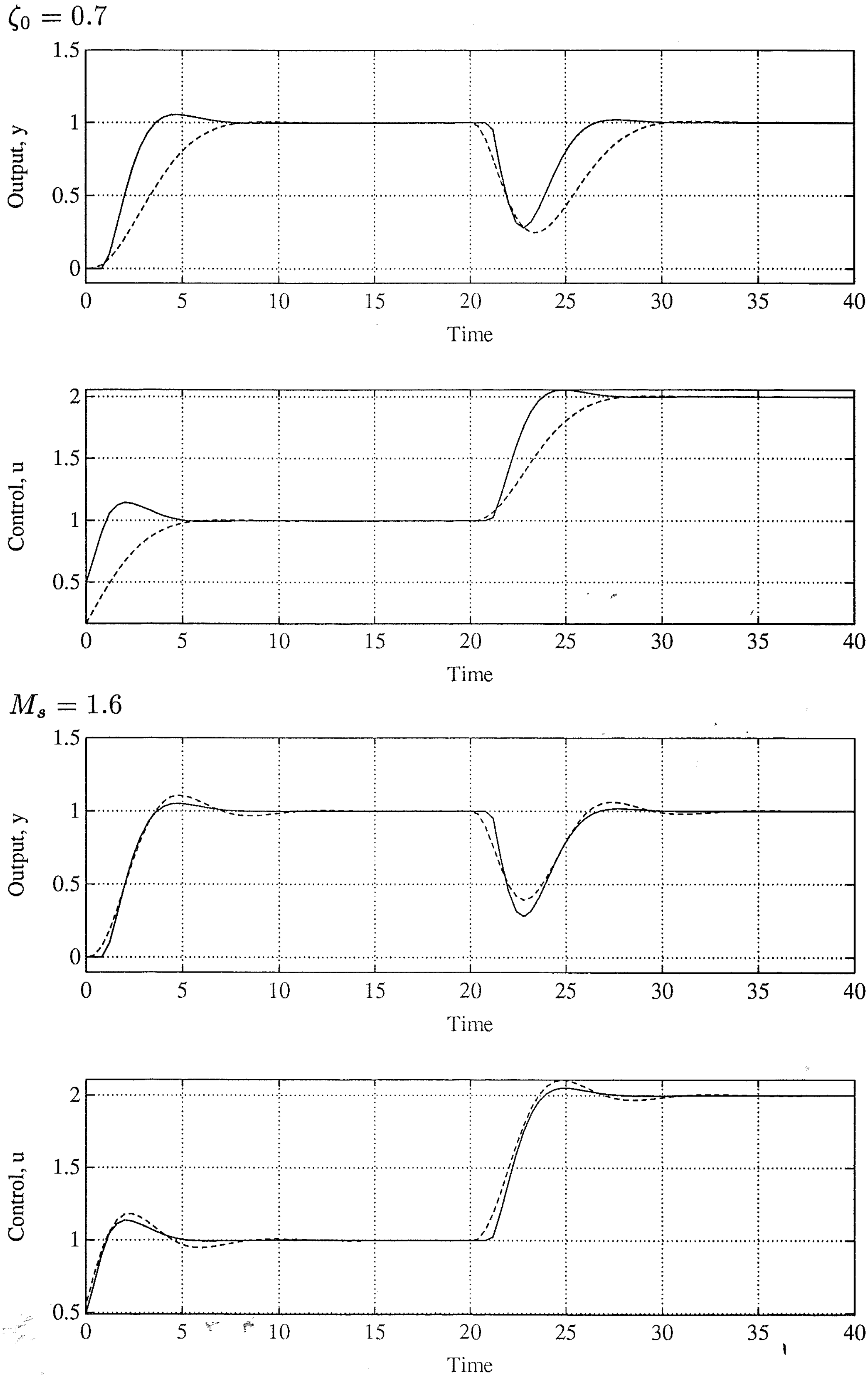



**Figure 4.10** The processes $G_1$ (solid line) and $G_2$ (dashed line) controlled by PI controllers with $\zeta_0 = 0.7$ and controlled by PI controllers designed such that such that $M_s = 1.6$.

If a system has a certain $M_s$ then we also have bounds on the amplitude and phase margins, namely

$$A_m \geq \frac{M_s}{M_s - 1} \tag{4.97}$$

$$\varphi_m \geq 2 \arcsin \frac{1}{2M_s}. \tag{4.98}$$

The controller is now determined by computing $k(\omega_0, \zeta_0)$ and $k_i(\omega_0, \zeta_0)$ such that for each $\omega_0$ the closed system has the prescribed $M_s$ value. This implies solving

$$M_s = \max_{\omega} \left| \frac{1}{1 + G(i\omega)G_c(i\omega)} \right| \tag{4.99}$$

with respect to $\zeta_0$ for each $\omega_0$ to get $\zeta_0 = \zeta_0(\omega_0, M_s)$. The function $k_i = k_i(\omega_0, \zeta_0(\omega_0, M_s))$ is then optimized with respect to $\omega_0$ for a given $M_s$. In doing this $\zeta_0$ is chosen to fulfill the condition on $M_s$, and $\omega_0$ is the free parameter. This way of computing $\omega_0$ and $\zeta_0$ is extremely computationally demanding. A nonlinear equation must be solved each time a $k_i$ value is needed in the optimization.

An approximate way to compute the controller is to optimize $k_i(\omega_0, \zeta_0)$ for constant $\zeta_0$ and choose $\zeta_0$ such that the closed system has the prescribed $M_s$. This implies solving (4.99) with respect to $\zeta_0$, when $\omega_0$ is chosen to maximize $k_i$ for a constant $\zeta_0$. The equation can be solved numerically, e.g., with the bisection method. This approximate method requires about 1/10 of the computations in the complete case, and is much more robust with respect to initial guesses in the equation solving. The following example will illustrate this approximation.

EXAMPLE 4.15—Controller settings for a given $M_s$
Consider the processes

$$G_1(s) = \frac{e^{-0.5s}}{1 + s} \tag{4.100}$$

$$G_2(2) = \frac{1}{(1 + s)^8}. \tag{4.101}$$

In Figure 4.11 loci in the $(k, k_i)$ plane for constant values of $M_s$ of the process $G_1(s)$ can be found. Above in the figure $M_s$ assumes the values $M_s = 2.0$, 2.5, 3.0, 3.5, 4.0, 4.5, and 5.0 (solid lines). Below in the figure $M_s = 1.6$, 1.8, 2.0, 2.2, and 2.4. PI designs with the exact method is marked with 'o' and the approximate method is marked with '+'. Curves for constant $\zeta_0$ corresponding to the $\zeta_0$ obtained with the approximate design are shown with dashed lines. Points on the $M_s$ curves with constant $\omega_0$ are joined by dotted lines.

In Figure 4.12 loci in the $(k, k_i)$-plane for constant values of $M_s$ of the process $G_2(s)$ can be found. The $M_s$ assumes the values $M_s = 1.6$, 1.8, 2.0, 2.2, and 2.4. The curve markings are the same as those in Figure 4.11.

As can be seen from the figures the simplified version yields solutions close to the solutions of the exact problem. □

This way of selecting $\zeta_0$ can be applied to all principles of choosing $\zeta_0$ presented in this section. The 'correct' way would be: for every $\omega_0$ select a $\zeta_0$ such that the robustness condition expressed in a parameter $P$, is fulfilled. This gives $\zeta_0 = \zeta_0(\omega_0, P)$. Then maximize $k_i = k_i(\omega_0, \zeta_0(\omega_0, P))$ with respect to $\omega_0$. The computation of $\zeta_0(\omega_0, P)$ can be very complicated and may take long time.

The approximation which will be used in the following is: optimize $k_i$ with respect to $\omega_0$ for constant $\zeta_0$. Choose $\zeta_0$ such that the condition is fulfilled at the extremum. This is moderately computationally demanding and gives reasonably accurate results.

Suppose we want to use the idea for modifying $\omega_0$, presented before. If we follow a curve for constant $\zeta_0$ when $\omega_0$ is increased the resulting controller will give a system with a lower value of $M_s$ than the system with the $k_i$ optimal controller. Hence, when we do this kind of modification we should really follow a curve for constant $M_s$ instead. This requires, however, much computation.

The parameter $M_s$ cannot be chosen completely freely. By letting $\zeta_0 = 0$ the Nyquist curve of the loop transfer function goes through $-1$ which gives $M_s = \infty$. Thus $M_s$ can be made arbitrarily high. For most non resonant systems we get smaller values of $M_s$ at the point corresponding to maximal $k_i$ as $\zeta_0$ is increased. The smallest value is normally obtained for $\zeta_0 = 1$. At other points on the $k_i(k)$ curve corresponding to $\zeta_0 = 1$ the $M_s$ value may of course be even lower.

EXAMPLE 4.16—Possible values of $M_s$ – second order system
To get some intuition in how to choose $M_s$, consider the loop transfer function which causes the transfer function from the set point signal to the output signal to be

$$G(s) = \frac{\omega^2}{s^2 + 2\omega\zeta s + \omega^2}. \tag{4.102}$$

This loop transfer function is

$$L(s) = \frac{\omega^2}{s(s + 2\omega\zeta)}. \tag{4.103}$$

In Figure 4.13 $M_s$ is shown as a function of $\zeta$. This plot indicates that a

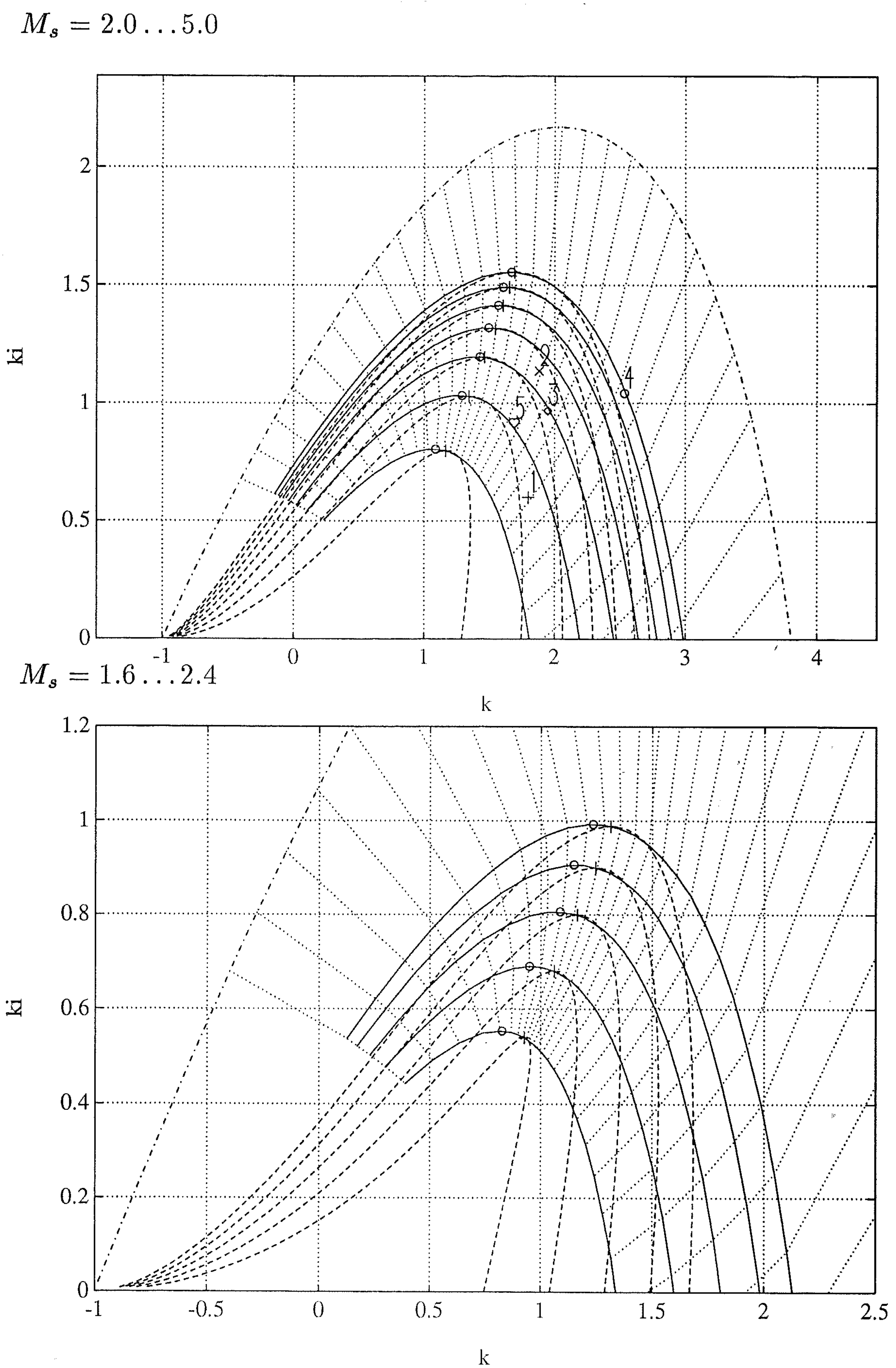



**Figure 4.11** Controller settings for $G_1(s)$. The figure shows curves for different values of $M_s$ (solid lines), for constant $\zeta_0$ (dashed lines), and $\zeta_0 = 0$, the stability limit (dashed-dotted line). Exact PI designs are marked with 'o' and the approximate design with '+'. Standard tuning rules are shown as Ziegler-Nichols (1), Cohen-Coon (2), IAE (3), ISE (4), and ITAE (5).

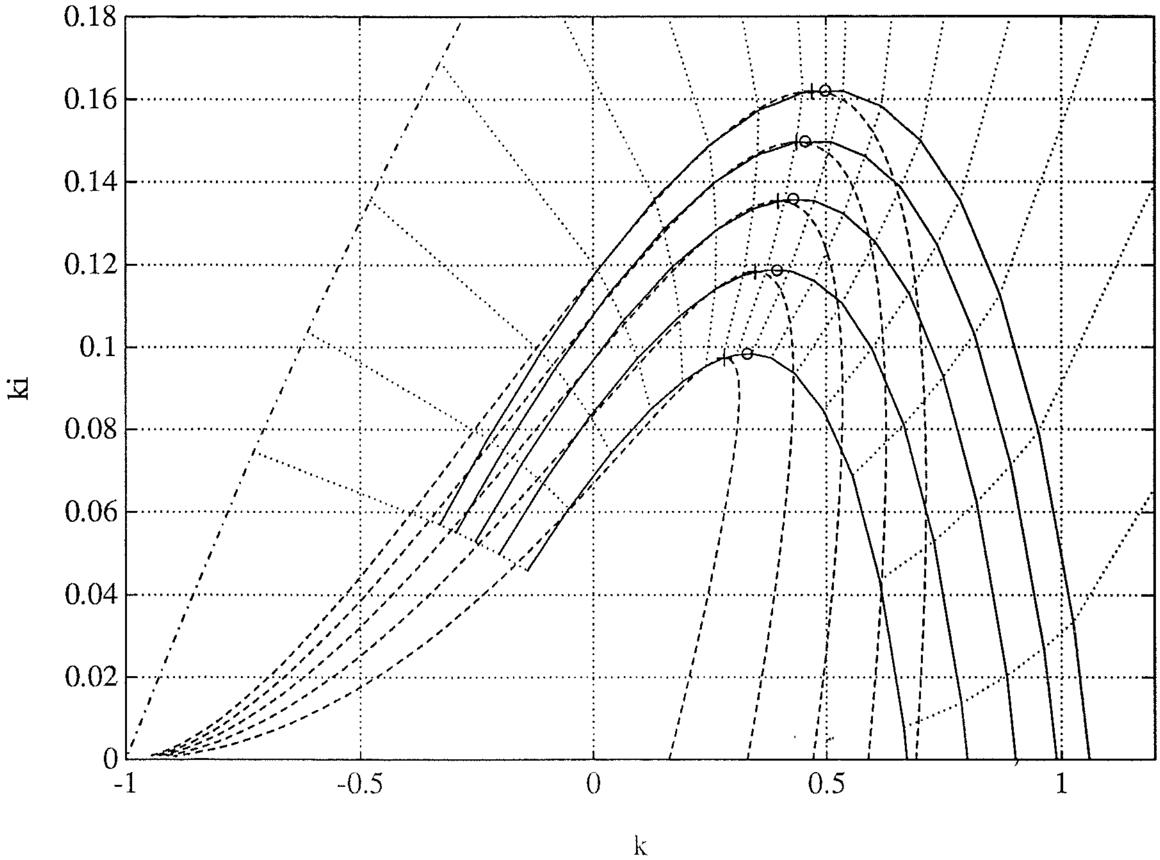



**Figure 4.12** Controller settings for $G_2(s)$. Curves for constant $M_s = 1.6$, 1.8, 2.0, 2.2, and 2.4. Exact PI designs are marked with an 'o' and the approximate design with a '+'.

choice of $M_s$ in the interval 1.1 to 2 would give a second order system with acceptable behaviour. □

From Example 4.16 one would assume that $M_s = 1.3$ would be a good choice. However, as more poles are added to the system the $M_s$ for a given $\zeta_0$ is increased. This makes the results from Example 4.16 overly conservative.

EXAMPLE 4.17—Possible values of $M_s$
Consider the standard example

$$G(s) = \frac{e^{-sL}}{1+s}. \tag{4.104}$$

In Figure 4.14 are shown the $M_s$ values of $G(s)$ controlled by a PI controller. The PI controller has been designed by maximizing $k_i$ for $\zeta_0 = 0.3, \ldots, 1.0$.

From the figure we can see that the minimal $M_s$ value increases dramatically when $\zeta_0$ is decreased. The $M_s$ value does not show that much variation for long time delays. □

When there are many control loops interacting in a plant it is often desired that certain loops should not cause oscillations. To avoid this, loops which may cause oscillations should be tuned with a smaller value of $M_s$,

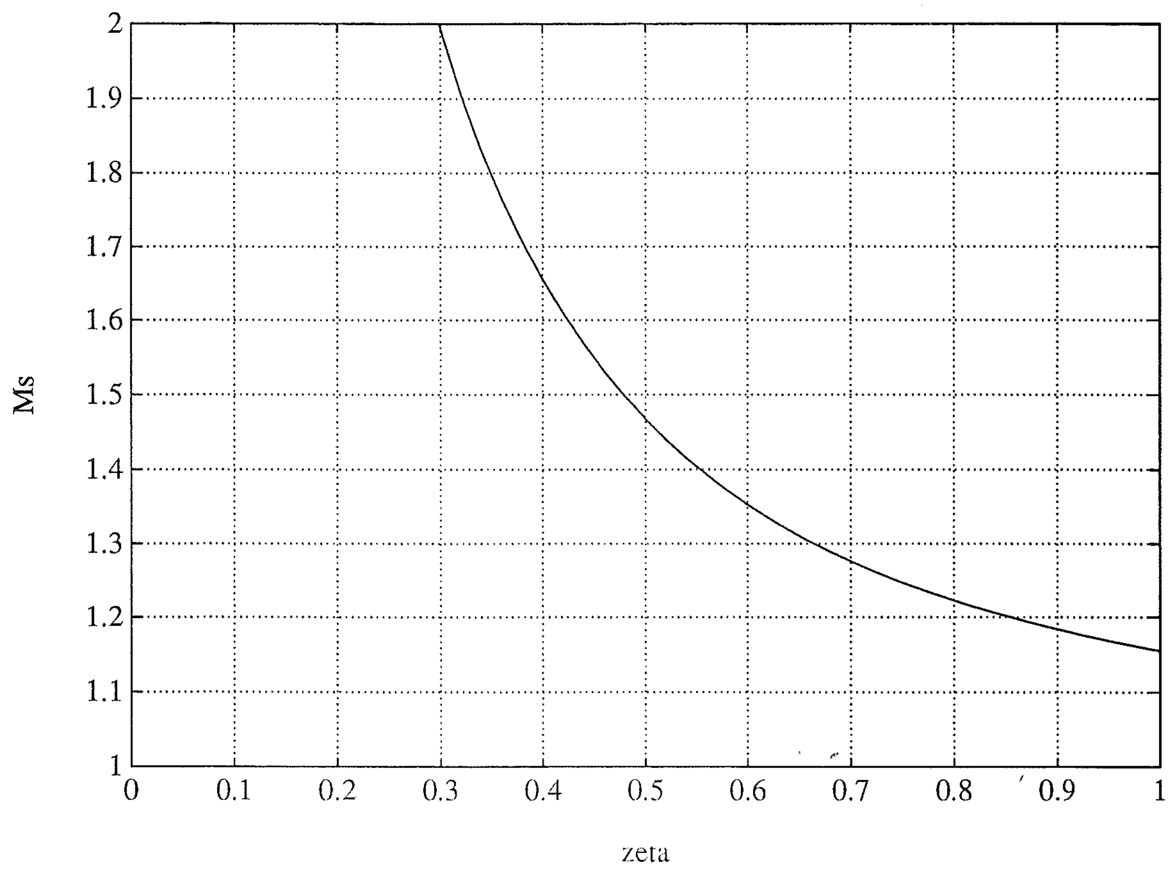



**Figure 4.13** $M_s$ as function of $\zeta$ for $L(s) = \omega/s(s + 2\omega\zeta)$.

normally $M_s \approx 1.6$. Loops for which rapid elimination of disturbances is the key issue should be tuned with a larger $M_s$. A good choice is often $M_s = 2$.

***Design with specification on*** $M_p$ The $M_p$ value of a transfer function is defined as

$$M_p = \max_{\omega} |G(s)|. \tag{4.105}$$

One way of choosing $\zeta_0$ could be to make a dominant pole design, e.g., by maximizing $k_i$, and choose $\zeta_0$ such that $M_p$ of the closed system will have a prescribed value.

Just like in the case of the $M_s$ values $M_p$ values puts limits on the phase and amplitude margins, namely

$$A_m \geq 1 + \frac{1}{M_p} \tag{4.106}$$

$$\varphi_m \geq 2 \arcsin \frac{1}{2M_p}. \tag{4.107}$$

These values are obtained from the points where the $M_p$ circles,

$$(x + \frac{M_p^2}{M_p^2 - 1})^2 + y^2 = \frac{M_p^2}{(M_p^2 - 1)^2} \tag{4.108}$$

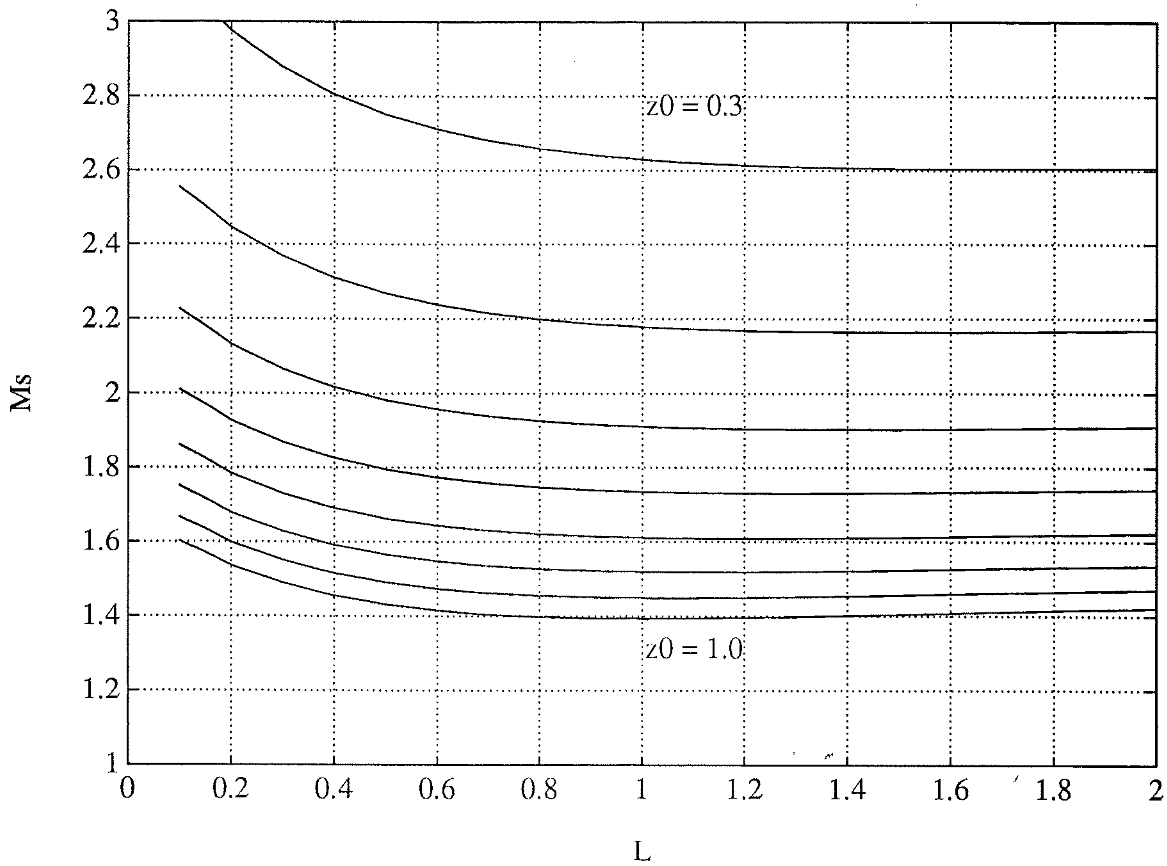



**Figure 4.14** $M_s$ as function of $L$ for a $k_i$ optimal controller for $\zeta_0 = 0.3 \ldots 1.0$.

intersect $y = 0$ and $x^2 + y^2 = 1$. The values are rather conservative. This is also discussed in [Seborg *et al.*, 1989] in Chapter 16, where a design procedure based on $M_p$ values is presented.

When the design with specifications on $M_p$ is done, we design with respect to the transfer function $G_cG/(1 + G_cG)$, and assume that $\beta = 1$. Another value of $\beta$ will most certainly change the $M_p$-value, and the $M_p$ used for design will no longer determine the closed loop set point response. For a unit feedback system the $M_p$ value determines the robustness of the system. Therefore, using $M_p$ as a design parameter may still have a relevance.

Computationally, using $M_p$ or $M_s$ as specification parameter is approximately equivalent. Both $M_p$ (with $\beta = 1$) and $M_s$ can be estimated from a Nyquist diagram of the loop transfer function. The $M_p$ value is strongly coupled to the set point response which is dependent on $\beta$. The $M_s$ value is much easier to visualize in a Nyquist diagram than the $M_p$ value. Therefore we have chosen to use $M_s$ as the design parameter. Later we will propose a method for choosing the $\beta$ parameter based on $M_p$ values.

***Design with specification on $\varphi_m$*** One standard measurement of robustness and stability of a controlled system is the phase margin, $\varphi_m$. One possibility to choose $\zeta_0$ would be to specify the phase margin of the closed system.

In Example 4.18 we give an example of the drawbacks of using $\varphi_m$ as the design parameter. The problem is, of course, that we use only one point on the Nyquist curve to specify the design. Especially for processes with integration this method is not good.

EXAMPLE 4.18—Design with specification on $\varphi_m$
Consider the two systems

$$G_1(s) = \frac{e^{-s}}{0.1s+1} \tag{4.109}$$

$$G_2(s) = \frac{1}{s(0.2s+1)^2}. \tag{4.110}$$

PI controllers were designed for these systems where $\zeta_0$ was chosen to get a specified phase margin for an $\omega_0$ which maximizes $k_i$.

**Table 4.2**

| System | $\omega_0$ | $\zeta_0$ | $\varphi_m$ | $M_s$ | $k$ | $T_i$ | IE | IAE |
|---|---|---|---|---|---|---|---|---|
| $G_1$ | 1.72 | 0.80 | 63.44 | **1.60** | 0.18 | 0.31 | 1.74 | 1.76 |
| $G_2$ | 1.33 | 0.73 | 41.88 | **1.60** | 1.16 | 2.15 | 1.84 | 1.86 |
| $G_1$ | 1.73 | 0.28 | **40.00** | 2.83 | 0.32 | 0.31 | 0.96 | 2.09 |
| $G_2$ | 1.40 | 0.66 | **40.00** | 1.66 | 1.23 | 1.99 | 1.61 | 1.64 |

From Table 4.2 and the simulations in Figure 4.15 we can see that when $\varphi_m$ is the primary design parameter we may get large $M_s$ values as well as relatively small ones, and very different behaviour in the step responses. If, however, we design with respect to a given $M_s$ we get reasonably similar step responses in both cases. □

***Design with specification on*** $A_m$ Another measurement of robustness and stability is the amplitude margin, $A_m$. One way to specify the behaviour of the closed system is to prescribe an $A_m$ and choose $\zeta_0$ such that this value of $A_m$ is obtained.

EXAMPLE 4.19—Design with specification on $A_m$
Consider the two systems

$$G_1(s) = \frac{e^{-s}}{0.1s+1} \tag{4.111}$$

$$G_2(s) = \frac{1}{s(0.2s+1)^2}. \tag{4.112}$$

PI controllers were designed for these systems where $\zeta_0$ then chosen to get a specified amplitude margin for an $\omega_0$ which maximizes $k_i$.

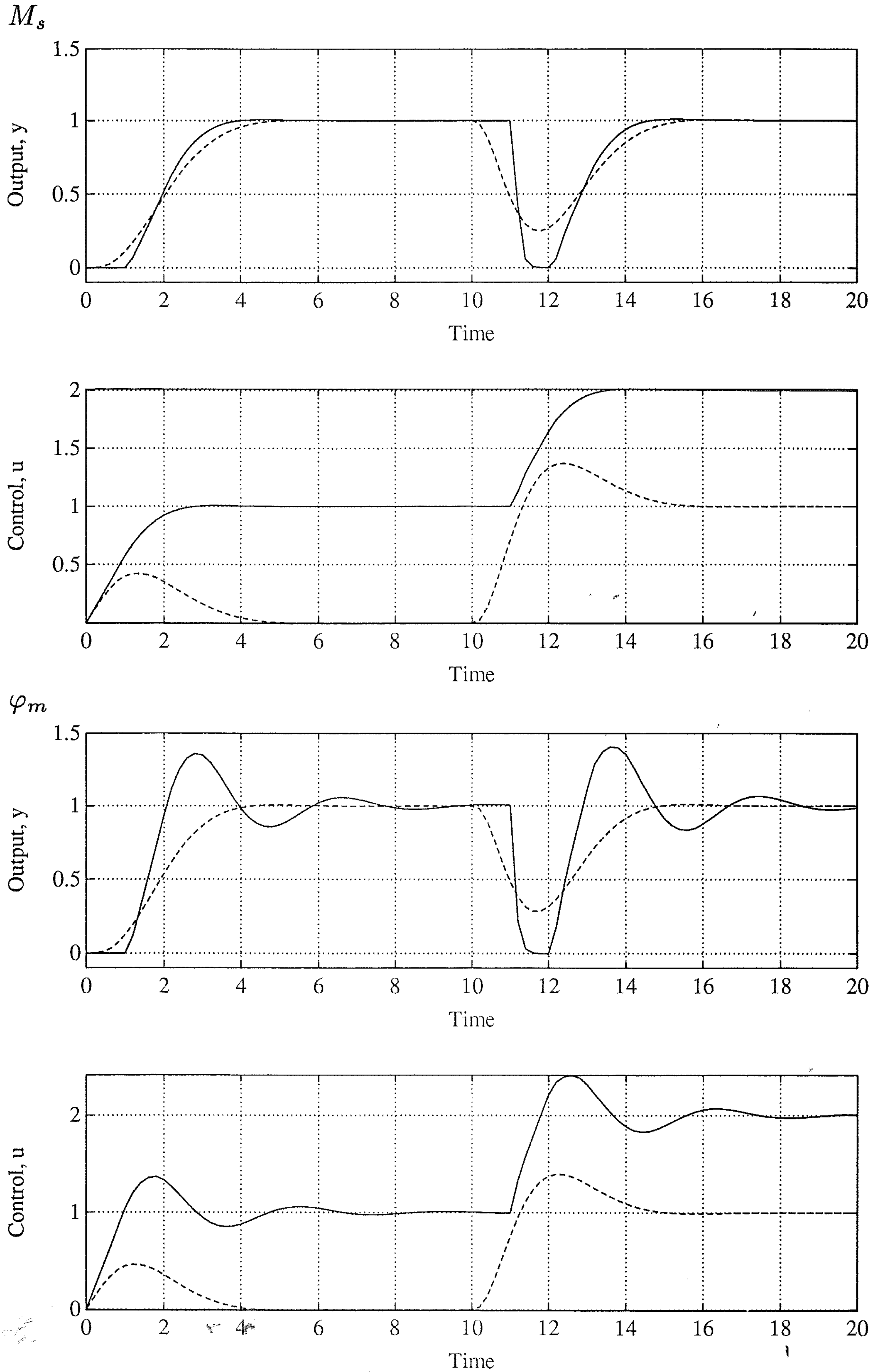



**Figure 4.15** Design of PI controllers for $G_1$ (solid) and $G_2$ (dashed). Design with specified $M_s$ above in the figure and with specified $\varphi_m$ below in the figure.

Table 4.3

| System | $\omega_0$ | $\zeta_0$ | $A_m$ | $M_s$ | $k$ | $T_i$ | IE | IAE |
|---|---|---|---|---|---|---|---|---|
| $G_1$ | 1.72 | 0.80 | 2.91 | **1.60** | 0.18 | 0.31 | 1.74 | 1.76 |
| $G_2$ | 1.33 | 0.73 | 7.01 | **1.60** | 1.16 | 2.15 | 1.84 | 1.86 |
| $G_1$ | 1.70 | 0.71 | **2.70** | 1.67 | 0.20 | 0.32 | 1.61 | 1.69 |
| $G_2$ | 2.33 | 0.23 | **2.70** | 3.07 | 2.37 | 1.11 | 0.47 | 0.72 |

The time responses of the controlled system are shown in Figure 4.16. Just like in the case of design with specification on phase margin, specifications on the amplitude margin may give very different results when applied to different processes. □

## Optimization

Most optimization criteria give very flat minima, which can be seen from the level curves of the loss function in the $(k, T_i)$ plane, see [Hazebroek and van der Waerden, 1950b], [Wills, 1962], [Sood and Huddleston, 1973], and [Weber and Bhalodia, 1979]. This makes the use of time domain optimization criteria difficult, since a rather high accuarcy is needed in the integrating routines to get reliable results.

Controller parameters have been computed for the standard system $G(s) = e^{-sL}/(1 + sT)$, using various optimization criteria, for details see Chapter 3. As as soon as another process is given no such results are commonly available.

The parametrization in $\omega_0$ and $\zeta_0$, which have been suggested here, could of course also be used for optimization. One benefit of this parametrization is that it would probably be easier to get good initial guesses for the optimizer, than using the standard $(k, T_i)$ parametrization. From the study of the standard system we know roughly where in the stability areas the minima of the various integral criteria are located.

## Design of PD controllers

Lack of integration in the controller makes it impossible to optimize any of the integral criteria for a load disturbance in the process input. If we have an integrating process then integral criteria can be used if the disturbance is a step in the reference signal. To be able to handle loads, the following way to choose $\omega_0$ can be used.

***The choice of*** $\omega_0$. Consider the transfer function from reference value and load disturbance to process output

$$Y(s) = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)} Y_r(s) + \frac{G(s)}{1 + G_c(s)G(s)} V(s). \qquad (4.113)$$

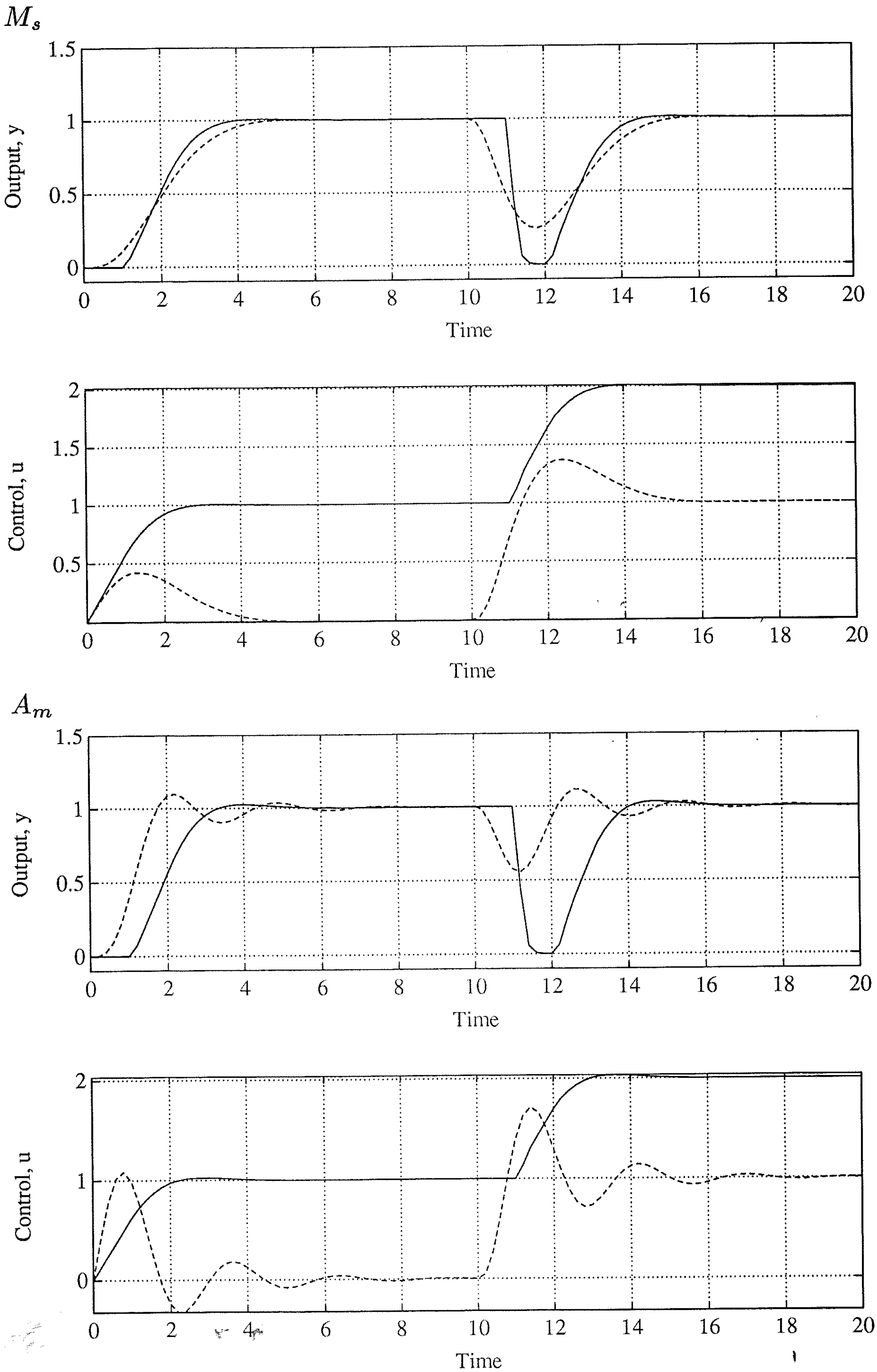



**Figure 4.16** In the figure $G_1(s)$ (solid) and $G_2(s)$ (dashed). Design with specified $M_s$ above, and with specified $A_m$ below.

If we have a PD controller $G_c(s) = k(1 + sT_d)$ and we apply steps at $y_r$ and $v$, we get from the final value theorem

$$\lim_{t\to\infty} y(t) = \frac{kG(0)}{1+kG(0)} y_r^0 + \frac{G(0)}{1+kG(0)} v^0. \tag{4.114}$$

We want $y$ to follow $y_r$ as close as possible and we want $v$ to have as little influence on $y$ as possible. Hence it seems natural to require that $k$ should be maximal. From (4.57) we see that $k$ and $k_d$ of a PD controller depend on the phase of $G(\omega_0(-\zeta_0 + i\sqrt{1-\zeta_0^2}))$ in a similar way as a PI controller does.

***The choice of*** $\zeta_0$ For a process with integration the loop transfer function of the process with a PD controller is similar in structure of that of a process without integration with a PI controller. Thus it is reasoanble to use the same principles in determining $\zeta_0$ for a PD controller as for a PI controller.

## 4.5 A design procedure

For reasons presented in this chapter the following method is recommended for design of PI and PD controllers. Choose $\omega_0$ by maximizing $k_i$ in the formula (4.33). The parameter $\zeta_0$ should be chosen by specifying a value of $M_s$ and solving Equation (4.99) with respect to $\zeta_0$. One exact and one approximate method for determining $\zeta_0$ have been suggested. The approximate method for determining $\zeta_0$ is usually sufficient. Normally $M_s = 2.0 - 2.2$ will a sensible choice, but if a system with no oscillations is desired $M_s$ should be chosen lower, $M_s = 1.5 - 1.8$. This method will be called 2PM (Two Pole Method).

A certain performace enhancement can be obtained by modifying $\omega_0$ according to the procedure described on page 69. This modification may give time responses rather like the responses obtained with controllers optimized with the standard methods, IAE, ISE, etc. The method will be called MPI (Modified PI controller).

The numerical implementation of the design methods will be discussed in Chapter 8.

## 4.6 Examples

In this section are given a number of analytical and numerical examples concerning the methods which have been presented earlier in this chapter.

## Analytical Examples

It is necessary to have a good understanding of the method in simple cases where much can be solved analytically. These examples exhibit phenomena which also occur for more complex systems.

EXAMPLE 4.20—First order system
If

$$G(s) = \frac{a}{s+b} \tag{4.115}$$

we get

$$k = \frac{-b + 2\omega_0\zeta_0}{a} \tag{4.116}$$

$$k_i = \frac{\omega_0^2}{a}. \tag{4.117}$$

We see that $k_i$ does not have a maximum. This is quite natural, since the closed system is of second order and then we can place the two poles wherever we want with a PI controller. □

EXAMPLE 4.21—PI Controller and second order system
Consider

$$G(s) = \frac{\omega_p^2}{s^2 + 2\omega_p\zeta_p s + \omega_p^2}, \tag{4.118}$$

controlled by a PI controller. We want to have the closed loop poles in $\omega_0(-\zeta_0 \pm i\sqrt{1-\zeta_0^2})$. According to (4.33) we obtain

$$k = \frac{-4\omega_0^2\zeta_0^2 + 4\zeta_p\omega_p\omega_0\zeta_0 + \omega_0^2 - \omega_p^2}{\omega_p^2} \tag{4.119}$$

$$k_i = -2\omega_0^2\frac{\omega_0\zeta_0 - \zeta_p\omega_p}{\omega_p^2}. \tag{4.120}$$

The characteristic equation is $s(s^2+2\omega_p\zeta_p s+\omega_p^2)+\omega_p^2(sk+k_i) = 0$. The closed loop system will be a third order system with the characteristic equation $(s+b)(s^2 + 2\omega_0\zeta_0 s + \omega_0^2) = 0$, and we immediately get

$$\begin{aligned} &s^3 + s^2(b + 2\omega_0\zeta_0) + s(2\omega_0\zeta_0 b + \omega_0^2) + b\omega_0^2 \equiv \\ &s^3 + s^2 2\omega_p\zeta_p + s(\omega_p^2 + k\omega_p^2) + \omega_p^2 k_i \end{aligned} \tag{4.121}$$

and

$$b = (\frac{\omega_p}{\omega_0})^2 k_i = -2(\omega_0\zeta_0 - \omega_p\zeta_p). \tag{4.122}$$

Thus we have two poles which move from the origin as $\omega_0$ increases, and one pole which moves to the right on the real axis. The locations of all the poles are

$$p_{1,2} = \omega_0(-\zeta_0 \pm i\sqrt{1-\zeta_0^2}) \tag{4.123}$$

$$p_3 = -2(\omega_p\zeta_p - \omega_0\zeta_0). \tag{4.124}$$

With this method it is impossibe to choose an $\omega_0$ greater than $\omega_p\zeta_p/\zeta_0$ without getting an unstable system.

To find the optimal $k_i$, differentiate $k_i$

$$\frac{dk_i}{d\omega_0} = -2\omega_0\frac{3\omega_0\zeta_0 - 2\omega_p\zeta_p}{\omega_p^2}, \tag{4.125}$$

and we have a maximum for $k_i$ at $\omega_0 = 2\zeta_p\omega_p/3\zeta_0$. For this value of $\omega_0$ the PI parameters will be

$$k = \frac{8\zeta_p^2\zeta_0^2 + 4\zeta_p^2 - 9\zeta_0^2}{9\zeta_0^2} \tag{4.126}$$

$$k_i = \frac{8\zeta_p^3\omega_p}{27\zeta_0^2}. \tag{4.127}$$

For the optimal value of $\omega_0$ we get the closed loop poles

$$\begin{aligned} p_{1,2} &= \frac{2\zeta_p}{3\zeta_0}\omega_p(-\zeta_0 \pm i\sqrt{1-\zeta_0^2}) = \\ &= -\frac{2}{3}\zeta_p\omega_p \pm i\frac{2}{3}\zeta_p\omega_p\sqrt{\frac{1}{\zeta_0^2}-1} \end{aligned} \tag{4.128}$$

$$p_3 = -\frac{2}{3}\zeta_p\omega_p. \tag{4.129}$$

We see that the $k_i$ optimal system in this case has the same real part of all its poles.

Another way to choose $\omega_0$ would be to specify the configuration of the three poles which are closest to the origin. We could, e.g., require that the three poles should lie in $\omega_0(-\zeta_0 \pm i\sqrt{1-\zeta_0^2})$ and $-\kappa\omega_0$, and from these assumptions determine $\omega_0$. We get the equation

$$2(\omega_p\zeta_p - \omega_0\zeta_0) = \kappa\omega_0 \Leftrightarrow \omega_0 = \frac{2\zeta_p}{\kappa + 2\zeta_0}\omega_p. \tag{4.130}$$

Thus, in this case the design parameters are $\zeta_0$ and $\kappa$. □

EXAMPLE 4.22—PI controller and second order system with real poles
Consider the process

$$G(s) = \frac{a}{(s+1)(s+a)}. \tag{4.131}$$

From (4.32) and (4.33) we get

$$k = -\frac{(a - 2\omega_0\zeta_0(1+a) + \omega_0^2(4\zeta_0^2 - 1)}{a} \tag{4.132}$$

$$k_i = -\omega_0^2 \frac{2\omega_0\zeta_0 - a - 1}{a}. \tag{4.133}$$

The parameter $k_i$ has maximum for $\omega_0 = (a+1)/3\zeta_0$, and the characteristic equation has the roots

$$p_{1,2} = -\frac{a+1}{3} \pm i\frac{(a+1)\sqrt{1-\zeta_0^2}}{3\zeta_0} \tag{4.134}$$

$$p_3 = -\frac{a+1}{3}. \tag{4.135}$$

The real parts of all the three roots of the characteristic equation are equal.
□

EXAMPLE 4.23—PI controller and first order system with integrator
Consider the process

$$G(s) = \frac{a}{s(s+b)}. \tag{4.136}$$

From (4.32) and (4.33) we get

$$k = -\omega_0 \frac{4\omega_0\zeta_0^2 - 2\zeta_0 b - \omega_0}{a} \tag{4.137}$$

$$k_i = -\omega_0^2 \frac{2\omega_0\zeta_0 - b}{a}. \tag{4.138}$$

The parameter $k_i$ has maximum for $\omega_0 = b/3\zeta_0$, and the characteristic equation has the roots

$$p_{1,2} = -\frac{b}{3} \pm i\frac{b}{3}\frac{\sqrt{1-\zeta_0^2}}{\zeta_0} \tag{4.139}$$

$$p_3 = -\frac{b}{3}. \tag{4.140}$$

The real parts of all the three roots of the characteristic equation are equal.
□

EXAMPLE 4.24—PD control for a third order system
Consider the process

$$G(s) = \frac{\omega_p^3 \alpha_p}{(s + \alpha_p \omega_p)(s^2 + 2\omega_p \zeta_p s + \omega_p^2)}. \tag{4.141}$$

If we specify a PD controller to give closed loop poles in $p_{1,2} = \omega_0(-\zeta_0 \pm i\sqrt{1-\zeta_0^2})$ and choose $\omega_0$ to maximize $k_i$, we get $\omega_0 = \omega_p(\alpha_p + 2\zeta_p)/3\zeta_0$. This results in closed loop poles in

$$p_{1,2} = \frac{\omega_p(\alpha_p + 2\zeta_p)}{3}(-1 \pm \frac{i\sqrt{1-\zeta_0^2}}{\zeta_0}) \tag{4.142}$$

$$p_3 = -\frac{\omega_p(\alpha_p + 2\zeta_p)}{3}. \tag{4.143}$$

Note that the real parts of the closed loop poles are equal, like in the PI case for a second order system. □

### Numerical examples

EXAMPLE 4.25—The relation between $\omega_0$ and $\omega_b$
PI controllers were designed for the system

$$G(s) = \frac{e^{-sL}}{1 + sT} \tag{4.144}$$

with 2PM for $M_s = 2.0 \ldots 2.5$. The parameter $T = 1$ and $L$ varies between 0.1 and 1. In the design we compute the parameter $\omega_0$. The bandwidth was computed with (4.88), and the ratio $\omega_0/\omega_b$ can be found in Figure 4.17.

The parameter $\omega_0$ varies between 1.1 and 8 in the different designs, but the $\omega_0/\omega_b$ remain approximately the same. Hence it is reasonable to take $\omega_0$ as an equally good measure of the 'bandwidth' of the system as the bandwidth computed with (4.88). This is good to know since the parameter $\omega_0$ is obtained as a by-product in all designs. It also feels natural that the distance from the origin to the dominant poles is a measure of the bandwidth of the system. □

## 4.7 Summary

In this chapter the Dominant Pole Design principle has been investigated for PI and PD controller design. Methods for the design of PI and PD controllers

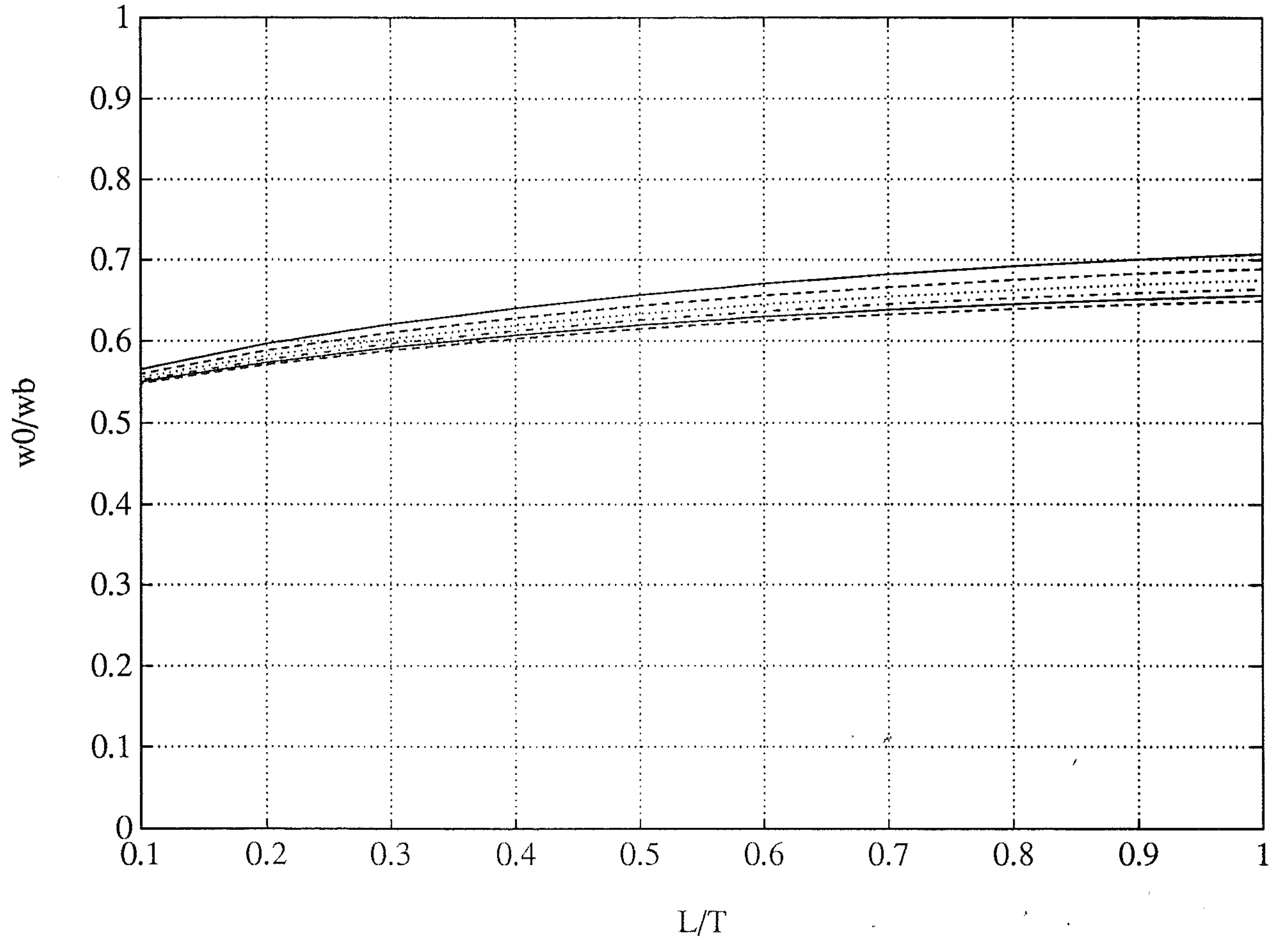



**Figure 4.17** The $\omega_0/\omega_b$ for the closed loop system where $G(s) = e^{-sL}/(1+sT)$ is controlled by a PI controller designed with 2PM.

has been derived and discussed from different points of view. Design procedures have been recommended. Examples of the use of them and comparisons with other methods will be made in Chapter 7.

Only a few processes can be analyzed completely, due to the complicated expressions of the controller parameters. A couple of simple processes have been analyzed analytically.

# 5

# PID control

In this chapter we will present and discuss methods for PID controller tuning, based on the same principles as for PI controller tuning in Chapter 4. Two methods have been tried. First, direct specification of three poles. This method is a direct analogy to the methods discussed in Chapter 4. The second method is based on modification of a well tuned PI or PD controller.

PID control may be needed for two reasons. First, to be able to control the process at all. Certain processes cannot be controlled by PI control, a derivative term is needed to stabilize the system. The second reason for PID control is to get better performance from the controlled system.

## 5.1 PID control

Two methods for designing PID controllers based on the dominant pole design principle will now be considered. The first method uses the fact that a PID controller has three parameters and can thus be specified by placement of three closed loop poles, two complex and one on the negative real axis. This is not a completely general method because there are closed loop systems under PID control which do not have a pole on the negative real axis. Another way to design a PID controller is to start with a well tuned PI controller and add derivative action until some design criterion is fulfilled. Modifying a PI controller is the most natural, because for most plants integral action is always wanted to handle load disturbances.

**Specification of three poles**

Let a PID controller be parametrized as

$$G_c(s) = k + \frac{k_i}{s} + k_d s. \tag{5.1}$$

This form will be used for simplicity. Modifications in the set point weighting and limitation of the derivative gain will be taken care of later. Three poles of the closed loop system must be specified to find the parameters $k$, $k_i$, and $k_d$. We choose them as

$$p_{1,2} = \omega_0(-\zeta_0 \pm i\sqrt{1-\zeta_0^2}) \tag{5.2}$$

$$p_3 = -\alpha_0\omega_0. \tag{5.3}$$

Introduce $G(p_{1,2}) = A \pm iB$, and $G(p_3) = C$. The condition that

$$1 + G(s)G_c(s) = 0 \tag{5.4}$$

should have zeros in $p_1, p_2$, and $p_3$ gives,

$$k = -\frac{\sqrt{1-\zeta_0^2}(-2\alpha_0\zeta_0(A^2+B^2) + (1+\alpha_0^2)AC) + (\alpha_0^2-1)\zeta_0 BC}{(1-2\alpha_0\zeta_0+\alpha_0^2)\sqrt{1-\zeta_0^2}(A^2+B^2)C} \tag{5.5}$$

$$k_i = -\alpha_0\omega_0\frac{(\alpha_0-\zeta_0)BC + \sqrt{1-\zeta_0^2}(AC-A^2-B^2)}{(1-2\alpha_0\zeta_0+\alpha_0^2)\sqrt{1-\zeta_0^2}(A^2+B^2)C} \tag{5.6}$$

$$k_d = -\frac{(\alpha_0\zeta_0-1)BC + \alpha_0\sqrt{1-\zeta_0^2}(AC-A^2-B^2)}{\omega_0(1-2\alpha_0\zeta_0+\alpha_0^2)\sqrt{1-\zeta_0^2}(A^2+B^2)C}. \tag{5.7}$$

Define $a(\omega_0)$ and $\phi(\omega_0)$ by $G(\omega_0 e^{i(\pi-\gamma)}) = a(\omega_0)e^{i\phi(\omega_0)}$, where $\gamma$ is regarded as a constant. For the poles $p_{1,2}$ we get

$$p_{1,2} = \omega_0(-\zeta_0 \pm i\sqrt{1-\zeta_0^2}) = \omega_0 e^{i(\pi M_p \gamma)}, \tag{5.8}$$

hence $\cos\gamma = \zeta_0$, and for $p_3 = -\alpha_0\omega_0$ we get $\gamma = 0$. With this parametrization we get

$$A = a(\omega_0)\cos\phi(\omega_0) \tag{5.9}$$

$$B = a(\omega_0)\sin\phi(\omega_0) \tag{5.10}$$

$$C = -b(\alpha_0\omega_0), \tag{5.11}$$

where $b(\omega_0)$ is the magnitude function for $\gamma = 0$. Insert these expressions for $A$, $B$, and $C$ in (5.5), (5.6), and (5.7) and we get

$$k = -\frac{\alpha_0^2 b(\omega_0)\sin(\gamma+\phi) + b(\omega_0)\sin(\gamma-\phi) + \alpha_0 a(\omega_0)\sin 2\gamma}{a(\omega_0)b(\omega_0)(\alpha_0^2 - 2\alpha_0\cos\gamma + 1)\sin\gamma} \quad (5.12)$$

$$k_i = -\alpha_0\omega_0\frac{a(\omega_0)\sin\gamma + b(\omega_0)(\sin(\gamma-\phi) + \alpha_0\sin\phi)}{a(\omega_0)b(\omega_0)(\alpha_0^2 - 2\alpha_0\cos\gamma + 1)\sin\gamma} \quad (5.13)$$

$$k_d = -\frac{\alpha_0 a(\omega_0)\sin\gamma + b(\omega_0)(\alpha_0\sin(\gamma+\phi) - \sin\phi)}{\omega_0 a(\omega_0)b(\omega_0)(\alpha_0^2 - 2\alpha_0\cos\gamma + 1)\sin\gamma}. \quad (5.14)$$

As can be seen from these formulas there is no simple connection between the phase function $\phi(\omega_0)$ and $k_i$ as in the PI controller case.

There are a few problems with this approach. This method forces us to specify a pole on the negative real axis. If we try to specify more than two poles of the closed loop system and let the others be free, the locations of the free poles can be very sensitive to where the fixed poles have been located. This effect is most pronounced for resonant systems.

### PID controller based on PI controller

Suppose the controller

$$G_c(s) = k + \frac{k_i}{s} + k_d s, \quad (5.15)$$

is used and it is desired to have two closed loop poles in $p_{1,2} = \omega_0(-\zeta_0 \pm i\sqrt{1-\zeta_0^2})$. Solving

$$1 + G(s)G_c(s) = 0 \quad (5.16)$$

gives the controller parameters

$$k = 2k_d\omega_0\zeta_0 - \frac{\zeta_0 B + \sqrt{1-\zeta_0^2}A}{\sqrt{1-\zeta_0^2}(A^2+B^2)} = 2k_d\omega_0\zeta_0 + k_{\mathrm{PI}} \quad (5.17)$$

$$k_i = k_d\omega_0^2 - \frac{\omega_0 B}{\sqrt{1-\zeta_0^2}(A^2+B^2)} = k_d\omega_0^2 + k_{i,\mathrm{PI}}, \quad (5.18)$$

where $k_{\mathrm{PI}}$ and $k_{i,\mathrm{PI}}$ are the controller parameters for a pure PI controller, according to (4.32) and (4.33). With this parametrization the PID controller can be written

$$G_c(s) = G_{c,\mathrm{PI}}(s) + k_d(s^2 + 2\omega_0\zeta_0 s + \omega_0^2)\frac{1}{s}, \quad (5.19)$$

where $G_{c,\mathrm{PI}}(s)$ is a pure PI controller, i.e,. $G_c(s)$ with $k_d = 0$. The task is now to choose $\omega_0$, $\zeta_0$, and $k_d$ such that the system behaves well.

The characteristic equation of the closed loop system becomes

$$1 + G(s)G_c(s) = 1 + G(s)G_{c,\mathrm{PI}}(s) + G(s)k_d(s^2 + 2\omega_0\zeta_0 s + \omega_0^2)\frac{1}{s}. \quad (5.20)$$

For a system controlled by a PI controller we have

$$1 + G(s)G_{c,\mathrm{PI}}(s) = (s^2 + 2\omega_0\zeta_0 s + \omega_0^2)R(s). \quad (5.21)$$

The zeros of $R(s)$ are the free poles of the system controlled by the PI controller $G_{c,\mathrm{PI}}(s)$. Thus

$$1 + G(s)G_c(s) = (s^2 + 2\omega_0\zeta_0 s + \omega_0^2)(R(s) + G(s)k_d\frac{1}{s}). \quad (5.22)$$

The root locus of $1 + G(s)G_c(s)$ with respect to $k_d$ will start in the zeros of $R(s)$ and end in the zeros of $G(s)$ or in infinity.

This parametrization 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 $\omega_0$ may have to be modified, e.g., in such a way that IE is maximized.

## 5.2 Design choices

In this section the design choices of the PID controller design methods presented in the previous section will be discussed. Both the method of specifying three poles and the method of modifying a PI controller will be considered.

### PID controller based on specification of three poles

Of the three parameters $\omega_0$, $\zeta_0$, and $\alpha_0$, $\alpha_0$ is the parameter which has the smallest influence on the system, at least for systems normally controlled by PID controllers. The choice of $\alpha_0$ will therefore be discussed first. If $\alpha_0$ is chosen sensibly we than have a design problem which can be handled like the design of a PI controller.

***The choice of*** $\alpha_0$ The parameter $\alpha_0$ specifies the relative location of the pole on the negative real axis to the two dominant complex poles. A very simplified model of the control system is now that we have a closed loop system consisting of three poles. The question is now, how does the pole on the negative real axis influence the behaviour of the closed loop system. This is illustrated by an example.

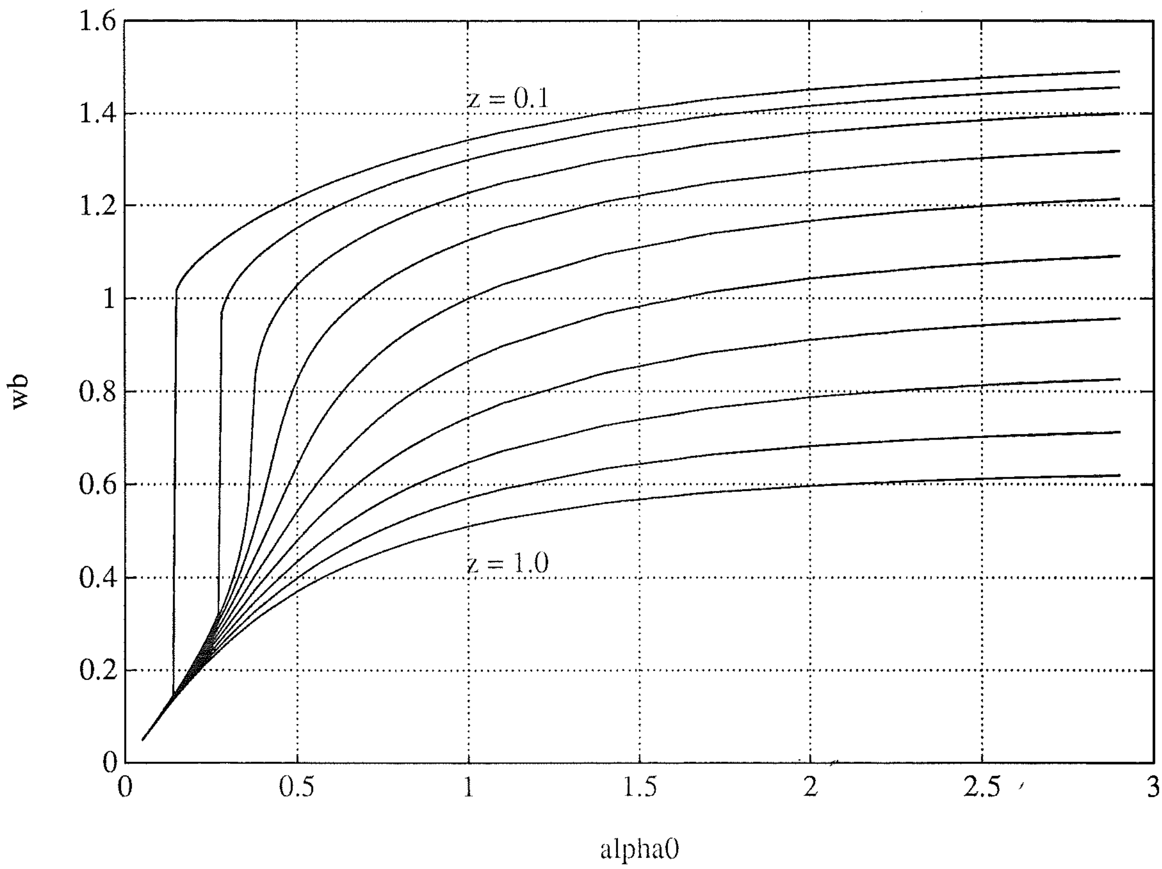



**Figure 5.1** Bandwidth of (5.23) as function of $\alpha_0$ for different $\zeta$.

EXAMPLE 5.1—Bandwidth of third order system
Consider the process

$$G(s) = \frac{\alpha_0 \omega^3}{(s^2 + 2\zeta\omega s + \omega^2)(s + \alpha_0\omega)}. \tag{5.23}$$

When $\alpha_0$ is increased the bandwidth of the system is also increased, but the system will eventually be dominated by the two complex poles at $\omega_0(-\zeta_0 \pm i\sqrt{1-\zeta_0^2})$. Figure 5.1 shows the bandwidth of system (5.23) as function of $\alpha_0$ for different values of $\zeta$. In this example we have $\omega = 1$. As can be seen from Figure 5.1 we do not gain very much bandwidth by increasing $\alpha_0$ over 1. High bandwidth for small $\zeta$ does not imply a well behaved system. High bandwidth means only that the system reacts fast in some sense. □

A reasonable $\alpha_0$ can also be obtained from an optimization criterion, as is shown in Example 5.2.

EXAMPLE 5.2—Determining $\alpha_0$ by optimization
Consider the system

$$G(s) = \frac{e^{-sL}}{1+s}. \tag{5.24}$$

PID controllers were designed by choosing $\omega_0$ to maximize $k_i$. The parameter $\zeta_0$ was chosen in two different ways. In the first case $\zeta_0$ was set constant

$\zeta_0 = 0.7$ and in the second chosen such that $M_s = 2$. The parameter $\alpha_0$ was varied. The loss function IAE was computed for a step disturbance on the process input. In Figure 5.2 the criterion IAE is plotted against $\alpha_0$. As can be seen the loss function has a minimum for $\alpha_0 \approx 1$, but the minimum is very flat. In Figure 5.3 we see that for processes designed with a specified $M_s$ value we get a minimum for lower values of $\alpha_0$.

Systems with resonances may behave differently. Consider

$$G(s) = \frac{1}{(s^2 + 2\zeta_p s + 1)(s+1)}. \tag{5.25}$$

PID controllers were designed for different $\alpha_0$ so that $M_s = 2.5$, and the loss function IAE was computed, which can be seen in Figure 5.4. The minima in this case are obtained for lower values of $\alpha_0$. This must be considered if we want to design PID controllers for resonant systems with this method. □

Another problem is if a too large $\alpha_0$ is chosen. If $\omega_0$ is chosen by maximizing $k_i$ then we may get another pole on the real axis to the right of $-\alpha_0\omega_0$, which clearly is not what we intended.

EXAMPLE 5.3—The effect of too large $\alpha_0$
Consider the system

$$G(s) = \frac{1}{(s+1)^n}, \tag{5.26}$$

where $n = 3 \ldots 10$.

Controllers were designed by letting $\zeta_0$ be constant $\zeta_0 = 0.5$ and by choosing $\zeta_0$ such that $M_s = 2$ and maximizing $k_i$ with respect to $\omega_0$ for different $\alpha_0$. The closed loop poles on the negative real axis are shown in Figure 5.5. The poles are normalized with $\omega_0$. The straight line corresponds to the prescribed pole at $-\alpha_0\omega_0$. Systems with large $n$ can tolerate a larger value of $\alpha_0$ without getting a pole to the right of $-\alpha_0\omega_0$. A reasonable $\alpha_0$ gives no extra pole to the right of $-\alpha_0\omega_0$. This example suggests that $\alpha_0$ should be chosen in the range 0.5 to 1.3. □

***The choice of*** $\omega_0$ ***and*** $\zeta_0$ Once $\alpha_0$ has been specified we have a controller design problem of the same complexity as the design of a PI controller. The same methods as in Chapter 4 can be applied.

## Modified PI and PD controllers

In most cases we want to gain performance by using PID control instead of PI control. A natural approach is then to start with a PI controller and to investigate the gains in performance that can be obtained by adding derivative action. This can be done by keeping $\zeta_0$ from the PI controller in the PID case, and modify $\omega_0$ and $k_d$ to gain performance. This may lead to a diminished robustness of the PI controller.

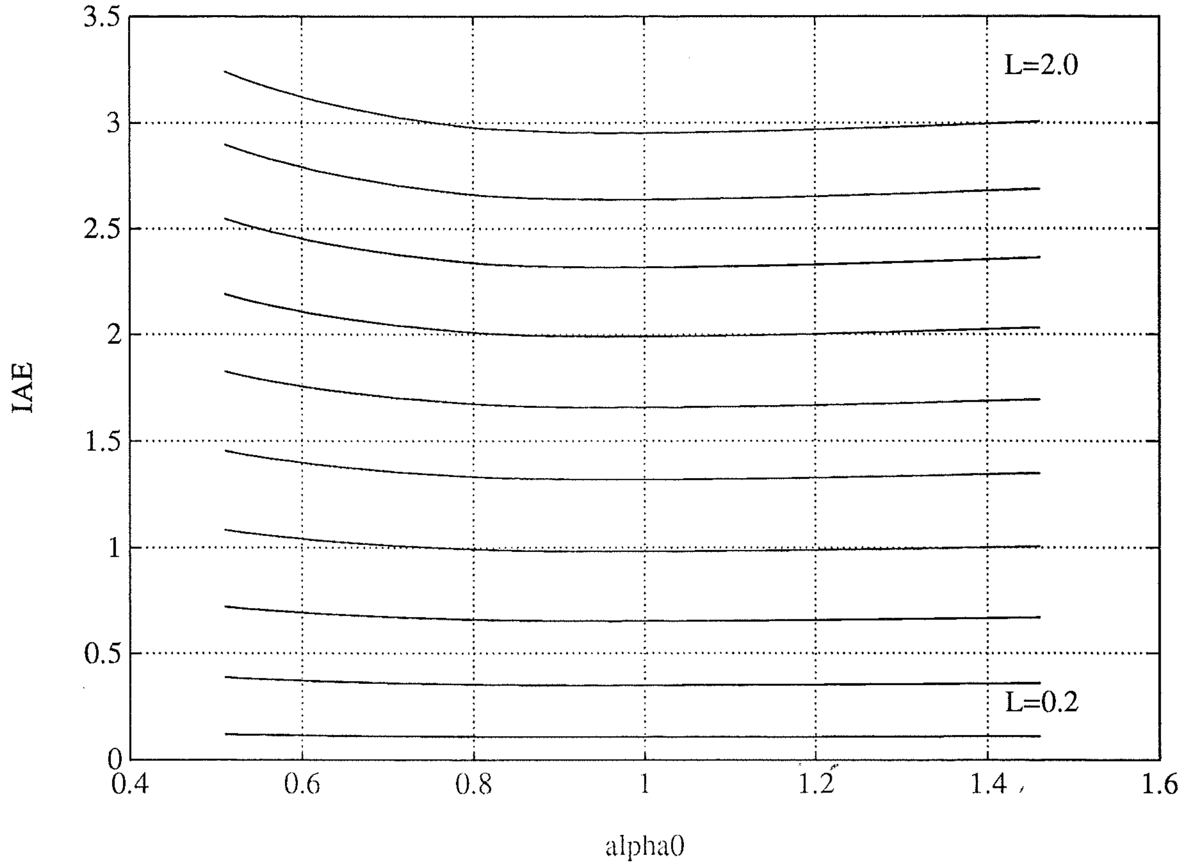



**Figure 5.2** Loss function IAE as function of $\alpha_0$ when (5.24) with $L = 0.2 \ldots 2.0$ is controlled by a PID controller with $\alpha_0$ as design parameter. The system is designed with constant $\zeta_0 = 0.7$.

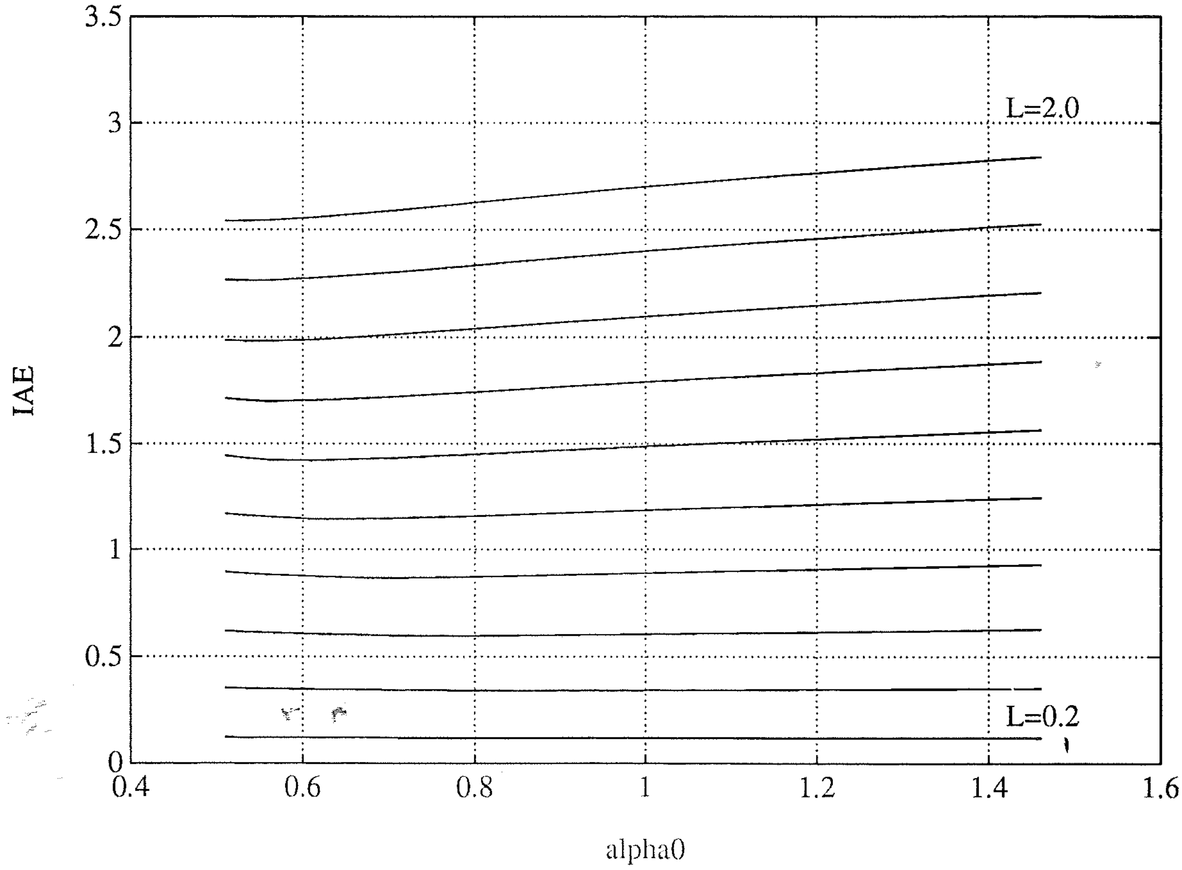



**Figure 5.3** Loss function IAE as function of $\alpha_0$ when (5.24) with $L = 0.2 \ldots 2.0$ is controlled by a PID controller designed with $M_s = 2.0$.

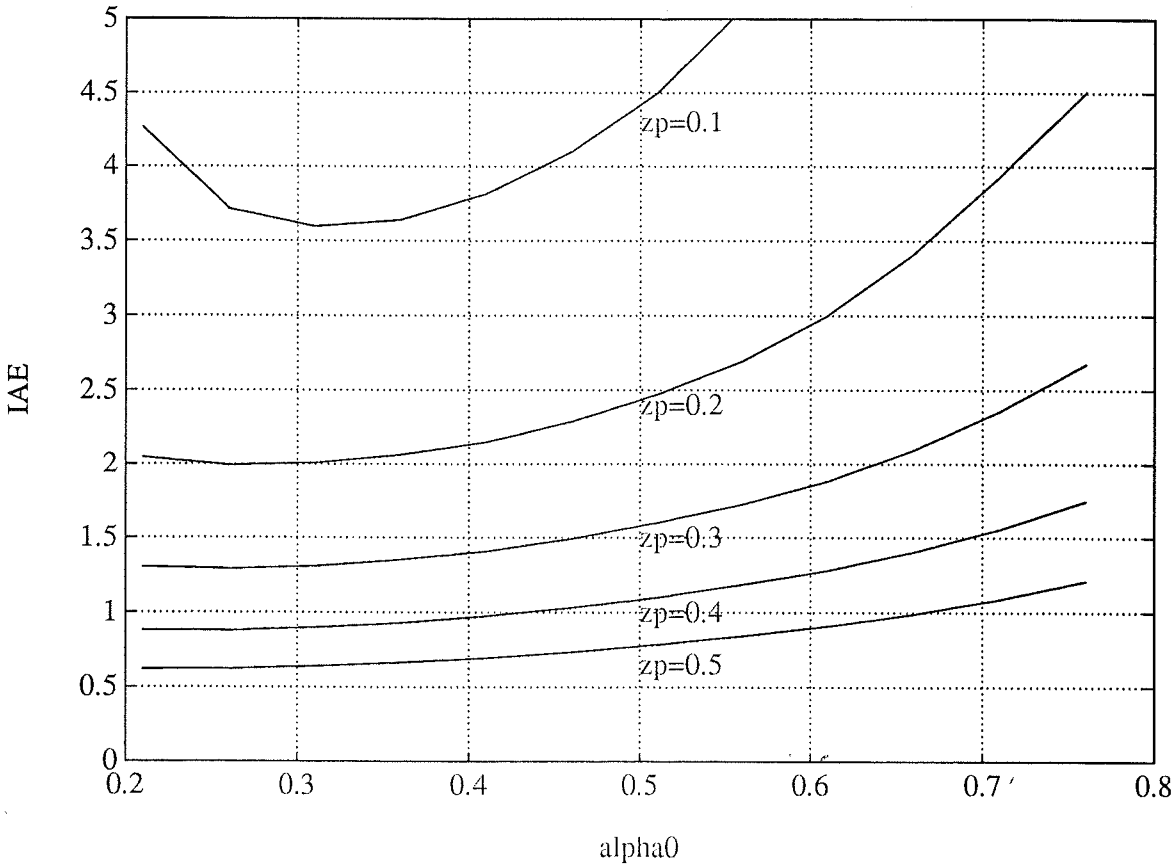



**Figure 5.4** Loss function IAE as function of $\alpha_0$ when (5.25) with $\zeta_p = 0.1, 0.2, 0.3, 0.4,$ and $0.5$ is controlled by a PID controller with $\alpha_0$ as design parameter. The parameter $\zeta_0$ has been chosen to make $M_s = 2.5$.

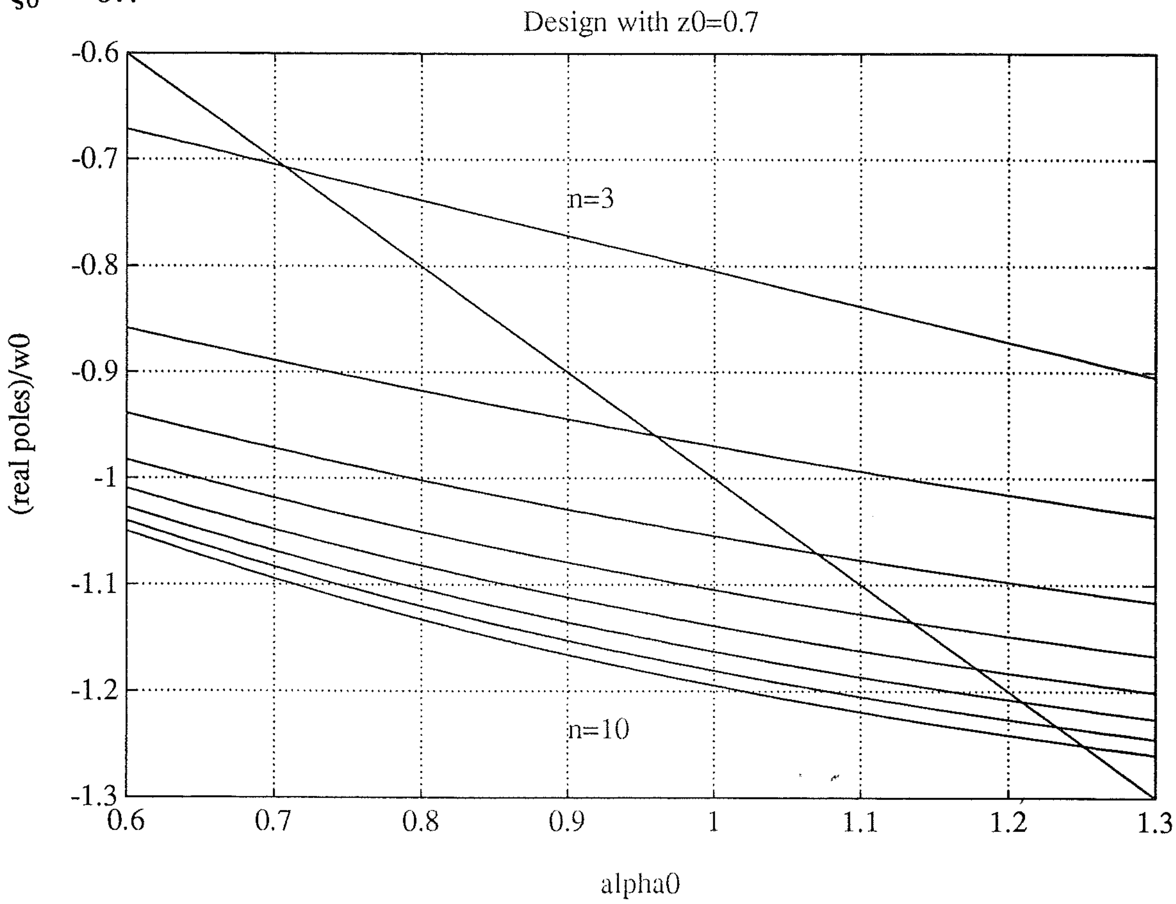
Systems with poorly damped poles are difficult or impossible to control with a PI controller, but can sometimes be stabilized with a PD controller. In many cases this can be interpreted as a state feedback from some kind of velocity. The $\zeta_0$ of a well tuned PD controller can in the same manner as the $\zeta_0$ of a PI controller be used in a PID controller. In the following will be presented a method for PID controller design based on the parameters $\omega_0$, $\zeta_0$, and $k_d$.

***The choice of*** $\zeta_0$ To find a appropriate $\zeta_0$, make well tuned PI or PD controller with a suitable $M_s$. The design procedure will give a $\zeta_0$which then will be used in the PID controller. The parameter $\zeta_0$ has the great impact on the robustness of the system. The $M_s$ value used in the design of the PI or PD controller will be called $M_{s1}$.

***The choice of*** $\omega_0$ Just as for PI we have IE $= 1/k_i$. Therefore $\omega_0$ will be chosen to maximize $k_i$.

***The choice of*** $k_d$ A second design parameter is needed to determine $k_d$. Let $M_{s2}$ be the desired $M_s$ value of the system controlled by the PID controller.

$\zeta_0 = 0.7$




$M_s = 2.0$

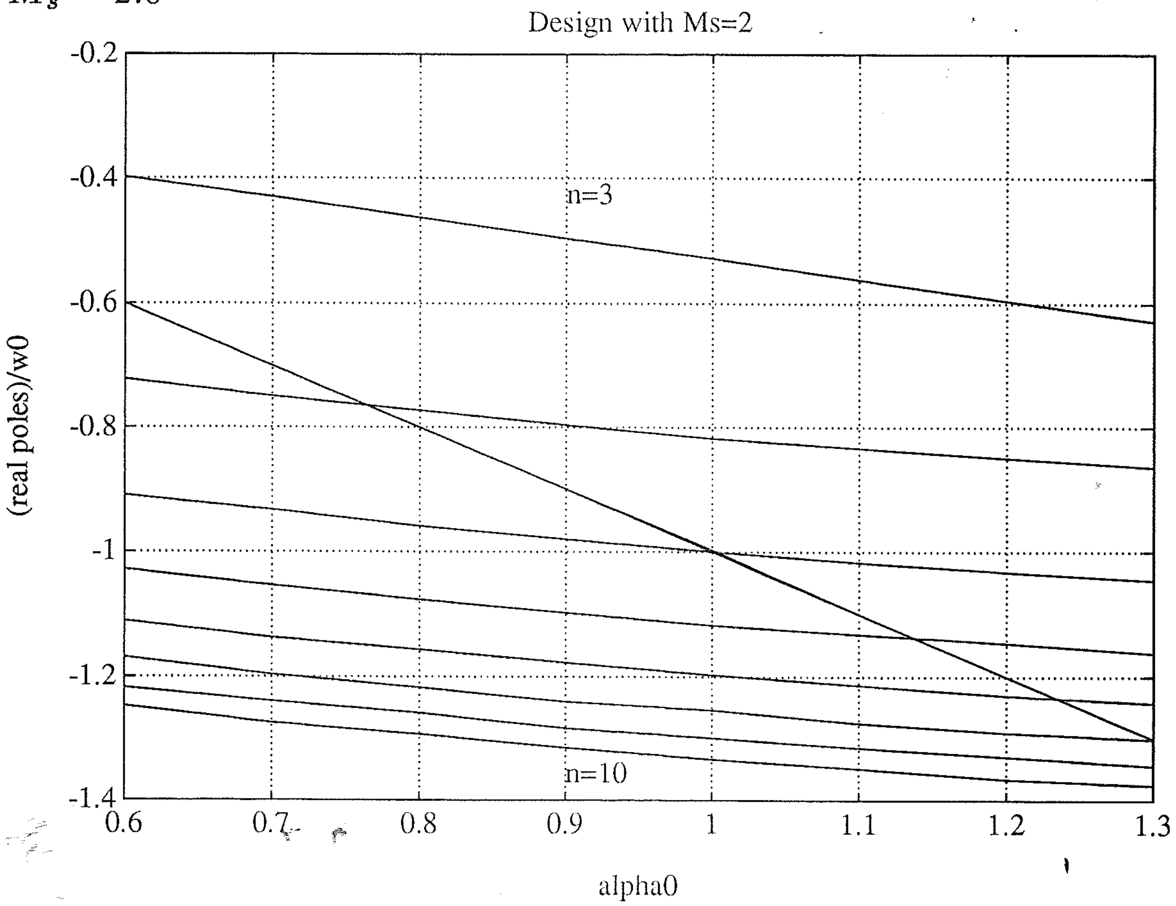



**Figure 5.5** The closed system's poles on the negative real axis, normalized by $\omega_0$. In the figure are shown design with constant $\zeta_0 = 0.7$, and design with $M_s = 2.0$. The straight line corresponds to the prescribed pole at $-\alpha_0\omega_0$.

The design algorithm is then as follows. Design a PI controller (or PD controller) such that the system controlled by this controller gets $M_s = M_{s1}$. Use the value of $\zeta_0$ from this controller to design the modified controller. Keep $\zeta_0$ fixed and increase $k_d$, while $\omega_0$ is adjusted to keep $k_i$ maximal until $M_s = M_{s2}$. This requires that a reasonable PI or PD controller can be found.

Unfortunately, any pair of $M_{s1}$ and $M_{s2}$ cannot be achieved if $\omega_0$ is chosen to maximize $k_i$. Possible pairs of $M_{s1}$ and $M_{s2}$ are strongly system dependent. Suppose that $M_{s1}$ is given. For systems without resonances the smallest possible $M_{s1}$ normally corresponds to $\zeta_0 = 1$. As $k_d$ is increased we see from (5.18) that if $k_i$ has a local maximum when $k_d = 0$ this maximum may disappear for a sufficiently large $k_d$, and an inflexion point appears. An example where this does not happen is for a third order system with two resonant poles and one real pole. As $k_d$ is increased the $\omega_0$ corresponding to a maximal $k_i$ is also increased. The maximal value of $M_{s2}$ is obtained for the $k_d$ which give an inflexion point in $k_i$. These claims have been made after the examination of a number of different systems. They may not be fulfilled for systems of low degree or for resonant systems. Example 5.4 shows the possible regions in the $(M_{s1}, M_{s2})$ plane.

EXAMPLE 5.4—The $M_{s1}$–$M_{s2}$ relation
Consider the systems

$$G_1(s) = \frac{e^{-s}}{1+sT}, \qquad T = 1,\ 2 \tag{5.27}$$

$$G_2(s) = \frac{1}{(s+1)^n}, \qquad n = 4,\ 8. \tag{5.28}$$

Possible values of $M_{s1}$ and $M_{s2}$ are shown in Figure 5.6. As can be seen from Figure 5.6 the parameter $M_{s2}$ must be chosen close to $M_{s1}$ for small $M_{s1}$. As $M_{s1}$ increases so does $M_{s2}$, but much more rapidly. When we make a cautious design, with a low value of $M_{s1}$, it will not be possible to use much derivative action to increase the $M_s$ value of the controlled system. If we want a larger $M_s$ of the final system, this should be accomplished by redesigning the PI controller with a higher value of $M_{s1}$. □

The function $M_{s2} = M_s(k_d)$ is interesting, because it tells us what happens when we increase $k_d$. To compute $M_s(k_d)$ for a given $M_{s1}$ and $k_d$ we optimize $k_i$ with respect to $\omega_0$ and use this controller to compute $M_{s2}$. Again, it is impossible to give a general statement that applies to all transfer functions. The typical curve is almost flat or decreasing for low $k_d$ and rapidly increasing for larger $k_d$, while $\omega_0$ increases all the time. A typical $M_s(k_d)$ function is described in Example 5.5.

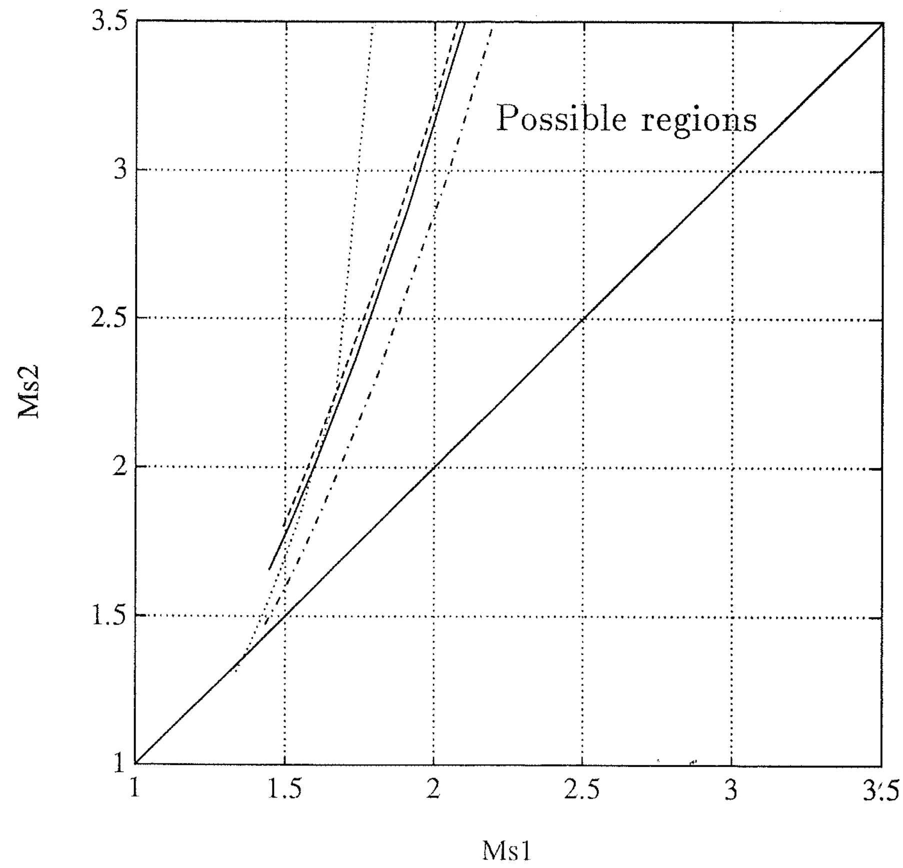



**Figure 5.6** Possible regions for $M_{s1}$ and $M_{s2}$. In the figure $G_1(s), T = 1$ (solid line), $G_1(s), T = 2$ (dashed line), $G_2(s), n = 4$ (dotted line), and $G_2(s), n = 8$ (dashed-dotted line).

EXAMPLE 5.5—The $M_s(k_d)$ function

Consider

$$G(s) = \frac{\alpha_p \omega_p^3}{(s + \alpha_p \omega_p)(s^2 + 2\zeta_p \omega_p s + \omega_p^2)} \tag{5.29}$$

with $\omega_p = 1$ and $\alpha_p = 1$. We design a PI controller with $M_s = 2.0$ and add derivative action gradually. As $k_d$ is increased $k_i$ is optimized with respect to $\omega_0$. The $M_s$ value is then computed for the different controlled systems. The functions are shown in Figure 5.7, where $\zeta_p$ has assumed values between 0.5 and 1. The derivative part decreases the $M_s$ value more for systems with low damping than for systems with high damping. If we want a system with a specified $M_s$ we may increase the derivative part (and $\omega_0$) more for systems with large damping than with small damping. From Figure 5.8 we see that $\omega_0$ also increases when $k_d$ is increased.

The root locus for the closed loop system with respect to $k_d$ is shown in Figure 5.9 for $\zeta_p = 1$. The behaviour of the closed loop poles shown in this figure is typical for this design procedure. When $k_d$ is increased, with $\omega_0$ adjusted to optimize $k_i$, two real poles of the system become complex and go towards the imaginary axis. This means that $k_i$ will eventually loose its

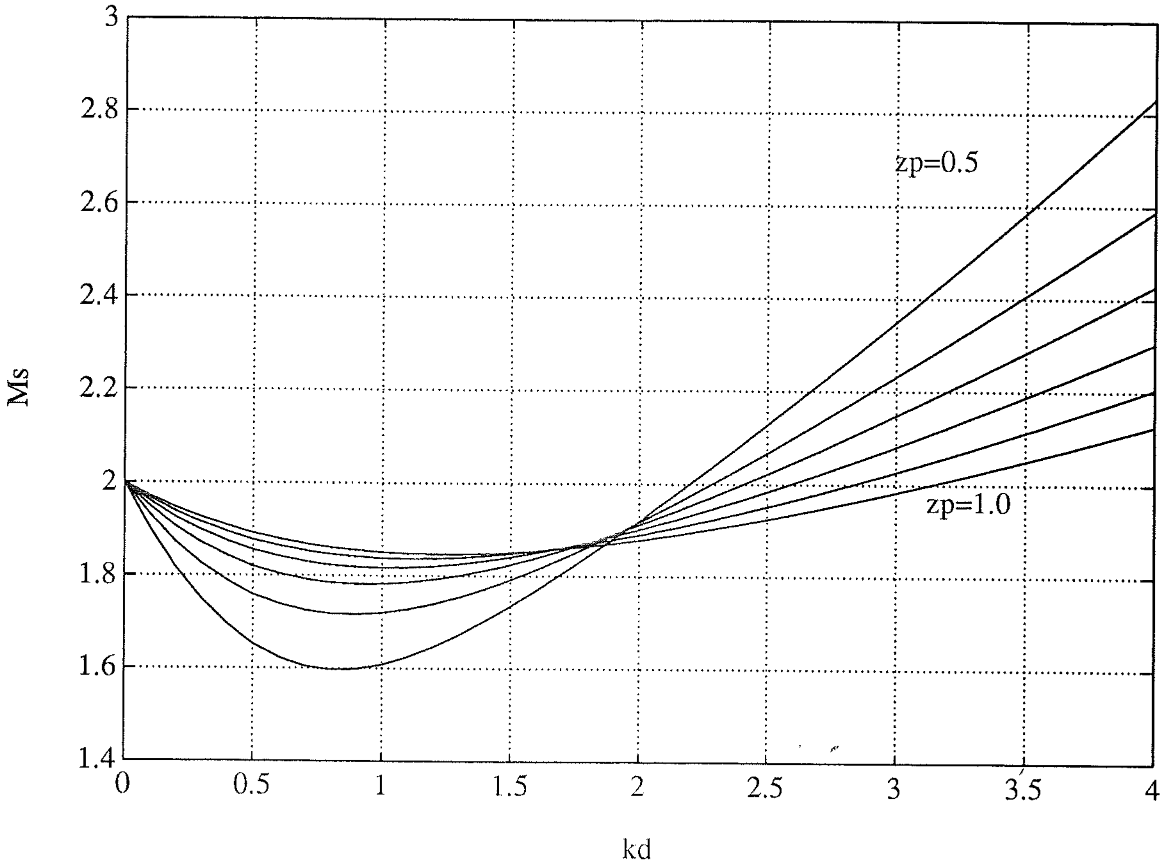



**Figure 5.7** $M_s$ as function of $k_d$.

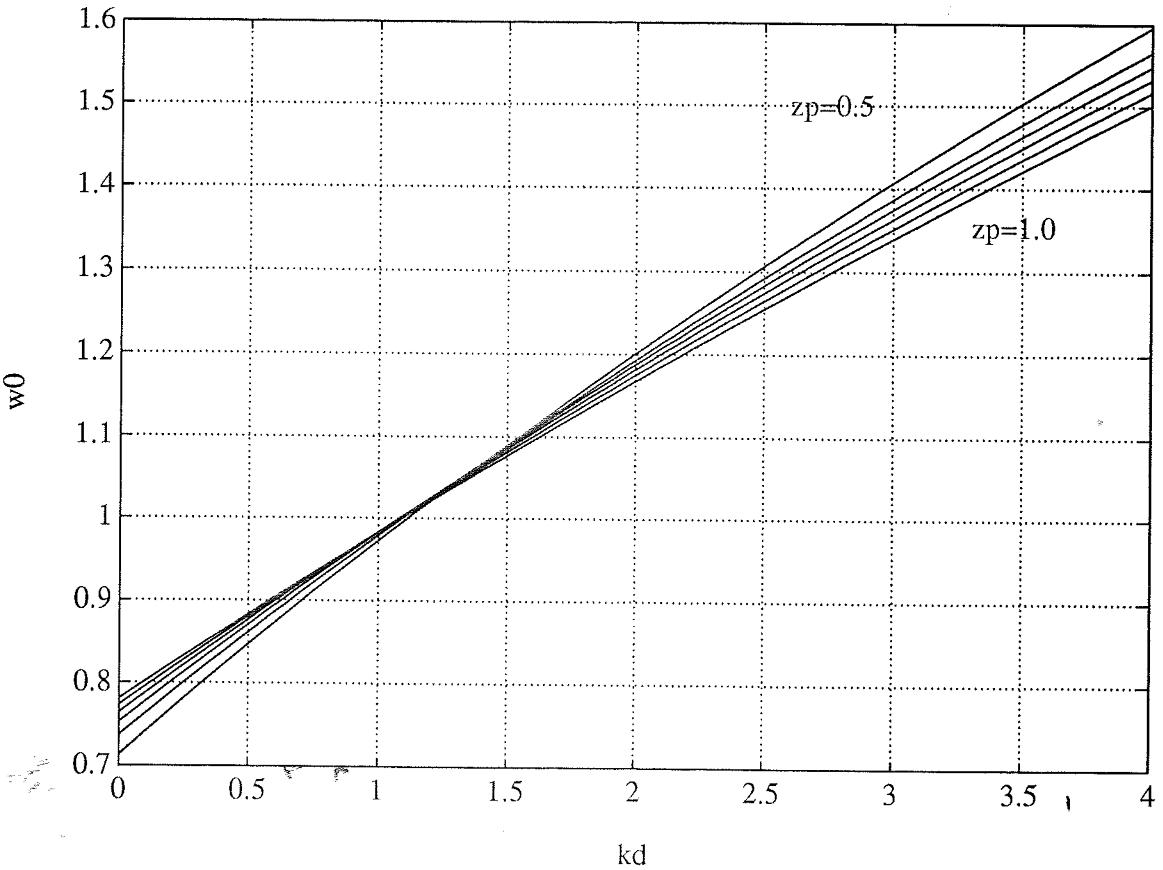



**Figure 5.8** $\omega_0$ as function of $k_d$.

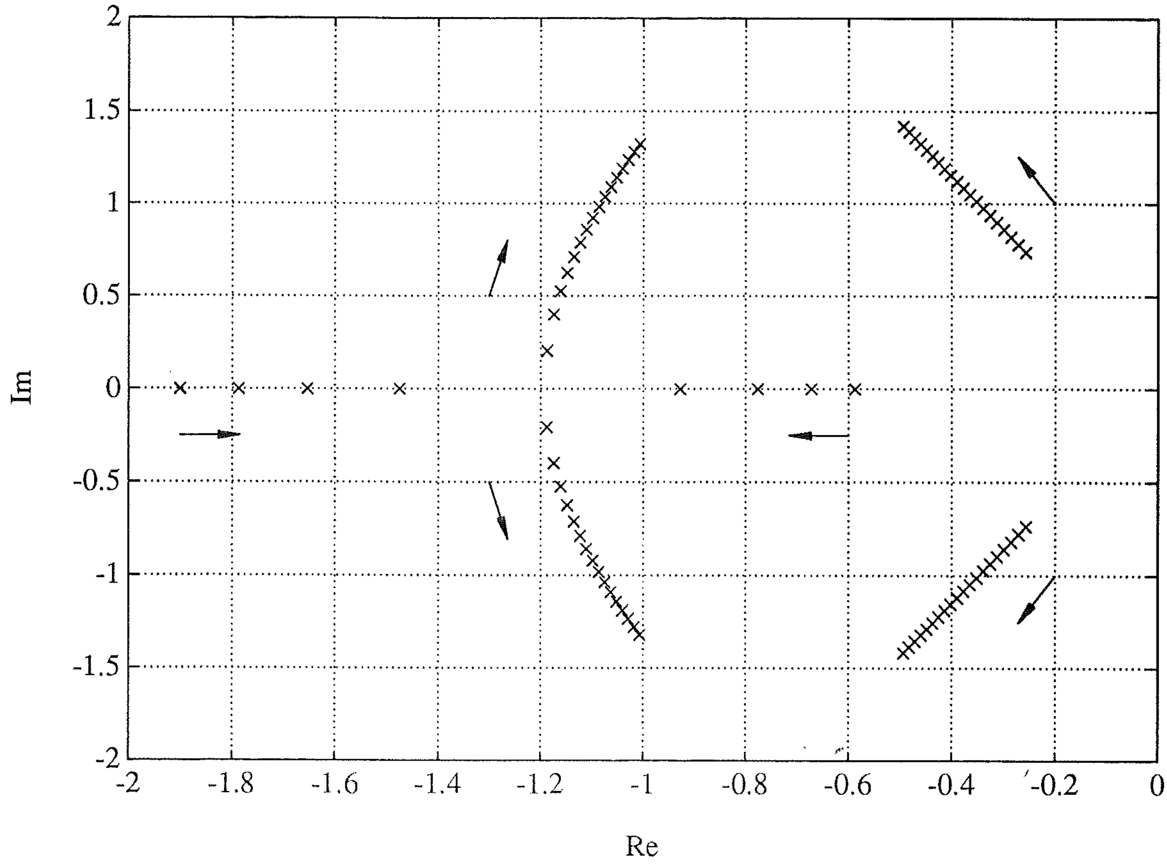



**Figure 5.9** Root locus for the closed loop system with respect to $k_d$, with $0 < k_d < 4$.

maximum. □

The value of $k_d$ where $k_{i,\mathrm{PID}}$ ceases to have a maximum is possible to estimate. Differentiate $k_{i,\mathrm{PID}} = k_{i,\mathrm{PI}} + \omega_0^2 k_d$ with respect to $\omega_0$, set the derivative to zero and we get

$$\frac{dk_{i,\mathrm{PI}}(\omega_0)}{d\omega_0} + 2\omega_0 k_d = 0. \tag{5.30}$$

The maximal $k_d$ which solves (5.30) equation is

$$k_d = \max_{\omega_0} -\frac{dk_{i,\mathrm{PI}}(\omega_0)}{d\omega_0}\frac{1}{2\omega_0}. \tag{5.31}$$

This maximum does not necessarily exist for systems with low order dynamics. This assessment of the maximal $k_d$ is useful in the numerical computations when we need to compute the $k_d$ which gives a certain $M_{s2}$.

### Modified PD controller

The parameter $\zeta_0$ is the only parameter from the design of a PD controller which will be used in the design of a PID controller. The design is completely analogous to the design of a PID controller from a PI controller.

## 5.3 The derivative filter

To avoid getting infinite gain for high frequencies, a low pass filter is usually introduced in the derivative part of a PID controller. Most implementors have chosen to use a first order filter, but higher order filters can also be found. This is to diminish the high frequency gain further. The controller's transfer function then becomes

$$G_c(s) = k(1 + \frac{1}{sT_i} + \frac{sT_d}{1 + \frac{sT_d}{N}}). \tag{5.32}$$

The filter constant $N$ is a design parameter which should be chosen. The quantity $k(1+N)$ is the high frequency gain of the controller. Small values of $N$ will affect the dynamics of the system considerably but not amplify the noise. To obtain a good derivative action much phase lead is wanted, hence a large value of $N$. A larger value of $N$ moves an additional pole further from the origin and does not affect the dynamics, but amplifies the noise. To choose $N = 10$ seems to be a fairly standard choice in commercial controllers.

In an implementation the derivative part should be replaced by

$$\frac{sT_D}{1 + s\frac{|T_D|}{N}}. \tag{5.33}$$

Certain systems may require a negative $T_d$, and cause an unstable mode in the controller if the negative $T_d$ is used in the derivative filter. From a physical point of view it is meaningless to have a negative time constant in a filter.

EXAMPLE 5.6—Root locus with respect to $N$
Consider the plant

$$G(s) = \frac{1}{(s+1)^4}. \tag{5.34}$$

A PID controller was designed with $M_s = 2.0$ and $\alpha_0 = 1$. In Figure 5.10 is shown the root locus with respect to $N$ when $N$ assumes the values 2, 5, 10, 15, and 20. For values of $N$ of 10 or greater the dominant poles have not moved much.

A simulation of the controlled system is shown in Figure 5.11. The derivative filter constant has been $N = 10$ and $N = 20$. Measurement noise starts acting on the system at $t = 40$. As can be seen there is no visible difference in the set point and load responses. When noise is acting on the system the control signal is however twice as large for $N = 20$ as for $N = 10$. The conclusion is that $N$ should be chosen differently depending on the noise level in the system, and on how much control action that is allowed. □

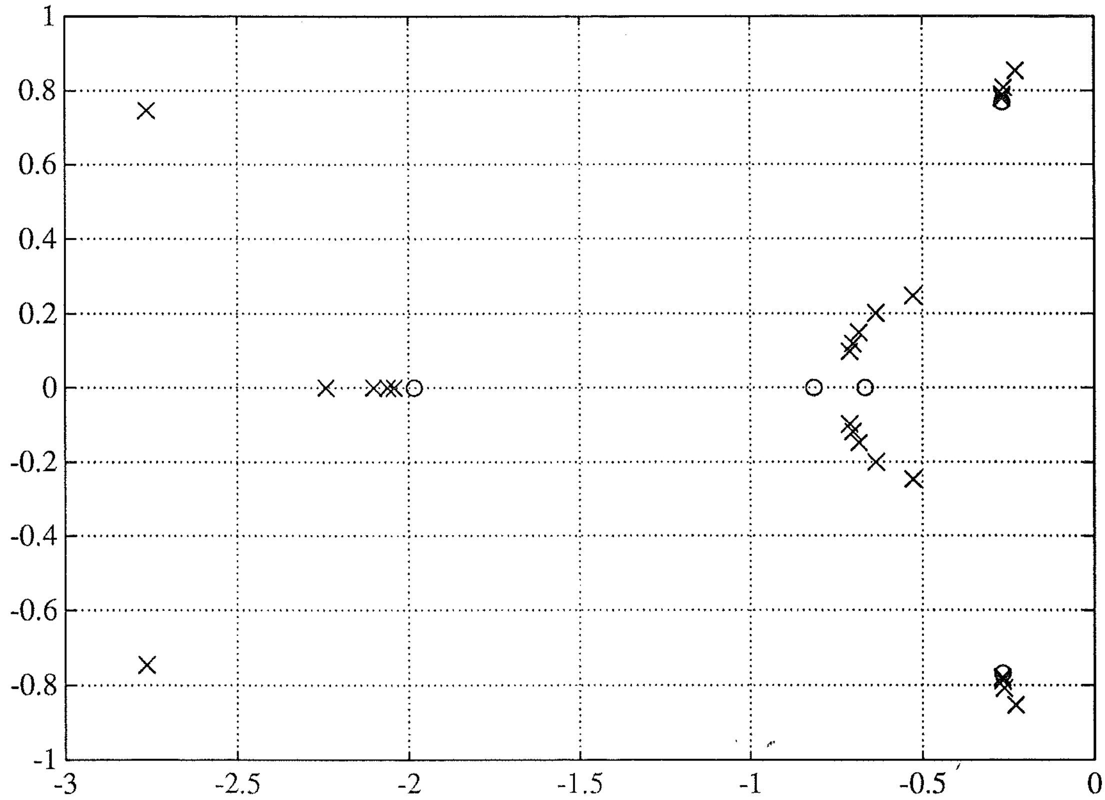



**Figure 5.10** Closed loop poles of $G(s)$ controlled by a PID controller designed with $M_s = 2.0$ and $\alpha_0 = 1$. The poles marked with 'o' correspond to $N = \infty$ (no derivative filter), poles marked with '×' correspond to $N = 2$, 5, 10, 15, and 20.

## Taking $N$ into account in the pole placement

Suppose we have a PID controller

$$G_{\text{PIDF}}(s) = k(1 + \frac{1}{sT_i} + \frac{sT_d}{1 + \frac{sT_d}{N}}), \tag{5.35}$$

and a plant $G(s)$. We want to place poles in $p_{1,2} = \omega_0(-\zeta_0 \pm \sqrt{1-\zeta_0^2})$, $p_3 = -\alpha_0\omega_0$ given $\alpha_0, \omega_0, \zeta_0$, and $N$. Solving

$$1 + G(p_i)G_{\text{PIDF}}(p_i) = 0, \tag{5.36}$$

for $i = 1, 2, 3$ gives an equation of the ninth degree in $k$ if we want to take $N$ into account. We do not get a set of linear equations anymore. The $\omega_0\zeta_0 k_d$ parametrization gives similarly complicated equations. The problem is not to solve the equations numerically, but to choose the correct solution out of many possible solutions.

This problem should not be confused with the case where we choose four poles, compute linear controllers, and then determine, $k, T_i, T_d$, and $N$. The

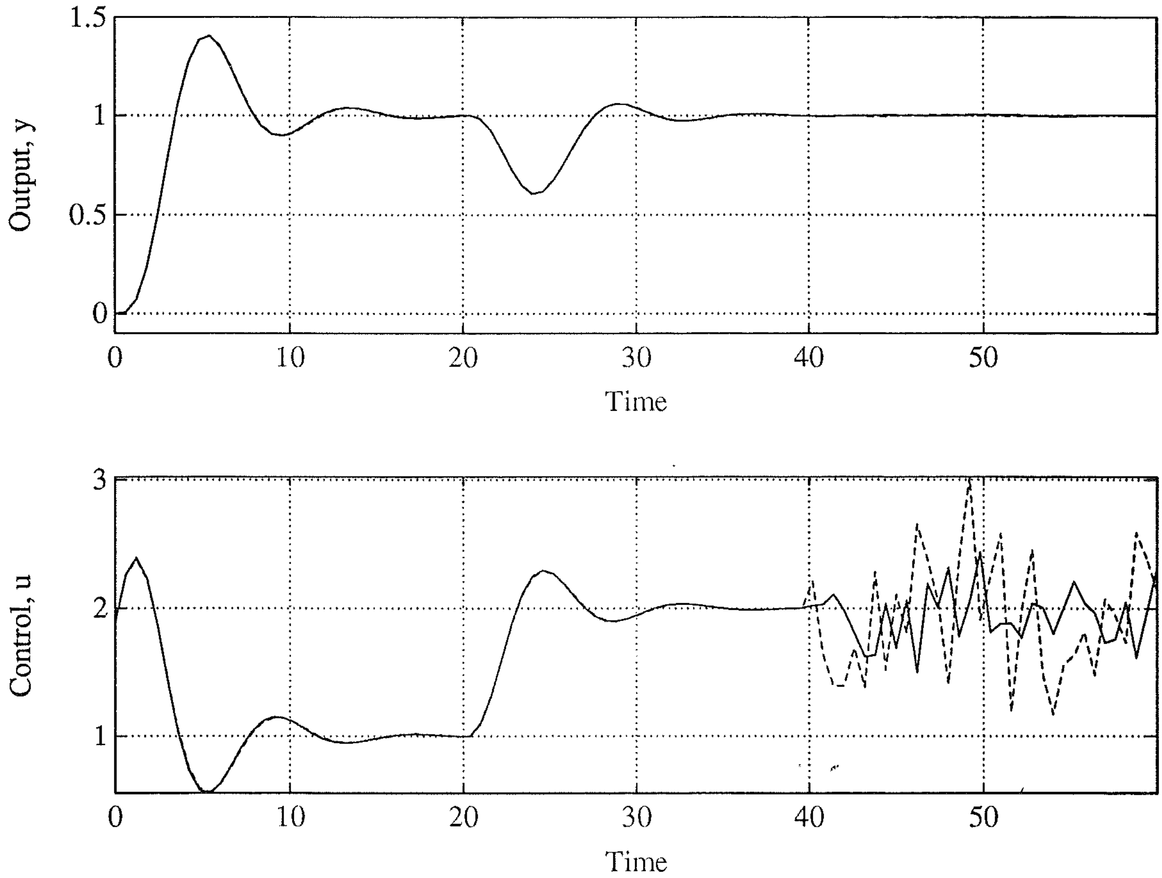



**Figure 5.11** Simulation of $G(s)$ controlled by a PID controller, $N = 10$ (solid line) and $N = 20$ dashed line.

parameter $N$ should be regarded as a design parameter and should not be specified by a pole.

However, in a specific case it is easy to solve the problem numerically. In the three pole case, choose the poles, $p_1, p_2$, and $p_3$ and the filter constant $N$. Compute $k$, $k_i$ and $k_d$ with (5.5), (5.6), and (5.7). These parameter values make good initial values for the numerical solution of (5.36) with respect to $k$, $k_i$, and $k_d$ for a controller with derivative filter. In case of the $\omega_0\zeta_0 k_d$ parametrization, the poles $\omega_0(-\zeta_0 \pm \sqrt{1-\zeta_0^2})$ and the parameter $k_d$ were considered to be correct, and modified values of $k$ and $k_i$ were computed. The numerical solution was made with routines from the Matlab Optimization Toolbox, see [MathWorks, 1990a].

EXAMPLE 5.7—Dominant Pole placement with $N$

Consider the same situation as in Example 5.6. Figure 5.12 shows the root locus with respect to $N$ when $k$ and $k_i$ have been recalculated to keep the dominant poles in their locations, specified by $\omega_0$ and $\zeta_0$. As can be seen from the figure there is no big difference for values of $N$ greater than 10. The recomputed parameters are shown in Table 5.1 for different values of $N$. □

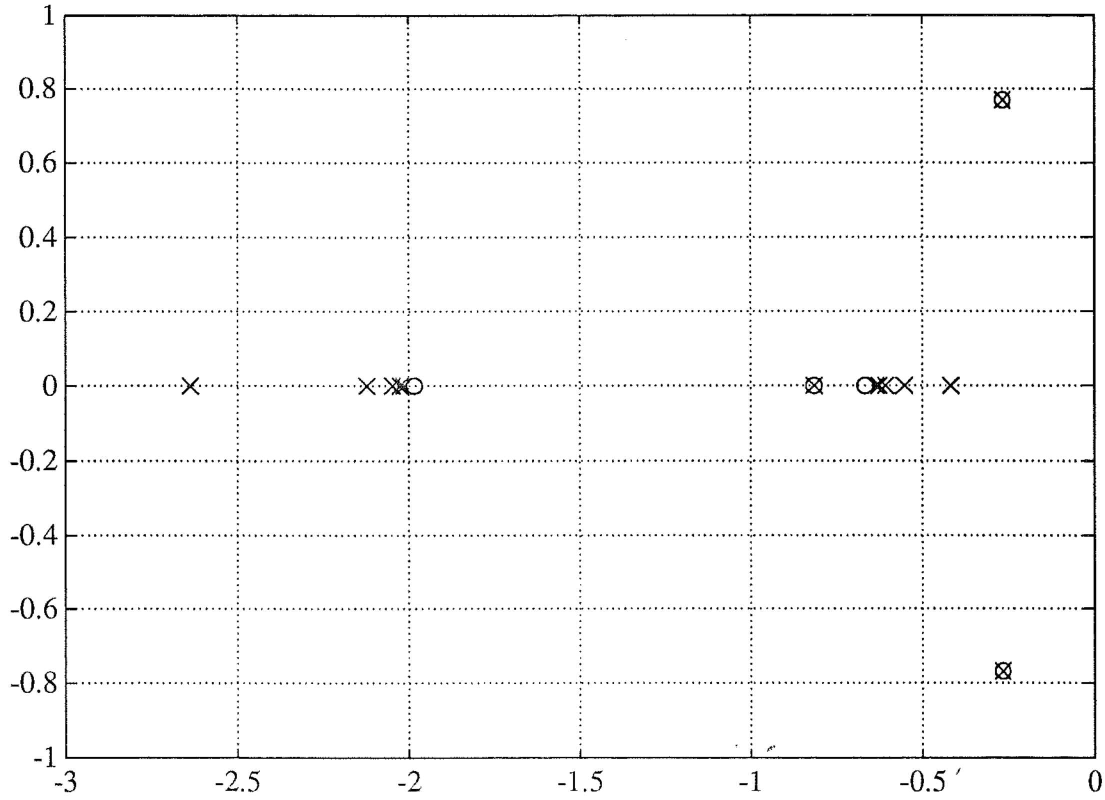



**Figure 5.12** Root locus with respect to $N$ with $k$ and $k_i$ recomputed to be in the same locations as without the derivative filter. The poles marked with 'o' correspond to $N = \infty$ (no derivative filter), poles marked with '×' correspond to values of $N$ from 2 to 20.

**Table 5.1**

| N | $k$ | $T_i$ | $T_d$ |
|---|---|---|---|
| 2 | 1.5585 | 2.9563 | 0.5559 |
| 5 | 1.7297 | 2.7372 | 0.6104 |
| 10 | 1.8028 | 2.6849 | 0.6327 |
| 15 | 1.8295 | 2.6697 | 0.6407 |
| 20 | 1.8433 | 2.6625 | 0.6449 |
| $\infty$ | 1.8867 | 2.6424 | 0.6580 |

## 5.4 Common issues for PI and PID controllers

### Set point weighting

By using a PI or PID controller with set point weighting, as defined in Chapter 2, we get the relation

$$Y(s) = \frac{k(s\beta + \frac{1}{T_i})G(s)}{s + k(s + \frac{1}{T_i})G(s)} Y_r(s) \tag{5.37}$$

between the set point signal $Y_r(s)$ and the output signal $Y(s)$. We see that the zero located in $-\frac{1}{\beta T_i}$ introduced by the integrator can be moved with the parameter $\beta$. This zero only affects responses to set point changes.

The most important use of the $\beta$ parameter is to reduce the overshoot in $y$ for a step disturbance in $y_r$. Setting $\beta = 0$ is in some cases too conservative, but it is still the standard choice in some commercial controllers. Instead we observe that an overshoot in the time domain in most cases correspond to $M_p > 1$, this is illustrated in Example 5.8.

EXAMPLE 5.8—Relation between overshoot and $M_p$
Consider

$$G(s) = \frac{e^{-sL}}{1 + sT}. \tag{5.38}$$

PI controllers were designed for $G(s)$ for $0.2 < \zeta_0 < 0.8$ and different $L/T$. The parameters $\beta$ was varied between 0 and 1, and corresponding overshoot values and $M_p$ values were computed for the closed systems. In Figure 5.13 the overshoot values are marked against the $M_p$ values. The figure was made to get a feel for how overshoot normally depends on $M_p$. Similar results were obtained for other plants. □

This suggests a method for choosing $\beta$. First set $\beta = 0$, and compute $M_p$. If $M_p > 1$, we will probably get an overshoot so this is the result of the $\beta$ design. If $M_p = 1$, i.e. there is no resonance peak, we increase $\beta$ until $M_p$ becomes greater than 1. Numerically we may solve $M_p(\beta) = 1.001$ say, with respect to $\beta$. An example of this way of choosing $\beta$ can be found in Chapter 7. This method of choosing $\beta$ assumes that an overshoot correspond to a resonance peak. This is not always the case. In general, the only way to choose $\beta$ to avoid overshoots is to select $\beta$, simulate and compute the overshoots numerically. Simulations studies and empirical formulas for $\beta$ in terms of process characteristics can be found in [Hang *et al.*, 1991].

The advantage of the $\beta$ modification is that the response to set point changes may be changed to get less overshoot, or faster response, depending on the system and controller. The set point weighting does not affect the

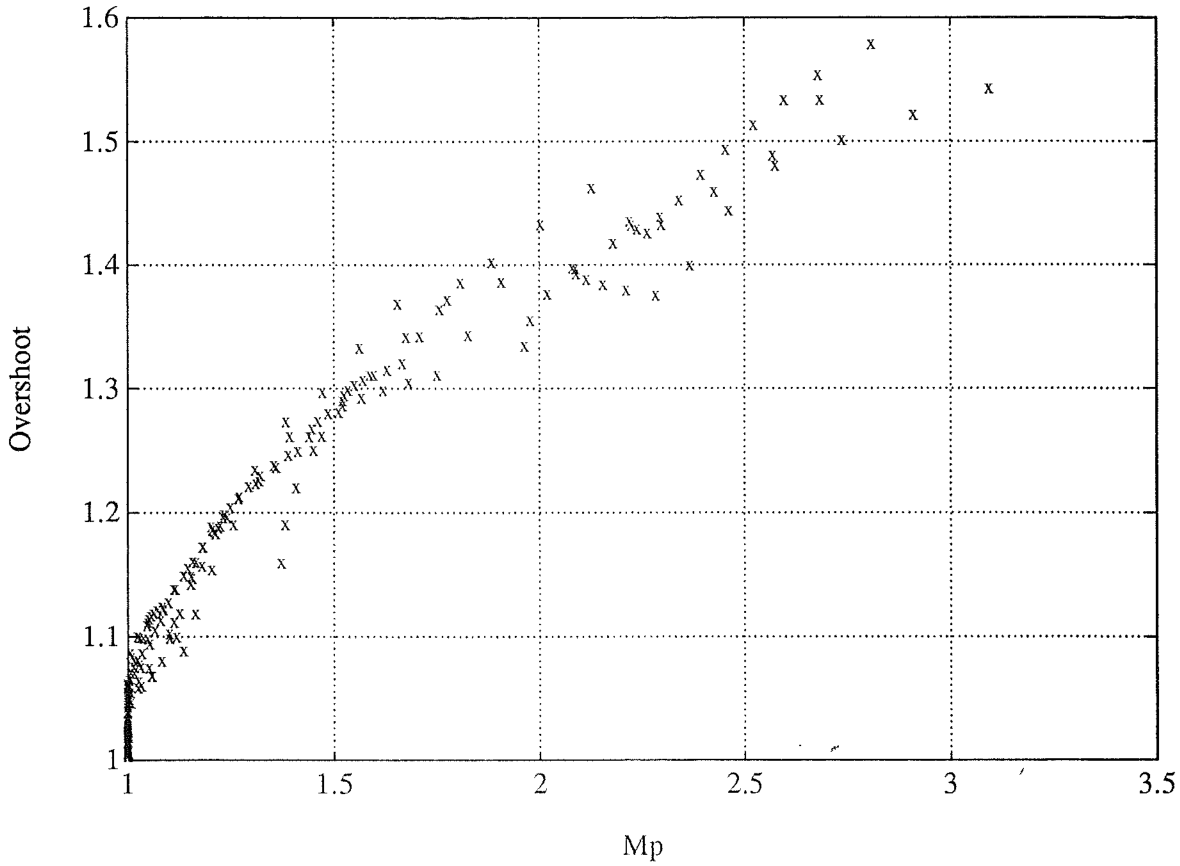



**Figure 5.13** Overshoot values marked against $M_p$ values.

system's load attenuation capability. The disadvantage is that we now have one more parameter to tune. Not all controllers permit change of the $\beta$ parameter. In commercially available controllers $\beta$ is usually set to 0 or 1. Commercial systems with adjustable $\beta$ have been reported, one such system is the Toshiba controller TOSDIC 211D8, see [Shigemasa *et al.*, 1987]. For a compilation of properties of commercial PID controllers, see [Åström and Hägglund, 1993]. We do not believe that this is the way to approach simple controllers; making them more and more complex by introducing more parameters which should be tuned according to some magical rules of thumb. Simple controllers should remain simple.

Overshoots can also be avoided by changing set point values with a ramp instead of abruptly with a step.

## 5.5 A design procedure

Two methods for PID controller design have been presented and discussed, the method of specifying three poles and the method of modifying a PI or PD controller.

The method of specifying three poles requires two tuning parameters, $\alpha_0$ and $M_s$. For most well damped systems $\alpha_0 = 1$ is a good choice. For resonant systems $\alpha_0$ should be chosen lower, $\alpha_0 \approx 0.5$. The parameter $\omega_0$ is obtained from the maximization of $k_i$, and $\zeta_0$ is chosen to give the closed loop system a specified $M_s$. The method is moderately computationally demanding. This method will be called 3PM (Three Pole Method).

The method of modifying a PI or PD controller requires two tuning parameters $M_{s1}$ and $M_{s2}$. First a well tuned PI controller is designed with $M_{s1}$ as design parameter. The $\zeta_0$ obtained from this design is then used in the modification of the PI controller. The parameter $k_d$ is increased, while $\omega_0$ is changed to keep $k_i$ maximal, until $M_s = M_{s2}$. To compute maximal possible interval $[M_{s1}\ M_{s2}]$ requires much computation. When the method is used for design, a pair of $M_{s1}$ and $M_{s2}$ is usually found without too much difficulty, by trial and error. However, this method requires much more computations than 3PM, and does not work well for resonant systems. The method will be called MCM (Modified Controller Method).

Both 3PM and MCM have $M_s$ as design parameter. Setting $M_s = 2$ is a sensible choice for most processes. If a more conservative design with a higher degree of robustness and with less overshoot is desired, a smaller value of $M_s$ should be chosen, 1.6 is usually a good choice.

The time responses for most well damped plants controlled by controllers designed with the two different methods look very similar. The method 3PM often gives a significantly larger $\omega_0$. Normally 3PM gives a slightly smaller value of $\zeta_0$ than MCM, and a smaller overshoot. Although the 3PM at first seems less general, the ease of using it and the broad spectrum of processes it can handle makes it a better choice than MCM.

Due to this 3PM will be recommended as the PID controller design method. This will be illustrated in Chapter 7.

The filter parameter $N$ should be considered as a design parameter and should not be chosen from pole placement. The value of $k(1 + N)$ is the controller's high frequency gain and should be such that the noise present does not disturb the process too much. From experience with examples it is seen that there is no need to take $N$ into account when the tuning of $k$, $k_i$, and $k_d$ is made. In practise $N$ is often set to 10, which is the value we will use in the following simulations.

## 5.6 Analytical examples

The formulas for $k$, $k_i$, and $k_d$ become very complex, and it is not very rewarding to solve anything but the very simplest case analytically

EXAMPLE 5.9—PID and second order system
Consider

$$G(s) = \frac{\omega_p^2}{s^2 + 2\omega_p\zeta_p s + \omega_p^2}. \tag{5.39}$$

With the $\alpha_0\omega_0\zeta_0$ parametrization the controller parameters are

$$k = \frac{\omega_0^2(2\alpha_0\zeta_0 + 1) - \omega_p^2}{\omega_p^2} \tag{5.40}$$

$$k_i = \frac{\alpha_0\omega_0^3}{\omega_p^2} \tag{5.41}$$

$$k_d = \frac{\omega_0(\alpha_0 + 2\zeta_0) - 2\zeta_p\zeta_0}{\omega_p^2}. \tag{5.42}$$

With the $\omega_0\zeta_0 k_d$ parametrization the controller parameters are

$$k = \frac{2\omega_p^2 k_d\omega_0\zeta_0 - \omega_p^2 + 4\zeta_p\omega_p\omega_0\zeta_0 + \omega_0^2(1 - \zeta_0^2) - 3\omega_0^2\zeta_0^2}{\omega_p^2} \tag{5.43}$$

$$k_i = \frac{(k_d\omega_p^2 + 2\zeta_p\omega_p - 2\omega_0\zeta_0)\omega_0^2}{\omega_p^2}. \tag{5.44}$$

In this case we can place all poles. This is completely analogous to PI control of a first order system. □

## 5.7 Summary

In this chapter design methods for PID controllers have been discussed. Two methods based on the dominant pole principle have been discussed and examined. In the first method three poles are placed to get the controller parameters. The second method uses results from a well tuned PI or PD controller to place two poles and gradually introduce derivative action. The method based on placement of three poles is recommended.

# 6

# Modeling for PI and PID control

In this chapter we are trying to find out if the design methods described in the previous chapters can be based on approximate knowledge of the Nyquist curve of the process. It will also be investigated if knowledge of the Nyquist curve in a special frequency range is of particular interest. We will assume that the frequency response of a dynamic system is known in a number of points. The design methods presented in Chapter 4 and Chapter 5 make use of $G(\alpha+i\beta)$, $\alpha \neq 0$. The usual frequency domain identification methods give $G(i\omega)$. To compute values of the transfer function in an arbitrary point in the complex plane a model must be fitted to the experimental data. The investigations in this chapter will be carried out mainly by examples.

Although the approximated transfer function $\hat{G}(i\omega)$ may be very close to the true $G(i\omega)$ in an interesting frequency interval it is not certain that $\hat{G}(e^{i\gamma}\omega)$ is close to $G(e^{i\gamma}\omega)$ in the same interval for $\gamma \neq \pi/2$. This may become very problematic when $\hat{G}$ has poorly damped complex poles.

The problem of getting a frequency domain model from a time domain experiment will not be addressed here. For readings in system identification, see [Ljung and Söderström, 1983]. For a survey of simple model estimation techniques, see [Rake, 1980] and [Unbehauen and Rao, 1987].

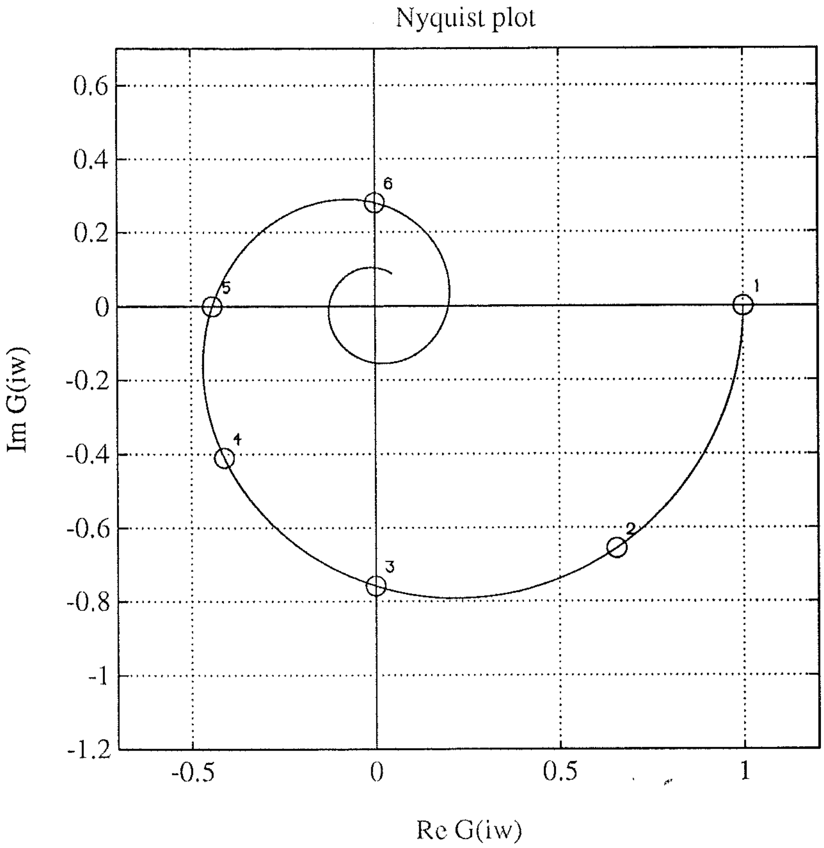



**Figure 6.1** Known points on the Nyquist curve.

## 6.1 Models from frequency domain data

Suppose we know a few points on the Nyquist curve of a process. These points may have been obtained from frequency analysis or from some kind of relay experiments. Relay experiments for obtaining information of transfer functions are discussed in [Åström and Hägglund, 1988] and have been used in autotuning of PID controllers. To use the information of $G(s)$ in a few points in the design computations we need to estimate a model from the available points. In our investigations it is assumed that the Nyquist curve is known in 2, 3, 4, 5, or 6 points. The points correspond to the phase lags 0, $\pi/4$, $\pi/2$, $3\pi/4$, $\pi$, and $3\pi/2$. The points will be used according to Table 6.1. The points of a typical Nyquist curve is shown in Figure 6.1. For processes with integrators we will make use of points corresponding to the phase lags $3\pi/4$, $\pi$, $5\pi/4$, and $3\pi/2$.

**Table 6.1**

| Number of points | Points in Figure 6.1 |
|---|---|
| 2 | 1, 5 |
| 3 | 1, 3, 5 |
| 4 | 1, 2, 4, 5 |
| 5 | 1, 2, 3, 4, 5 |
| 6 | 1, 2, 3, 4, 5, 6 |

The results of the approximations will be misleading if the plants are purely polynomial systems of too low order. The coefficients of the models will then be estimated with almost 100% accuracy from only a few points. Therefore we will make use of more complicated transfer functions of higher degrees or transfer functions with time delays. To be specific, the following processes will be considered

$$G_1(s) = \frac{e^{-s}}{1+sT}, \quad T = 1,\, 2,\, 10 \tag{6.1}$$

$$G_2(s) = \frac{1}{(1+s)^8} \tag{6.2}$$

$$G_3(s) = \frac{e^{-0.5s}}{(s+1)(s^2+2\zeta_p s+1)}, \quad \zeta_p = 0.3,\, 0.4,\, 0.5 \tag{6.3}$$

$$G_4(s) = \frac{e^{-s}}{(s+1)(sT+1)}, \quad T = 1,\, 2,\, 10 \tag{6.4}$$

$$G_5(s) = \frac{e^{-s}}{s(s+1)(sT+1)}, \quad T = 1,\, 2,\, 10. \tag{6.5}$$

Processes with and without integration require slightly different approximation methods, and allow different approximation points. They will therefore be treated separately.

### Processes without integration

In this section the model fitting methods used in the investigation will be presented. Different methods will be applied in the case where two points are known and in the case where more than two points are known. The two point case will be examined more thoroughly first.

***Transfer function known in two points*** Suppose we know the value of the frequency response in two points, $G(0) = g_0$ and $G(i\omega_\alpha) = g_\alpha e^{i\alpha}$. The parameter $g_0$ is easily estimated from the stationary value of a step response of the plant. Three parameters can be estimated from this knowledge of $G(s)$. We want the transfer function

$$G_1(s) = k_p \frac{e^{-sL}}{1+sT} \tag{6.6}$$

to agree with the known data in the two points. Solving the equations $G_1(i\omega_\alpha) = g_\alpha e^{i\alpha}$ and $G_1(0) = g_0$ for $k_p$, $L$, and $T$ gives

$$k_p = g_0 \tag{6.7}$$

$$T = \frac{1}{\omega_\alpha}\sqrt{(\frac{g_0}{g_\alpha})^2 - 1} \tag{6.8}$$

$$L = -\frac{\alpha + \arctan \omega_\alpha T}{\omega_\alpha}. \tag{6.9}$$

These formulas require that $g_0/g_\alpha > 1$. In particular, they work for any point of a Nyquist curve with monotonicly decreasing amplitude and phase.

The data obtained from a relay experiment or in the Ziegler-Nichols self cycling method correspond to $\alpha = -\pi$. If the relay has hysteresis then the estimated point corresponds to $\alpha > -\pi$. The nonlinearity suggested in [Holmberg, 1991] can be used to obtain points on the Nyquist curve corresponding to any $\alpha$. This nonlinearity can be described as a relay with hysteresis proportional to the amplitude of the input signal to the process, and gives a describing function with a constant phase.

To examine the influence of $\alpha$, the following systems were investigated

$$G_{\alpha 1}(s) = \frac{1}{(s+1)^8} \tag{6.10}$$

$$G_{\alpha 2}(s) = \frac{1-s}{(s+1)^3} \tag{6.11}$$

$$G_{\alpha 3}(s) = \frac{1}{(s+1)(s^2+1.2s+1)} \tag{6.12}$$

$$G_{\alpha 4}(s) = \frac{e^{-s}}{(1+s)(1+s)}. \tag{6.13}$$

PI and PID controllers for these systems were designed with 2PM and 3PM with $M_s = 2.0$ and $\alpha_0 = 1.0$. In Figure 6.2 a number of quantities are shown which are the ratios between the parameters obtained from a design with a model corresponding to a certain $\alpha$ and the parameters obtained from a design with the true model.

For PI controllers the ratios are closest to 1 for lower $\alpha$. PID controllers have ratios closer to 1 for higher $\alpha$. The interpretation is that more high frequency information is needed for PID controller design than for PI controller design. The $M_s$ values were also examined, but showed no significant changes. Note that for PI control $\text{IAE}/\text{IAE}_0 < 1$ for some systems and some $\alpha$. This means that the approximated model gives better control *in this respect*, than the true model. The purpose of the investigation was to examine for which $\alpha$ the approximated model gives results as close as possible to the true model, not to find optimal controllers. Observe that the figures are differently scaled. The PI controllers parameters from the estimated models deviate more from the true values than the corresponding PID parameters. Note also that although the PI parameters for $\alpha = -\pi$ may be 50% to 100% too large, the corresponding IAE does not deviate more than 30%.

The ratios for PID control tend to be close to 1 for a phase lag a little less than $\pi$. This is the frequency range where relay based auto tuners operate. Thus, a relay experiment can be expected to give good models for PID controller design, but not so good for PI controller design.

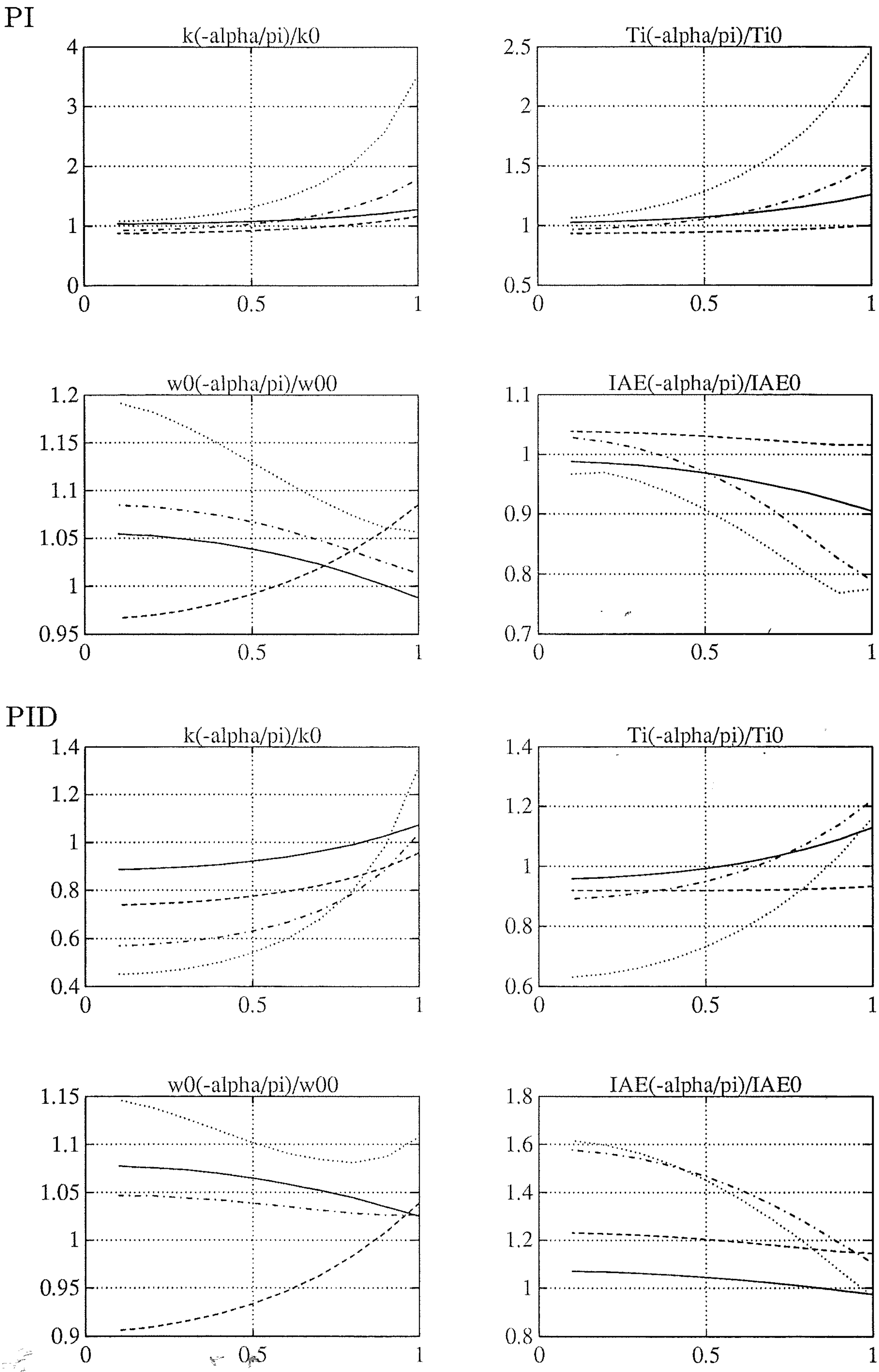



**Figure 6.2** A number of quantities obtained from controller design with a model estimated from two points, $G(0) = k_p$ and $G(\omega_\alpha) = g_\alpha e^{i\alpha}$, shown as functions of $\alpha/\pi$. The quantities have been divided by the values obtained for the true model. $G_{\alpha 1}$ (solid line), $G_{\alpha 2}$ (dashed line), $G_{\alpha 3}$ (dotted line), and $G_{\alpha 4}$ (dashed-dotted line). The ratios are shown for PI and PID controllers.

The resonant system $G_{\alpha 3}$ was the one which was the most difficult to approximate with $G_1$. This is seen in Figure 6.1 where the parameters from estimations of $G_{\alpha 3}$ deviates most from the true parameters.

PI controllers designed with models for small $\alpha$ give results very close to the results obtained with the true model, but a too small value of $\alpha$ will make the approximation very sensitive to errors in $G(\omega_\alpha)$. A good choice for PI controller design is $\alpha = -\pi/2$ and for PID controller design $\alpha = -\pi$. To be able to compare all approximations equally, we will use $\alpha = -\pi$ for all controller designs in the following.

***Transfer function known in more than two points*** Suppose we know the transfer function in $m$ points $G(\omega_r)$, $r = 1 \ldots m$. A method for fitting transfer functions of structure

$$G(s) = \frac{b_0 s^{n_b} + b_1 s^{n_b - 1} + \ldots + b_{n_b - 1} s + b_{n_b}}{s^{n_a} + a_1 s^{n_a - 1} + \ldots + a_{n_a - 1} s + a_{n_a}}$$

to such data is presented in [Lilja, 1989]. The method requires that $2m > n_a + n_b + 1$. Suppose that we want a pole excess of $p$, i.e., $n_a = n_b + p$. This gives $n_b < (2m - p - 1)/2$. The design methods proposed in Chapter 4 and Chapter 5 will give a controller if the pole excess is $\geq 2$. We will examine the cases where $p = 2$. The values of $n_a$ and $n_b$ will be those in Table 6.2, where polynomials of as high degree as possible have been chosen. Models with $p = 3$ tended to give poorly damped poles which can be difficult to handle in the design. Therefore all investigations were carried out for $p = 2$.

**Table 6.2**

| $p$ | $m$ | $n_b$ | $n_a$ |
|---|---|---|---|
| 2 | 3 | 1 | 3 |
| 2 | 4 | 2 | 4 |
| 2 | 5 | 3 | 5 |
| 2 | 6 | 4 | 6 |

## Processes with integration

For processes with integration we can fit similar models, but with slightly different preconditions.

***Integrator gain known and transfer function known in one point*** A reasonable assumption is that we know the integration gain and one point on the Nyquist curve. Knowing the integrator gain corresponds to knowing the slope in steady state of the step response of the plant. Now the model

$$G_1(s) = k_p \frac{e^{-sL}}{s(1 + sT)} \tag{6.14}$$

can be fitted to the given data, $G(i\omega_\alpha) = g_\alpha e^{i\alpha}$ and $k_p$. Solving the equation $G_1(i\omega_\alpha) = g_\alpha e^{i\alpha}$ for $L$ and $T$ gives

$$T = \frac{1}{\omega_\alpha}\sqrt{(\frac{k_p}{g_\alpha \omega_\alpha})^2 - 1} \tag{6.15}$$

$$L = -\frac{\pi/2 + \alpha + \arctan \omega_\alpha T}{\omega_\alpha}. \tag{6.16}$$

These formulas require that $k_p^2 > (g_\alpha \omega_\alpha)^2$.

***Transfer function known in more than two points*** Assume that $G(s)$ can be written

$$G(s) = \frac{B(s)}{A(s)}\frac{1}{s}. \tag{6.17}$$

To use the routines in [Lilja, 1989], form $g_x = \omega_x G(i\omega_x)$ and use the points $g_x$ as inputs to get $B(s)/A(s)$. Here we must know that the plant contains an integrator. The complete transfer function $G(s) = B(s)/(sA(s))$ is then used in the design.

### Results of the investigations

The results of the investigations will be presented in this section. The approximated models have been used for designing PI and PID controllers. The PI controllers have been designed with 2PM with $M_s = 2.0$. The PID controllers have been designed with 3PM with $M_s = 2.0$ and $\alpha_0 = 1.0$. For the resonant systems was used in the PI case $M_s = 2.5$, and in the PID case $M_s = 2.5$ and $\alpha_0 = 0.3$.

The results of design of controllers using the different approximated models are compared to the result of design of controllers using the true models. The results of the computations are shown in Figures 6.3, 6.4, 6.5, 6.6, and 6.7. In the figures each quantity has been divided by the quantity obtained from design with the true model. The ratios are plotted against the number of approximation points. The quantity IAE is the integrated absolute error when the controlled, true system, gets a load disturbance on the input of the process. In each figure are given the results of both PI control and PID control.

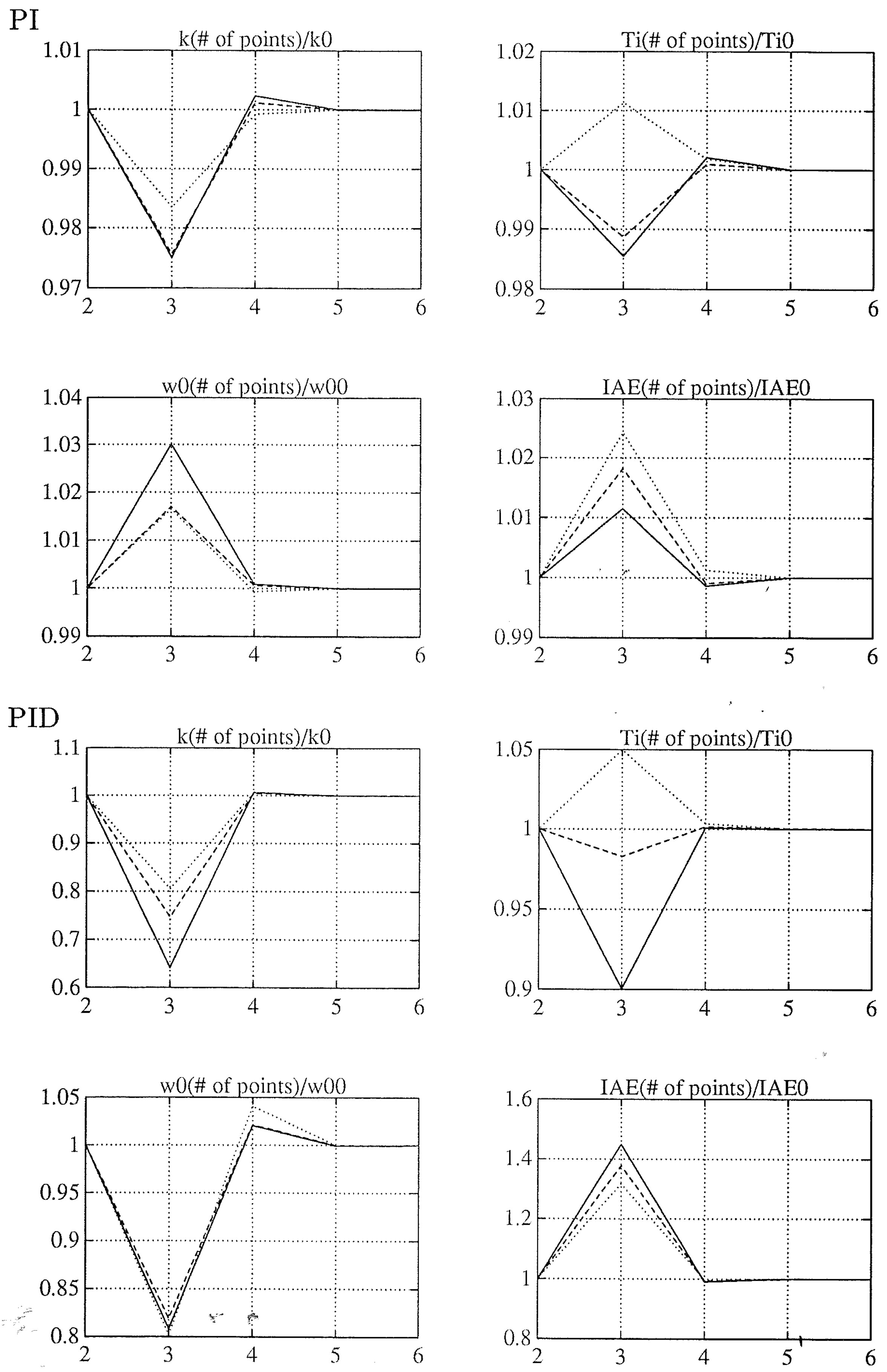



**Figure 6.3** Results for $G_1$, $T = 1$ (solid line), $T = 2$ (dashed line), and $T = 10$ (dotted line). Results are shown for PI and PID controllers.

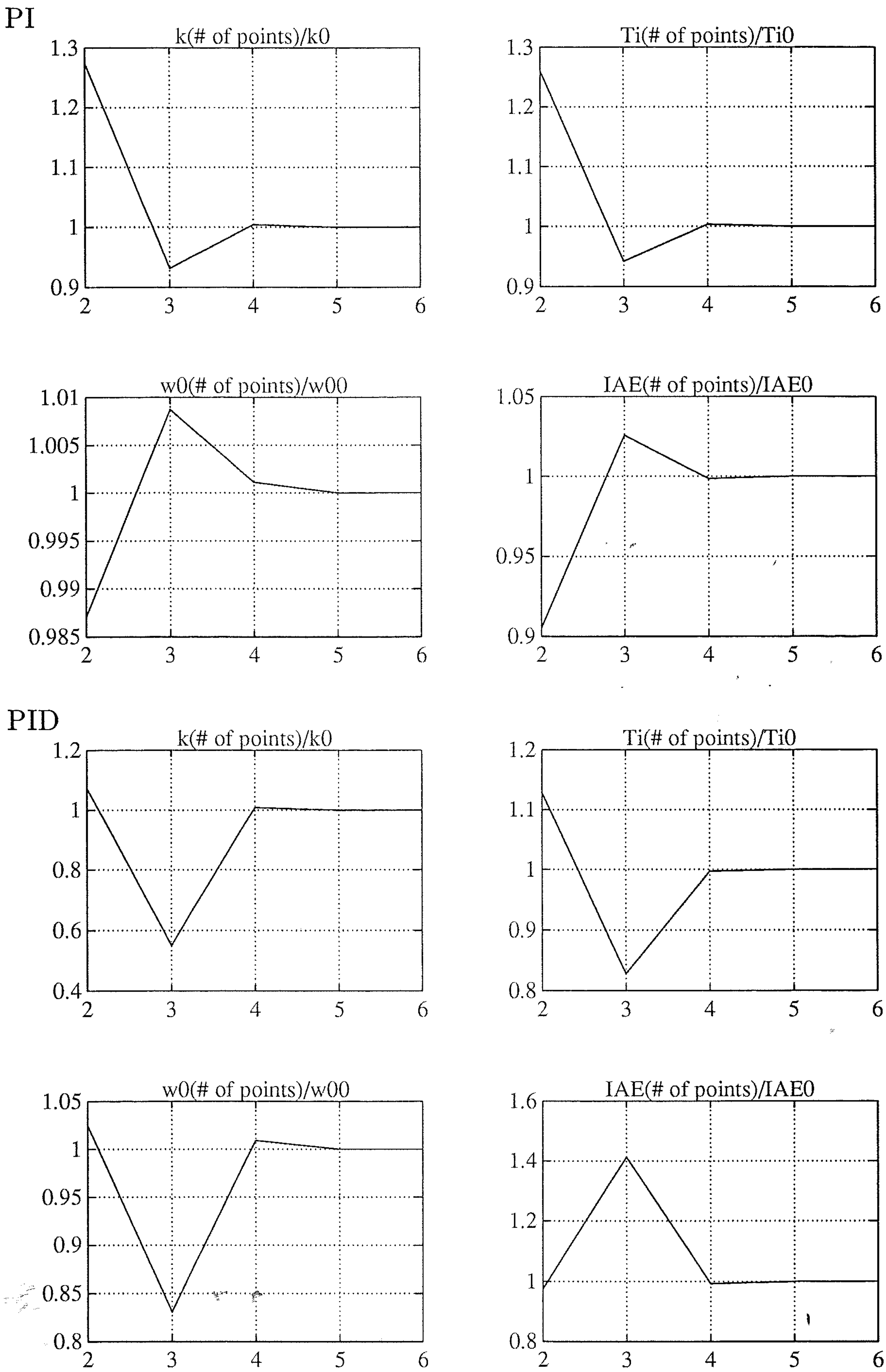



**Figure 6.4** Results for $G_2$. Results are shown for PI and PID controllers.

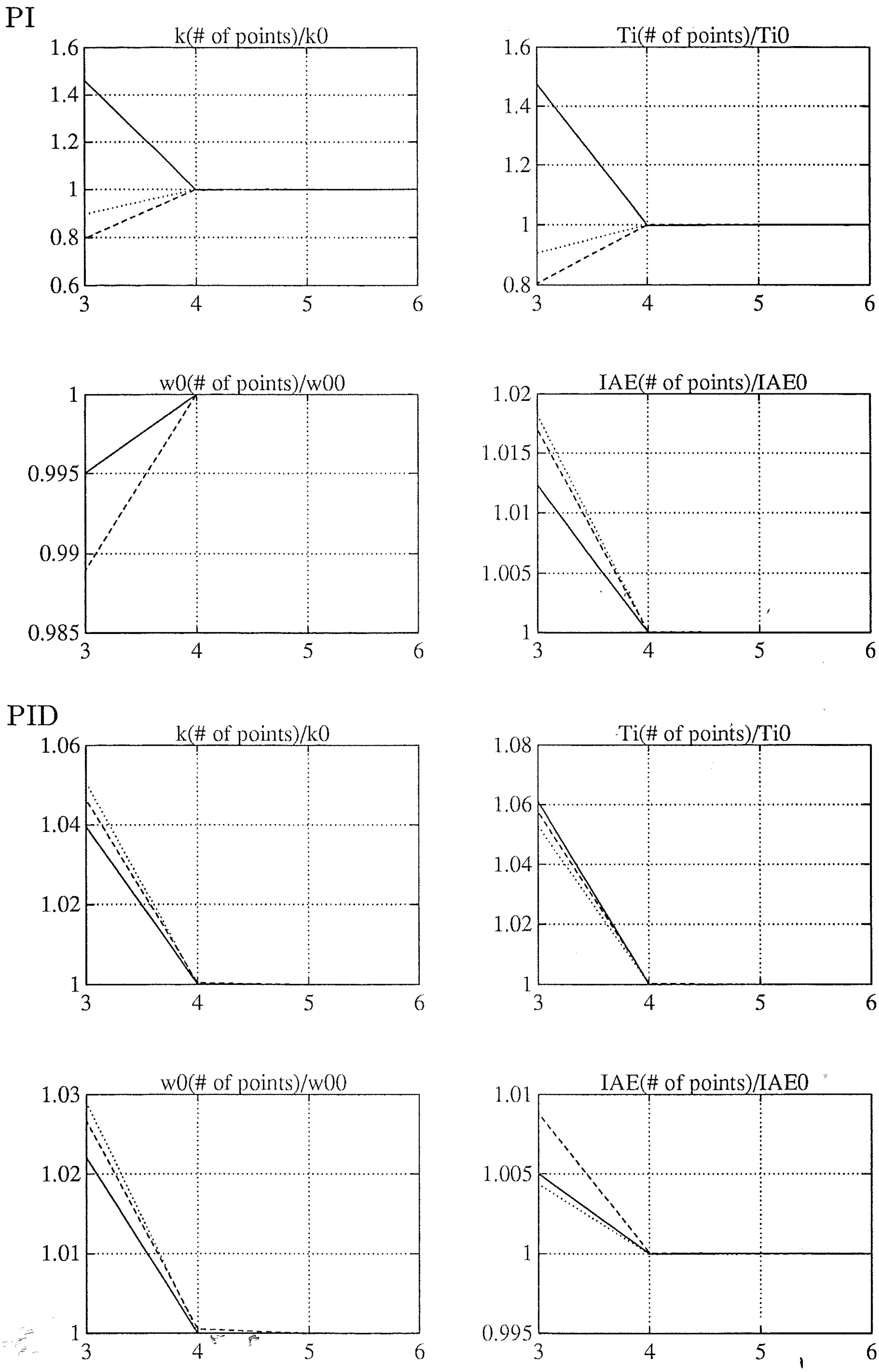



**Figure 6.5** Results for $G_3$. The lines correspond to $\zeta = 0.3$ (solid line), $\zeta = 0.4$ (dashed line), and $\zeta = 0.5$ (dotted line). Results are shown for PI and PID controllers.

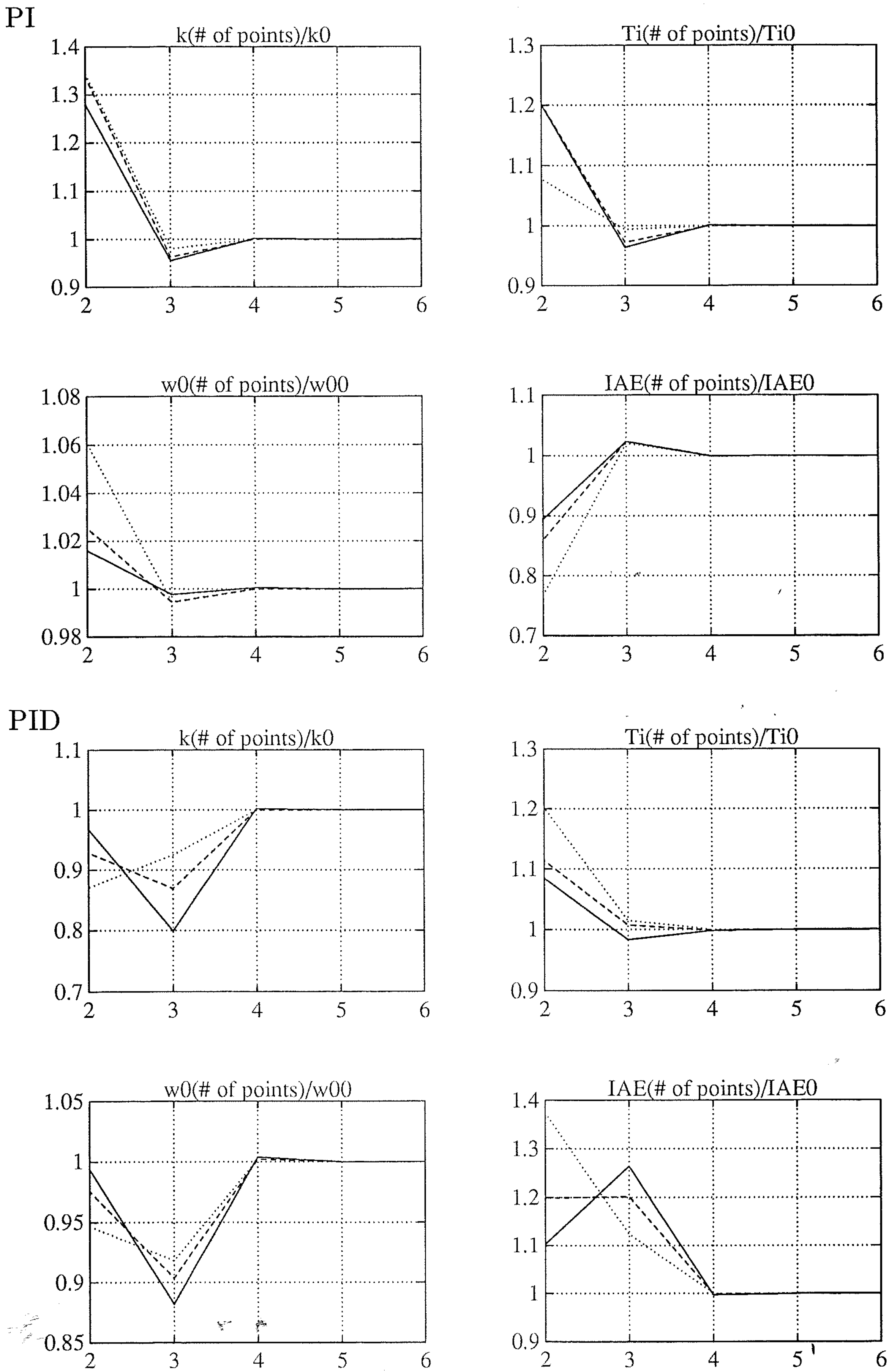



**Figure 6.6** Results for $G_4$, $T = 1$ (solid line), $T = 2$ (dashed line), and $T = 10$ (dotted line). Results are shown for PI and PID controllers.

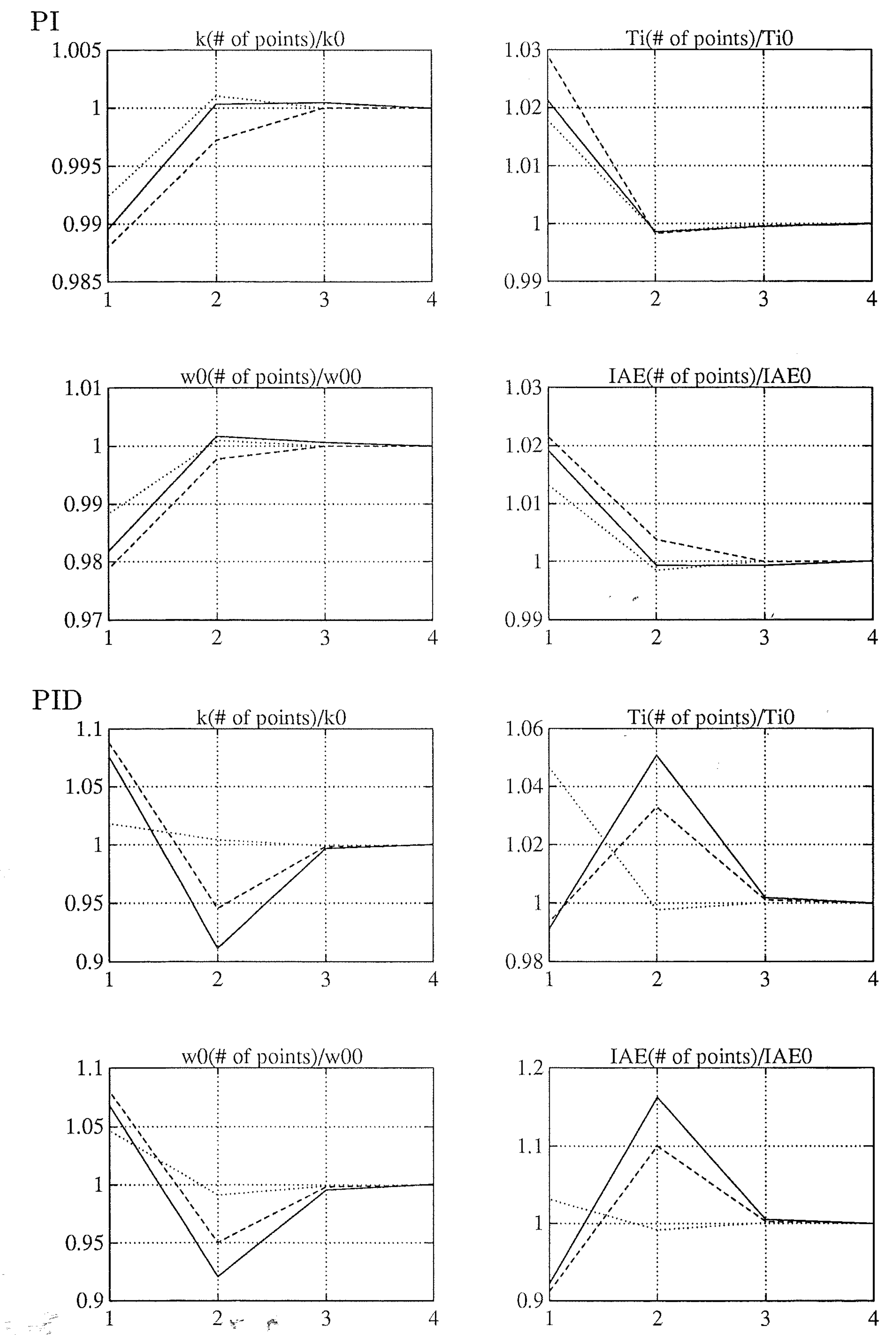



**Figure 6.7** Results for an integrating process $G_5$, $T = 1$ (solid line), $T = 2$ (dashed line), and $T = 10$ (dotted line). Results are shown for PI and PID controllers.

## Conclusions

The aim has been to investigate how different points on the Nyquist curve affect the design, rather than to find out which points were optimal. Therefore those estimated models were considered to be 'best' which have the ratios closest to 1.

***Non-integrating processes*** For the systems $G_1$, $G_2$, $G_3$, and $G_4$, approximations with 2, 3, 4, 5, and 6 points have been computed according to the methods presented above. In all two point estimations $\alpha = -\pi$ was used. No two point estimations were made for the resonant systems. The results of these computations are presented in Figures 6.3, 6.4, 6.5, and 6.6.

The two point approximations give PID control more close to the PID control obtained from the true model than PI control to the PI control of the true model. The reason is that this model is more correct for relatively high frequencies, and PID controller design uses more high frequency information than PI controller design. For $G_1$ the two point estimation, of course, gives the same result as with the true model. The reason is that the estimated model and the true model have exactly the same structure, and depend on only three parameters.

The three point approximation give better relative approximation for PI controller design than for PID controller design. The emphasis is here on lower frequencies, which is more important for PI controllers.

In all cases the 4, 5, and 6 point approximations yield practically identical results. Information from points with phase lags greater than $\pi$ do not give better performance of the closed system.

***Integrating processes*** The process $G_5$ was the only integrating process which was tested. The one point approximation was done for $\alpha = -\pi$. The results of these computations are presented in Figure 6.7.

Both one and two point estimation gives PI control closer to the PI control obtained from the true model than PID control to the PID control from the true model.

For the integrating process the 3 and 4 point approximations give the same result. For the design of PI and PID controllers for this integrating system there is no need for information of points with phase lag greater than $5\pi/4$.

PI and PID controllers are controllers with simple structures, hence one can assume that it is sufficient to have a good model for low frequencies.

When we make the controller design we do not optimize the controller with respect to IAE, hence it may very well happen that an approximate model give a better design in terms of IAE than the true model.

The step and load responses from design with information from different sets of data points do not look very different. From a practical point of view a 2 or 3 point estimation is usually sufficient. IAE of PID control was always less than IAE of PI control.

## 6.2 Summary

We have investigated how the design of PI and PID controllers depends on the available information of the frequency response of a plant. The design methods developed in Chapter 4 and Chapter 5 were used. For the investigation transfer functions were estimated given points on the Nyquist curve of known plants. PI and PID controllers were designed for the true system and for the systems obtained from the model estimations. The ratios between various quantities from the approximate systems and the true systems are shown in figures.

This empirical investigation has shown that it is indeed possible to base the design on ordinary frequency response data. There is a difference between PI and PID control in requirements of the process information. Not surprisingly, PI control design benefits from information at lower frequencies. PID controller design needs information from higher frequencies.

It has been shown that a sufficiently good model for PI and PID controller design can be estimated from information from points with phase lags $\pi/4$, $\pi/2$, $3\pi/4$, and $\pi$. No investigations were made on how to choose these points optimally.

For non-integrating processes information from points with phase lags greater than $\pi$ is of little use. Design of integrating processes requires information to a greater phase lag.

# 7

# Examples and Comparisons

In this chapter we will give some examples of the use of the design methods presented in the previous chapters. The methods will be applied to a number of standard systems. Comparisons with methods like Ziegler-Nichols, Cohen-Coon, IAE, ISE, and ITAE will also be given.

## 7.1 Examples and comparisons

The methods for design of PI and PID controllers presented in Chapter 4 and Chapter 5 will be used and compared in this section.

### Design methods

PI controllers will be designed by chosing $\zeta_0$ such that the closed system gets a prescribed $M_s$ value while $\omega_0$ is chosen to maximize $k_i$. This method will be denoted 2PM (Two Pole Method). In some cases the modified PI design method will be used. This method means that $\omega_0$ is increased beyond the maximum of $k_i$ until $k_i$ has been reduced to $0.8k_{i,\max}$. This method will be denoted MPI (Modified PI).

PID controllers will be designed by two methods. In the first method three poles are specified with $\alpha_0$, $\omega_0$, and $\zeta_0$. The parameter $\alpha_0$ will be set to 1 in most cases, except in the case of resonant systems. The parameter $\omega_0$

will be chosen to maximize $k_i$, and $\zeta_0$ will be chosen to get a prescribed $M_s$ value. This method will be denoted 3PM (Three Pole Method). The second method starts with a PI controller tuned to have $M_s = M_{s1}$, this controller is then modified by adding derivative action and recomputing $k$ and $k_i$ until the final controlled system has $M_s = M_{s2}$. The design parameters are $M_{s1}$ and $M_{s2}$. This method will be denoted MCM (Modified Controller Method). The design methods are described in detail in Chapter 4 and Chapter 5.

### The PI and PID controllers

In this chapter we will use the PID controller on series form, with the derivative filter constant $N = 10$. The set point signal has not been differentiated. The simulations have been made with the simulation package Simnon, see [SSPA, 1990]. The results of the simulations are shown in figures where the output signal of the process is in the upper half in the figures and the control signal is in the lower half. The initial part of the simulation shows the response to a step change in the set point signal. At half the simulation time a constant load disturbance acts on the input of the process. Load disturbance attenuation is evaluated with the criteria IE and IAE.

The robustness of the controlled system will be investigated by computing the time responses when a parameter in the process is changed while keeping the controller constant. Robustness is also evaluated by the sensitivity $M_s$ and the relative damping of the dominant poles.

The response to command signals can be changed by the set point weighting factor $\beta$. By introducing $\beta$ the controller has a two degree of freedom structure. The parameter $\beta$ is set to 0 in the simulations to get small overshoots. To get faster set point responses, and possibly more overshoot, $\beta$ may be increased.

A crude measure of the sensitivity to measurement noise is the high frequency gain of the controller, for PI controllers $k$ and for PID controllers $k(N+1)$.

### The design parameters

$M_s$ is the primary design parameter in the methods used here. The following example is given to get a feel for the meaning of $M_s$ as a design parameter.

EXAMPLE 7.1—$M_s$ as design parameter
PI and PID controllers were designed for the process

$$G(s) = \frac{1}{(1+s)^8} \tag{7.1}$$

with 2PM and 3PM for $M_s = 1.5$, 2.0, and 2.5. The time responses of the systems are found in Figure 7.1. As can be seen in the figure we get more well

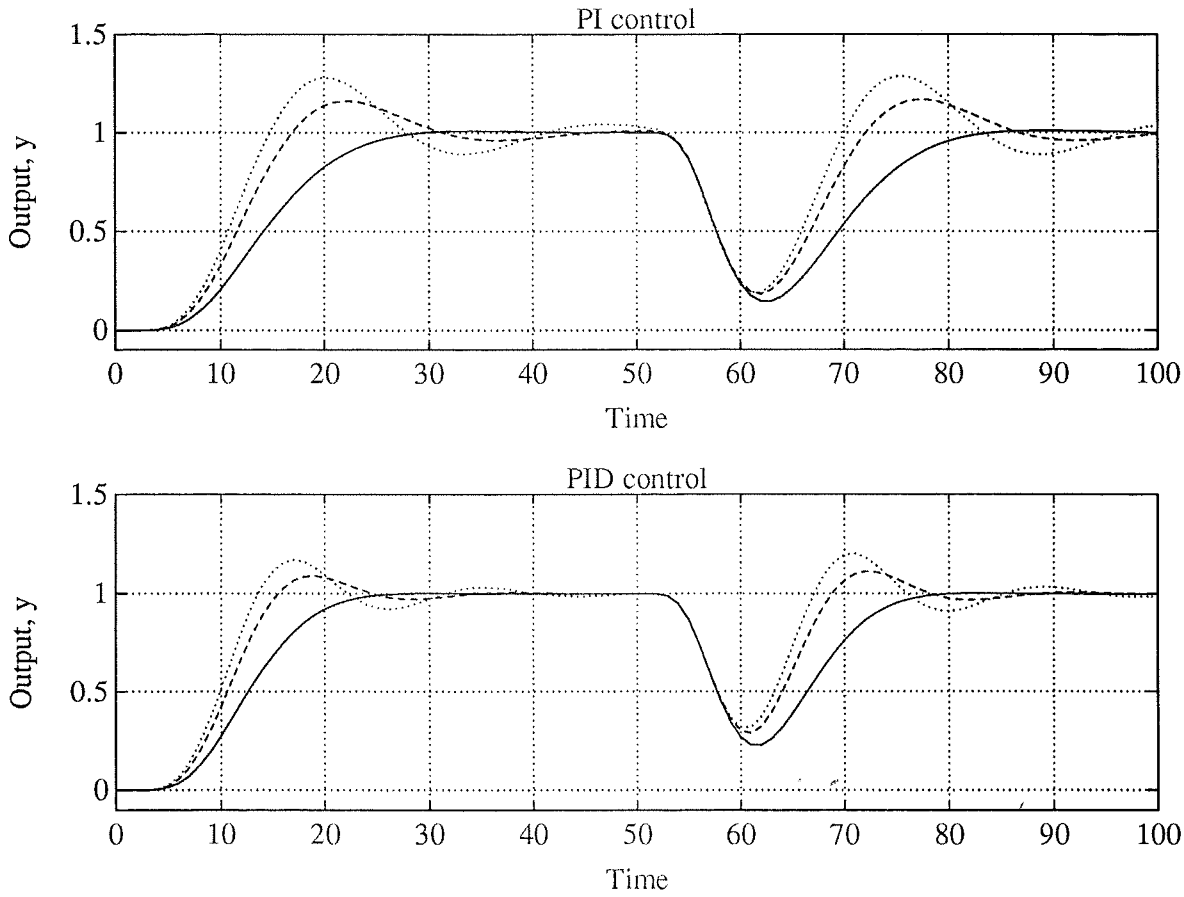



**Figure 7.1** PI and PID control with different $M_s$ values in the design. The values used were $M_s = 1.5$ (solid line), $M_s = 2.0$ (dashed line), and $M_s = 2.5$ (dotted line).

damped systems for lower $M_s$ values. Responses with PI and PID controllers are similar in shape, but PID controllers give a faster response. The IAE values were in the PI case 12.0, 10.5, and 11.4 and the IE values were 11.8, 7.4, and 6.0. Corresponding values for the PID case were for IAE 8.9, 7.0, and 7.3 and for IE 8.8, 5.6, and 4.4. The IE values decrease with increased $M_s$. The IAE values become large when IE is large or when IE is small due to big overshoots, as in the case of $M_s = 2.5$. Sensible values of $M_s$ for use in tuning usually are between 1.5 and 2.2. □

The example shows that parameter $M_s$ is a good design variable. Variation in the responses are easily obtained by changing $M_s$. Responses without overshoot can often be obtained by choosing $M_s$ sufficiently small. In this thesis we have designed with $M_s = 2.0$ as a standard value. The value was chosen because it is possible to design almost all plants with $M_s = 2.0$. This may be considered somewhat high. In many applications this value could be reduced. Lower $M_s$ values can be obtained with more complex controller structures.

## Test processes

Controllers for a number of systems have been designed and investigated. In this section the following systems will be investigated

$$G(s) = \frac{e^{-s}}{1+sT}, \quad T = 1, 4 \tag{S1}$$

$$G(s) = \frac{1}{(1+s)^n}, \quad n = 4, 8 \tag{S2}$$

$$G(s) = \frac{1}{(1+s)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}, \quad \alpha = 0.1, 0.2 \tag{S3}$$

$$G(s) = \frac{1-\alpha s}{(1+s)^3}, \quad \alpha = 0.2, 2 \tag{S4}$$

$$G(s) = \frac{\alpha}{(\alpha+s)(s^2+2\zeta_p s+1)}, \quad \zeta_p = 0.3, \alpha = 0.3, 3.3 \tag{S5}$$

$$G(s) = \frac{e^{-s}}{s(1+Ts)}, \quad T = 1, 4. \tag{S6}$$

The system S1 is a standard system that has been used for a long time in the studies of PID controllers. There is no loss in generality to assume that the dead time is 1, as is shown in Example 7.2.

Example 7.2
If the plant

$$G(s) = \frac{e^{-sL}}{1+sT} \tag{7.2}$$

is given, and we introduce $s' = sL$ and $\theta = L/T$ we get

$$G'(s') = G(s'/L) = \frac{e^{-s'}}{1+s'/\theta}. \tag{7.3}$$

Hence it is sufficient to study

$$G(s) = \frac{e^{-s}}{1+sT}. \tag{7.4}$$

Controller parameters for other transfer functions of type (7.2) can be obtained by scaling. □

The drawback of the system S1 is that the high frequency roll-off is quite slow. With a controller having derivative action the loop transfer function will have constant gain over a wide range of frequencies. For this reason it has been claimed that this is an unrealistic process. Still it is often used as a standard test process. The system S2 is another standard system. It is minimum phase and similar to S1 in the sense that both systems have significant phase lags. System S2 has however much more high frequency roll-off. The system S3 is also a multipole system where the spread of the poles is adjusted by parameter $\alpha$. The system S4 is a simple system with a zero in the right half plane. The system S5 is a system with a resonance. It represents a system where PID controllers cannot be expected to do so well. The system S6 is a simple system with integral action.

The open loop process gain is normalized to 1 in all cases, since this gain only affects $k$ by a constant factor. For processes with several time constants and time delays it is only interesting to change the ratio between these constants and delays. Controllers for other changes can be computed by scaling.

A more thorough examination will be carried out on S1, where formulas are available for optimal controllers for most integral criteria. Designs of the other systems will be presented more briefly.

### System S1 – $G(s) = e^{-s}/(1 + sT)$

***Design with 2PM, 3PM, and MCM*** In this section the system S1 will be investigated. Designs have been made with 2PM, 3PM, and MCM. The following design parameters were used, in 2PM $M_s = 2.0$, in 3PM $\alpha_0 = 1$ and $M_s = 2.0$, and in MCM $M_{s1} = 1.8$ and $M_{s2} = 2.0$. Responses to set point and load changes when the system is controlled by PI and PID controllers are found in Figure 7.2. Two cases are shown, a) $T = 1$ and b) $T = 4$. Table 7.1 gives a summary of the simulations.

**Table 7.1** Results of designs and simulations of S1

| Case | $k$ | $T_i$ | $T_d$ | $\omega_0$ | $\zeta_0$ | IE | IAE |
|---|---|---|---|---|---|---|---|
| 1a 2PM | 0.6334 | 0.8882 | | 1.1204 | 0.4614 | 1.4022 | 1.8397 |
| 1a 3PM | 1.0656 | 1.1724 | 0.2227 | 1.7024 | 0.5452 | 1.1002 | 1.1897 |
| 1a MCM | 1.0208 | 1.0427 | 0.2335 | 1.3876 | 0.5584 | 1.0215 | 1.2736 |
| 1b 2PM | 2.2387 | 2.1886 | | 0.8524 | 0.5434 | 0.9777 | 1.1176 |
| 1b 3PM | 3.4359 | 2.3162 | 0.2583 | 1.4186 | 0.7527 | 0.6741 | 0.6760 |
| 1b MCM | 3.2403 | 2.1496 | 0.2343 | 1.0790 | 0.6642 | 0.6634 | 0.7160 |

By going from PI to PID control we get a reduction of 35% in IAE in case a) and 40% in case b). Going from PI to PID control corresponds to a large increase of $k$ (68% and 53%), but a moderate increase of $T_i$ (32% and 5%). There is little difference between 3PM and MCM. The 3PM forces the closed loop system to have a pole on the negative real axis. MCM usually ends up with two pairs of complex poles in the closed system. Systems designed with 3PM usually get a larger $\omega_0$ than with MCM. The double pairs of complex poles also give a slightly larger overshoot in the set point responses. We get smaller IAE for the case $T = 4$ than for the case $T = 1$. The reason is that there is a time delay in the process. During the first time unit after a load disturbance the system runs in open loop and the error increases faster in case of $T = 1$ than for $T = 4$.

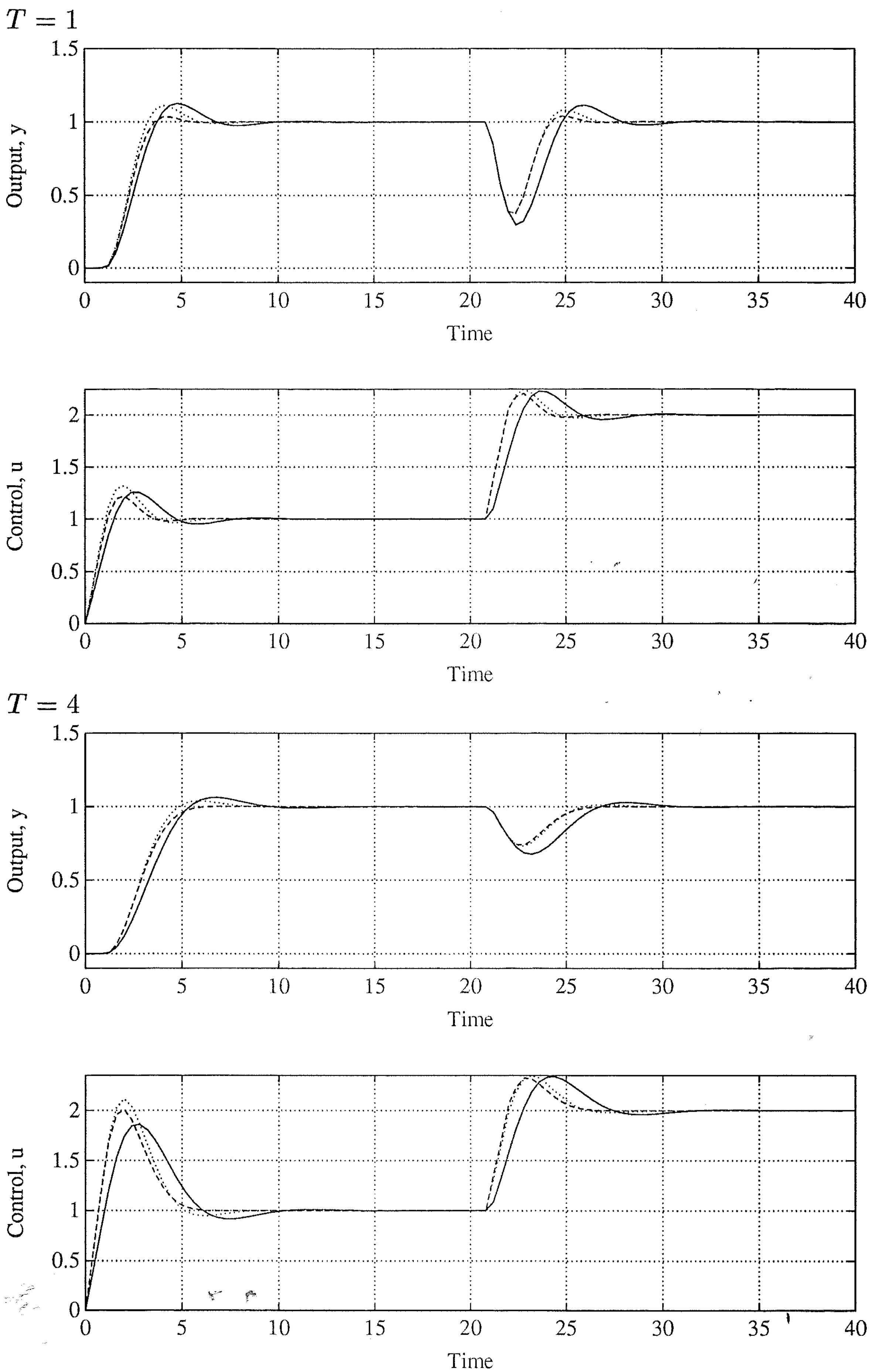



**Figure 7.2** Simulations of S1 for $T = 1$ and $T = 4$. Controllers designed with 2PM (solid), designed with 3PM (dashed), and with MCM (dotted).

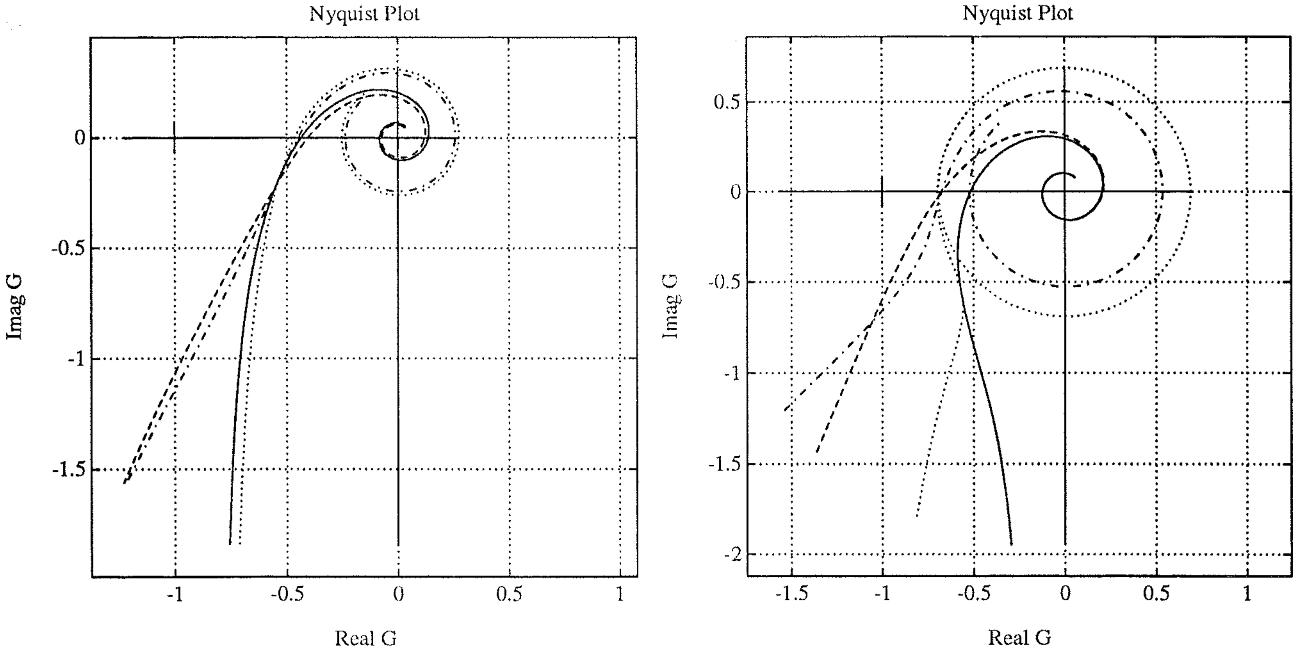



**Figure 7.3** To the left are shown Nyquist curves for loop transfer functions with PI controllers designed with 2PM, $T = 1$ (solid line), $T = 4$ (dashed line) and with PID controllers designed with 3PM, $T = 1$ (dotted line), $T = 4$ (dashed-dotted line). To the right are shown Nyquist curves for loop transfer functions for IAE optimal controllers PI, $T = 1$ (solid line), $T = 4$ (dashed line) and PID $T = 1$ (dotted line), $T = 4$ (dashed-dotted line).

***Nyquist plots*** The Nyquist curves of the loop transfer functions are shown in Figure 7.3. Curves for S1 with PI and PID controllers designed with 2PM and 3PM are shown in one diagram, and curves for S1 with IAE optimal PI and PID controllers are shown in the other.

Notice that the curves designed with 2PM and 3PM have similar shape for different $T$ in the interesting frequency interval. Controllers designed with IAE give rise to very different looking loop transfer functions and are also much closer to $-1$ than the other systems. Other design methods, e.g. Ziegler-Nichols, exhibit similar behaviour. The predicatability of the design is an advantage of 2PM and 3PM.

***Robustness*** The robustness properties are illustrated by the following example. PID controllers were designed for S1 with $T = 2$ with 3PM and with the IAE criterion. Figure 7.4 shows the time responses for the nominal system (solid), for the system with $T = 1.5$ (dashed) and $T = 2.5$ (dotted) for the same controller.

As can be seen from Figure 7.4, the IAE optimal solutions are somewhat faster, but has very poor robustness. For the IAE optimal design $M_s = 3.25$, which is the reason for its poor robustness. With 3PM we have control of the robustness of the closed loop.

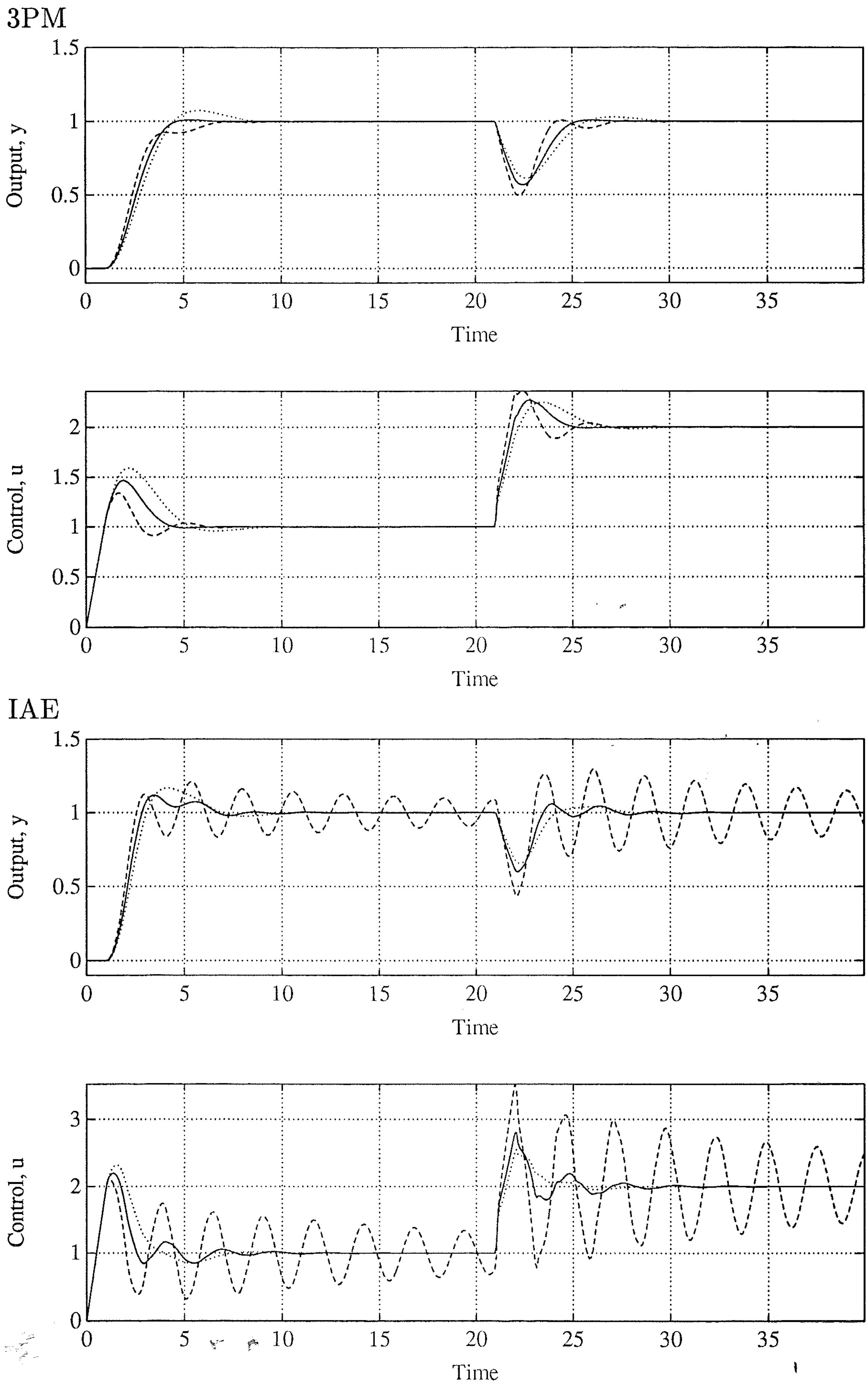



**Figure 7.4** Time responses for S1 for the nominal system $T = 2$ (solid), for the system with $T = 1.5$ (dashed), and $T = 2.5$ (dotted), controlled by the controller for the nominal system. In the figure controllers designed with 3PM and designed with IAE.

***Design with IAE*** As a comparison, controllers for S1 have also been designed with the IAE criterion. As can be seen from Table 7.2 the IAE optimal controllers give quite high $M_s$ values, which causes poor robustness. The time responses with IAE controllers are found in Figure 7.5

Note that the responses of the different systems look very different, e.g., compare the set point responses with PI control with the responses of controllers designed with 2PM in Figure 7.2.

**Table 7.2** Results of IAE designs and simulations of S1

| Case | $k$ | $T_i$ | $T_d$ | $\zeta_0$ | $M_s$ | IE | IAE |
|---|---|---|---|---|---|---|---|
| 1a PI | 0.9840 | 1.6440 | | 0.3292 | 2.1835 | 1.6707 | 1.6707 |
| 1a PID | 1.4350 | 1.1390 | 0.4820 | 0.1633 | 3.2964 | 0.7937 | 1.0449 |
| 1b PI | 3.8603 | 2.4678 | | 0.2082 | 3.6681 | 0.6393 | 0.8729 |
| 1b PID | 5.1446 | 1.6130 | 0.3986 | 0.2959 | 3.1719 | 0.3135 | 0.4200 |

***Design with MPI*** In Figure 7.6 the time responses for S1 with $T = 4$ are shown when the plant is controlled by a PI controller designed with MPI. As comparisons time responses are also shown for the system controlled with a controller designed with 2PM, and an IAE optimal controller. In this case 2PM used the exact calculation to find the IE optimal controller with specified $M_s = 2$. The numerical results of the design are found in Table 7.3.

**Table 7.3** Results of design with MPI of S1

| Case | $k$ | $T_i$ | $\omega_0$ | $\zeta_0$ | IE | IAE |
|---|---|---|---|---|---|---|
| MPI | 2.6583 | 3.1883 | 1.1467 | 0.4901 | 1.1994 | 1.1994 |
| 2PM | 2.0243 | 1.9423 | 0.7380 | 0.4996 | 0.9595 | 1.2549 |
| IAE | 3.8603 | 2.4678 | 1.2637 | 0.2082 | 0.6393 | 0.8729 |

The MPI design gives a somewhat faster response to load disturbances than the 2PM controller, without overshoot. To get substantially faster response we need to increase $M_s$. The IAE controller gives $M_s = 3.67$, and gives s oscillatory response. The high $M_s$ value indicates poor robustness of the system. If a system with no overshoot is wanted, it can be obtained by using a smaller $M_s$ in the design.

Because of the time delay in the process the closed loop system does not react on a set point change until 1 time unit after the change. This explains the sharp edges in the control signal.

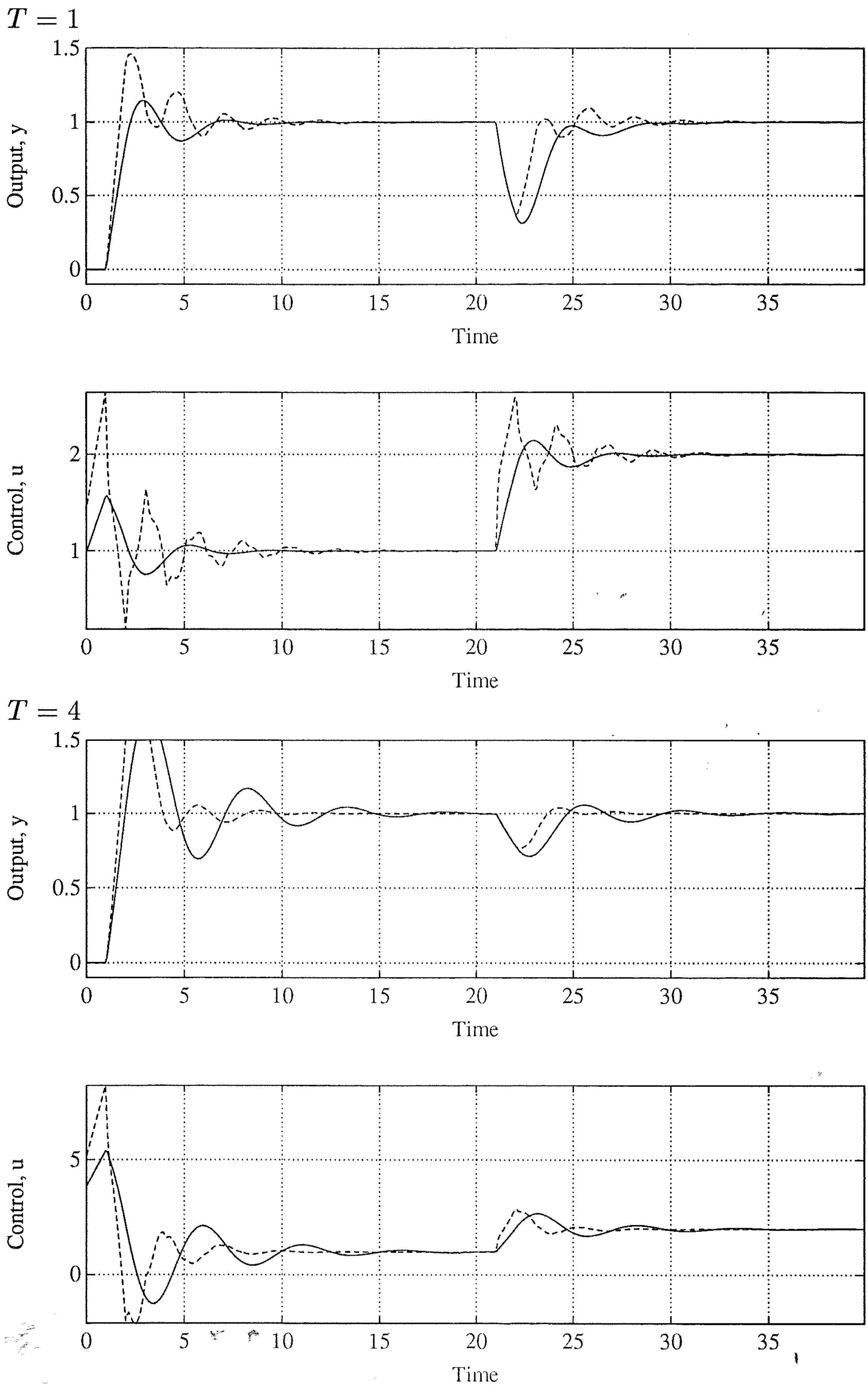



**Figure 7.5** Time response of S1 controlled by IAE optimal controllers, simulations for $T = 1$ and $T = 4$. Responses with PI controllers are marked with solid lines, and PID controllers with dashed lines.

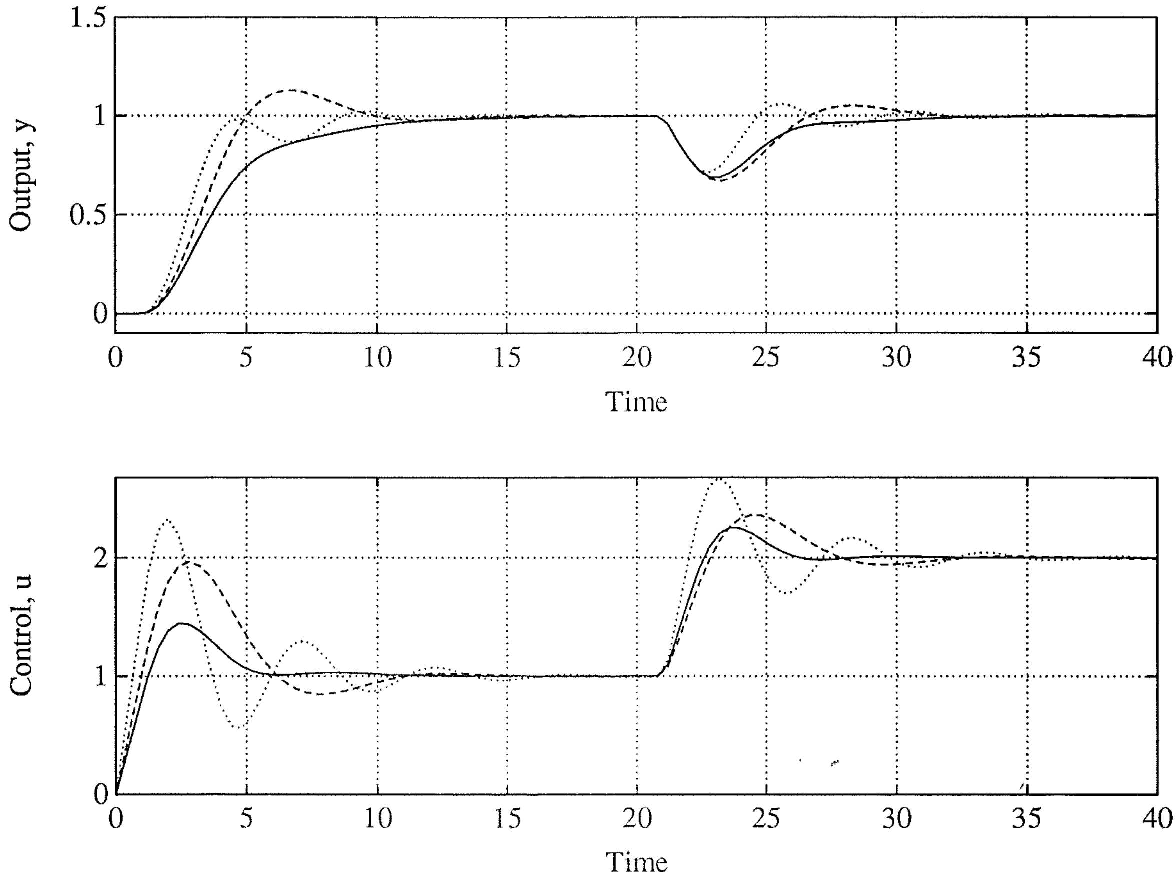



**Figure 7.6** Time responses of S1 controlled by PI controllers. Design methods were MPI (solid line), 2PM (dashed line), and IAE (dotted line).

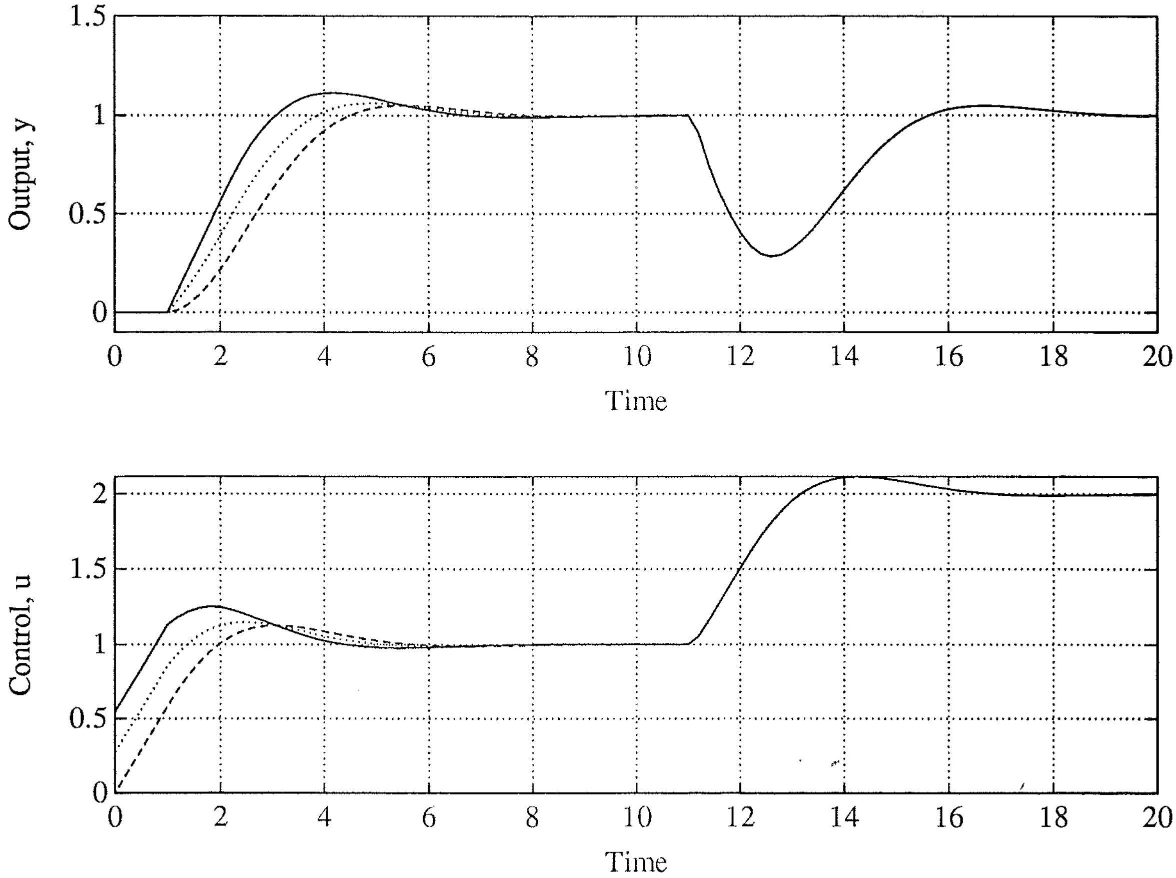



**Figure 7.7** Set point and load responses for $\beta = 1$ (solid line), $\beta = 0$ (dashed line), and $\beta = 0.5$ (dotted line).

***Set point weighting*** Large overshoots in the set point responses can be avoided by using set point weighting. As an example S1 with $T = 1$ is designed by maximizing $k_i$ for $\zeta_0 = 0.6$. The parameter $\beta$ has now been chosen according to the method proposed in Chapter 5, which gives $\beta = 0.5$. The responses to set point and load changes for $\beta = 0$, 0.5, and 1 are shown in Figure 7.7.

**System S2** – $G(s) = 1/(1+s)^n$

***Design with 2PM, 3PM, and MCM*** In this section the system S2 will be investigated. Designs have been made with 2PM, 3PM, and MCM. In 2PM $M_s = 2.0$, in 3PM $\alpha_0 = 1$ and $M_s = 2.0$, and in MCM $M_{s1} = 1.8$ and $M_{s2} = 2.0$ have been used. Responses to set point and load changes when the system is controlled by PI and PID controllers are found in Figure 7.8. Two cases are shown, a) $n = 4$ and b) $n = 8$. Table 7.4 gives a summary of the simulations.

**Table 7.4** Results of designs and simulations of S2

| Case | $k$ | $T_i$ | $T_d$ | $\omega_0$ | $\zeta_0$ | IE | IAE |
|---|---|---|---|---|---|---|---|
| 2a 2PM | 0.6914 | 1.8411 | | 0.5163 | 0.3580 | 2.6628 | 3.8548 |
| 2a 3PM | 1.8790 | 2.6363 | 0.6560 | 0.8126 | 0.3293 | 1.4030 | 1.7141 |
| 2a MCM | 2.0981 | 1.9402 | 0.9159 | 0.7646 | 0.4317 | 0.9247 | 1.5158 |
| 2b 2PM | 0.3975 | 2.9365 | | 0.2356 | 0.4156 | 7.3875 | 10.5254 |
| 2b 3PM | 0.8134 | 4.5403 | 1.1676 | 0.3393 | 0.3965 | 5.5818 | 7.0043 |
| 2b MCM | 0.8476 | 3.9823 | 1.5683 | 0.3114 | 0.5066 | 4.6983 | 6.8461 |

The relative advantage of using PID over PI control is larger in case a) than in case b). Apart from introducing a derivative part in the controller, we get more than twice as large controller gain in the PID case as in the PI case. The peak error is diminished less in case b) than in case a) when going from PI to PID control. We also get larger overshoots with MCM than with 3PM. Their IAE values do not differ significantly.

***Robustness*** As a test of the robustness of the design, the controller for a system with $n$ equal lags was used for systems with $n-1$ and $n+1$ equal lags. The results of the simulations are shown in Figure 7.9 for the case $n = 8$.

***Design with IAE and MPI*** Controllers for S2 were also designed with the MPI and IAE methods. To compute IAE optimal controllers, the simulation program Simnon with OPTA was used, see [Glad, 1974]. The accuracy of the results obtained from OPTA is limited. For these design cases only the first decimal will be the same for different initial guesses of the controller parameters in the optimization. The results of the computations are presented in Table 7.5. Time responses are shown in Figure 7.10. In this case the PI controllers designed with MPI and IAE behave rather similarly. The PI controller designed with MPI gives in this case a smaller peak value of the load response than the one designed with 2PM. This shows that IAE design can produce controllers with relatively low $M_s$ values.

**Table 7.5** Results of IAE and MPI designs and simulations of S2

| Case | $k$ | $T_i$ | $T_d$ | $\omega_0$ | $\zeta_0$ | $M_s$ | IE | IAE |
|---|---|---|---|---|---|---|---|---|
| 2a IAE PI | 1.660 | 4.183 | | 0.747 | 0.183 | 2.779 | 2.520 | 2.796 |
| 2a MPI | 1.209 | 4.001 | | 0.693 | 0.291 | 2.000 | 3.308 | 3.308 |
| 2a IAE PID | 3.537 | 1.776 | 1.460 | 1.440 | 0.205 | 2.627 | 0.502 | 0.796 |
| 2b IAE PI | 0.723 | 5.793 | | 0.318 | 0.284 | 2.174 | 8.011 | 8.622 |
| 2b MPI | 0.694 | 6.393 | | 0.325 | 0.308 | 2.000 | 9.207 | 9.207 |
| 2b IAE PID | 1.046 | 4.392 | 2.631 | 0.514 | 0.232 | 2.232 | 4.201 | 5.313 |

As can be seen, systems with PI controllers designed with MPI and IAE behave similarly. IAE optimal PI and PID controllers have rather high $M_s$ values. In some cases the IAE design can be acceptable, but in other cases the design yields a far too large $M_s$. With 2PM and 3PM we may get a little slower system, but we can always get well damped and robust systems.

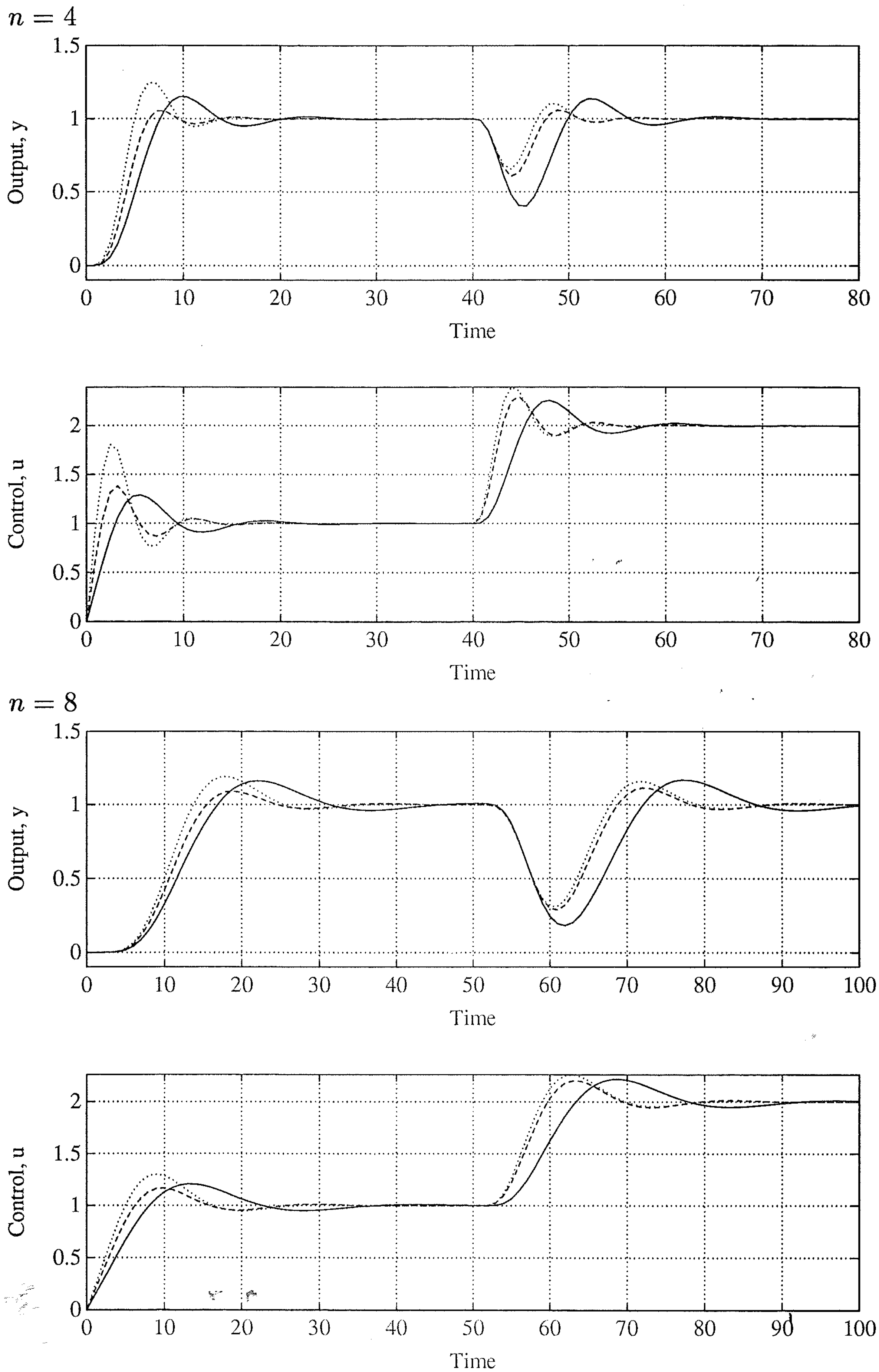



**Figure 7.8** Simulations of S2, with $n = 4$, and with $n = 8$. Controllers designed with 2PM (solid), designed with 3PM (dashed), and with MCM (dotted).

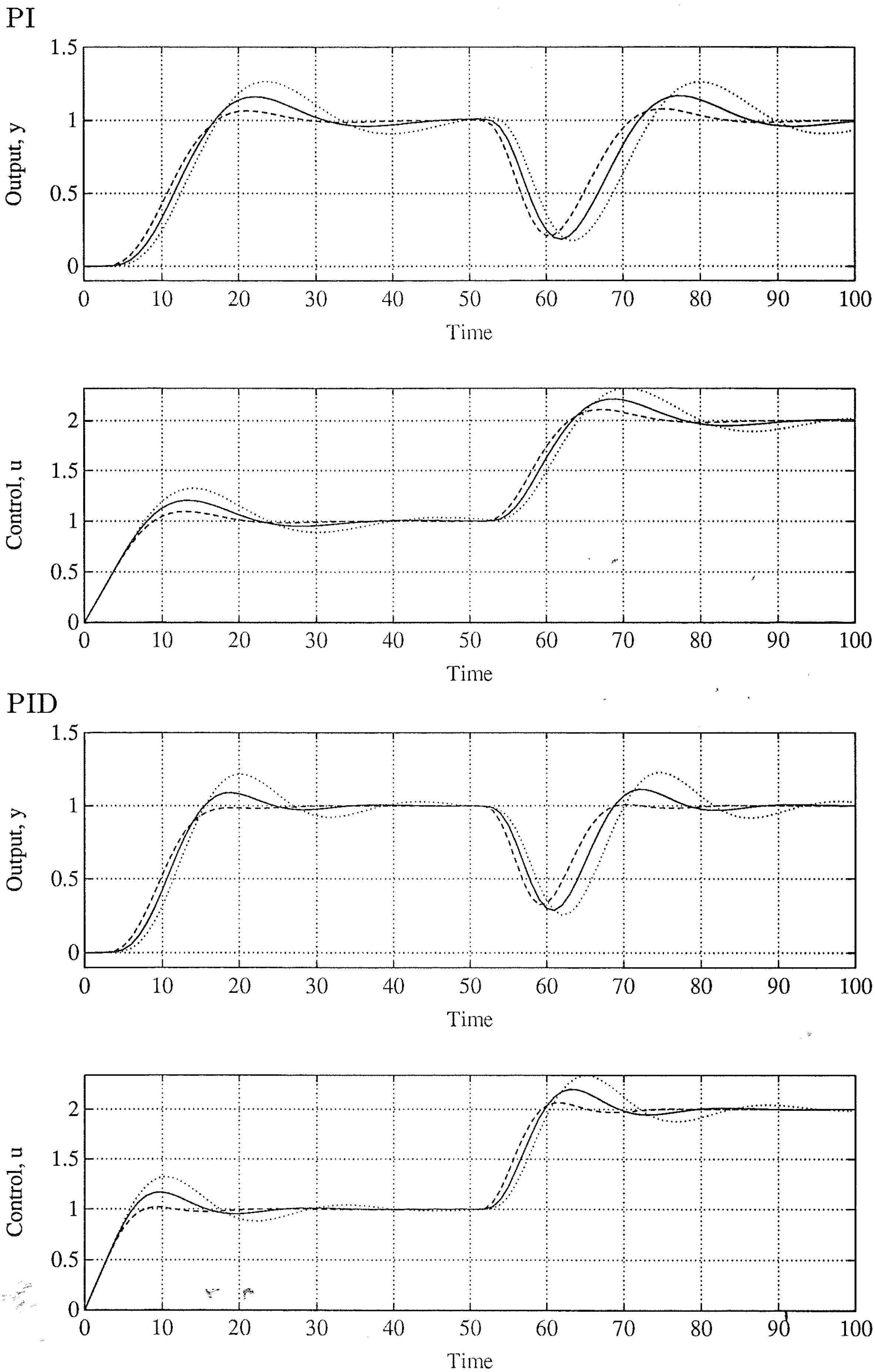



**Figure 7.9** The time responses for perturbed systems, for PI and PID control, $n = 8$ (solid line), $n = 7$ (dashed line), and $n = 9$ (dotted line).

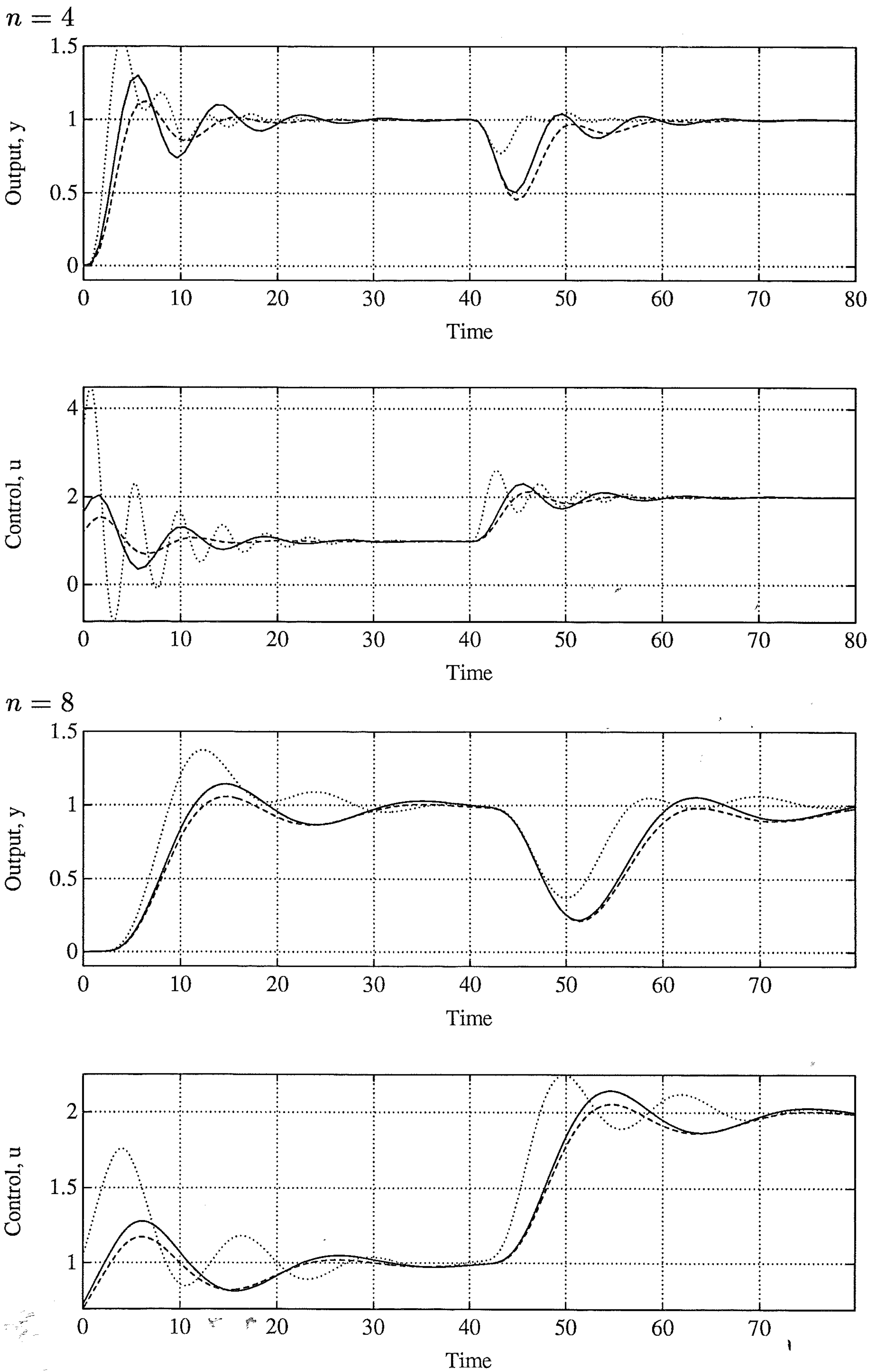



**Figure 7.10** Time responses of S2 for systems with $n = 4$ and with $n = 8$. PI controllers designed with IAE (solid), designed with MPI (dashed), and PID controllers designed with IAE (dotted).

**System S3** – $G(s) = 1/(1+s)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)$

***Design with 2PM, 3PM, and MCM*** In this section the system S3 will be investigated. Designs have been made with 2PM, 3PM, and MCM. In 2PM $M_s = 2.0$, in 3PM $\alpha_0 = 1$ and $M_s = 2.0$, and in MCM $M_{s1} = 1.8$ and $M_{s2} = 2.0$ have been used. Responses to set point and load changes when the system is controlled by PI and PID controllers are found in Figure 7.11. Two cases are shown, a) $\alpha = 0.1$ and b) $\alpha = 0.2$. Table 7.6 gives a summary of the simulations.

**Table 7.6** Results of designs and simulations of S3

| Case | $k$ | $T_i$ | $T_d$ | $\omega_0$ | $\zeta_0$ | IE | IAE |
|---|---|---|---|---|---|---|---|
| 3a 2PM | 9.1136 | 0.3225 | | 8.5867 | 0.3576 | 0.0354 | 0.0377 |
| 3a 3PM | 175.1807 | 0.0793 | 0.0264 | 43.0934 | 0.4100 | 0.0005 | 0.0005 |
| 3a MCM | 179.2038 | 0.0569 | 0.0283 | 38.0878 | 0.4261 | 0.0003 | 0.0004 |
| 3b 2PM | 3.8868 | 0.5580 | | 3.9843 | 0.3583 | 0.1436 | 0.1651 |
| 3b 3PM | 27.6872 | 0.2836 | 0.0810 | 11.6362 | 0.3947 | 0.0102 | 0.0106 |
| 3b MCM | 28.7862 | 0.2017 | 0.0893 | 10.3902 | 0.4242 | 0.0070 | 0.0096 |

When $\alpha = 0.1$ the system has poles in $-1$, $-10$, $-100$, and $-1000$, and is dominated by the pole in $-1$. Because of this, the design method gives high values of the controller coefficients and $\omega_0$. A similar system, although of second order, is examined in Example 4.22, where we see that the $\omega_0$ in the design is proportional $(1 + 1/\alpha)$. We also observe that in this case the derivative part improves the output signals very much. The IAE values are reduced drastically when going from PI to PID control, in case a) almost two orders of magnitude. The bandwidth is increased by a factor 5. Increasing the complexity of the controller made it possible to increase the loop gain significantly. The control signal and noise sensitivity will be high. In this case it is the noise level and limitations on the control signal which determine if PID control can be used.

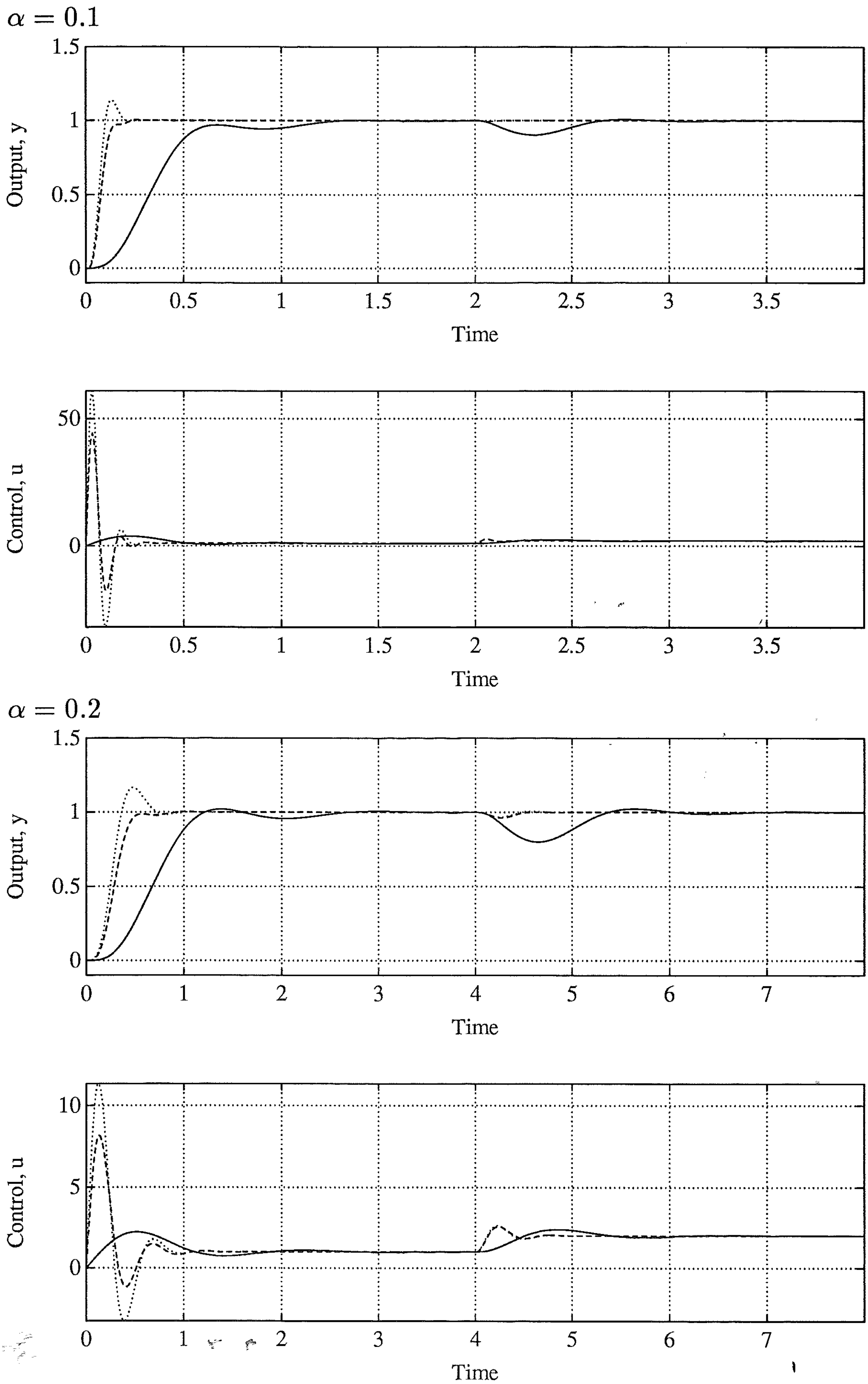



**Figure 7.11** Simulations of S3, with $\alpha = 0.1$, and with $\alpha = 0.2$. Controllers designed with 2PM (solid), designed with 3PM (dashed), and with MCM (dotted).

**System S4** – $G(s) = (1 - \alpha s)/(1 + s)^3$

***Design with 2PM, 3PM, and MCM*** In this section the system S4 will be investigated. Designs have been made with 2PM, 3PM, and MCM. In 2PM $M_s = 2.0$, in 3PM $\alpha_0 = 1$ and $M_s = 2.0$, and in MCM $M_{s1} = 1.8$ and $M_{s2} = 2.0$ have been used. Responses to set point and load changes when the system is controlled by PI and PID controllers are found in Figure 7.12. Two cases are shown, a) $\alpha = 0.2$ and b) $\alpha = 2$. Table 7.7 gives a summary of the simulations.

**Table 7.7** Results of designs and simulations of S4

| Case | $k$ | $T_i$ | $T_d$ | $\omega_0$ | $\zeta_0$ | IE | IAE |
|---|---|---|---|---|---|---|---|
| 4a 2PM | 0.8548 | 1.5872 | | 0.6902 | 0.3522 | 1.8569 | 2.6155 |
| 4a 3PM | 2.5537 | 2.1050 | 0.5073 | 1.1522 | 0.3437 | 0.8243 | 0.9490 |
| 4a MCM | 2.8653 | 1.4906 | 0.6764 | 1.0763 | 0.4216 | 0.5203 | 0.8197 |
| 4b 2PM | 0.3054 | 1.6578 | | 0.4255 | 0.5533 | 5.4291 | 7.6456 |
| 4b 3PM | 0.5446 | 2.4298 | 0.6181 | 0.6794 | 0.6865 | 4.4616 | 5.7751 |
| 4b MCM | 0.5141 | 2.2289 | 0.6025 | 0.5359 | 0.6676 | 4.3358 | 6.0569 |

This is a system with right half plane zeros. In case a) the IAE drops significantly when we go from PI to PID control. The zero is in this case located far from the origin and the other poles. The system is almost a pure third order system. As can be seen from Figure 7.12 the non minimum phase character of the is not so obvious. In case b) we have a zero close to the origin. It can be shown that poles and zeros in the right half plane set limits on what can be done by a controller. The decrease of IAE when PID control is used instead of PI control is much less than in case a). The overshoot in the set point response in a) is much less for the 3PM controller than for the MCM controller. From the time responses in Figure 7.12 we clearly see the characteristics of a non minimum phase system. In case b) there is not much reason for using PID control.

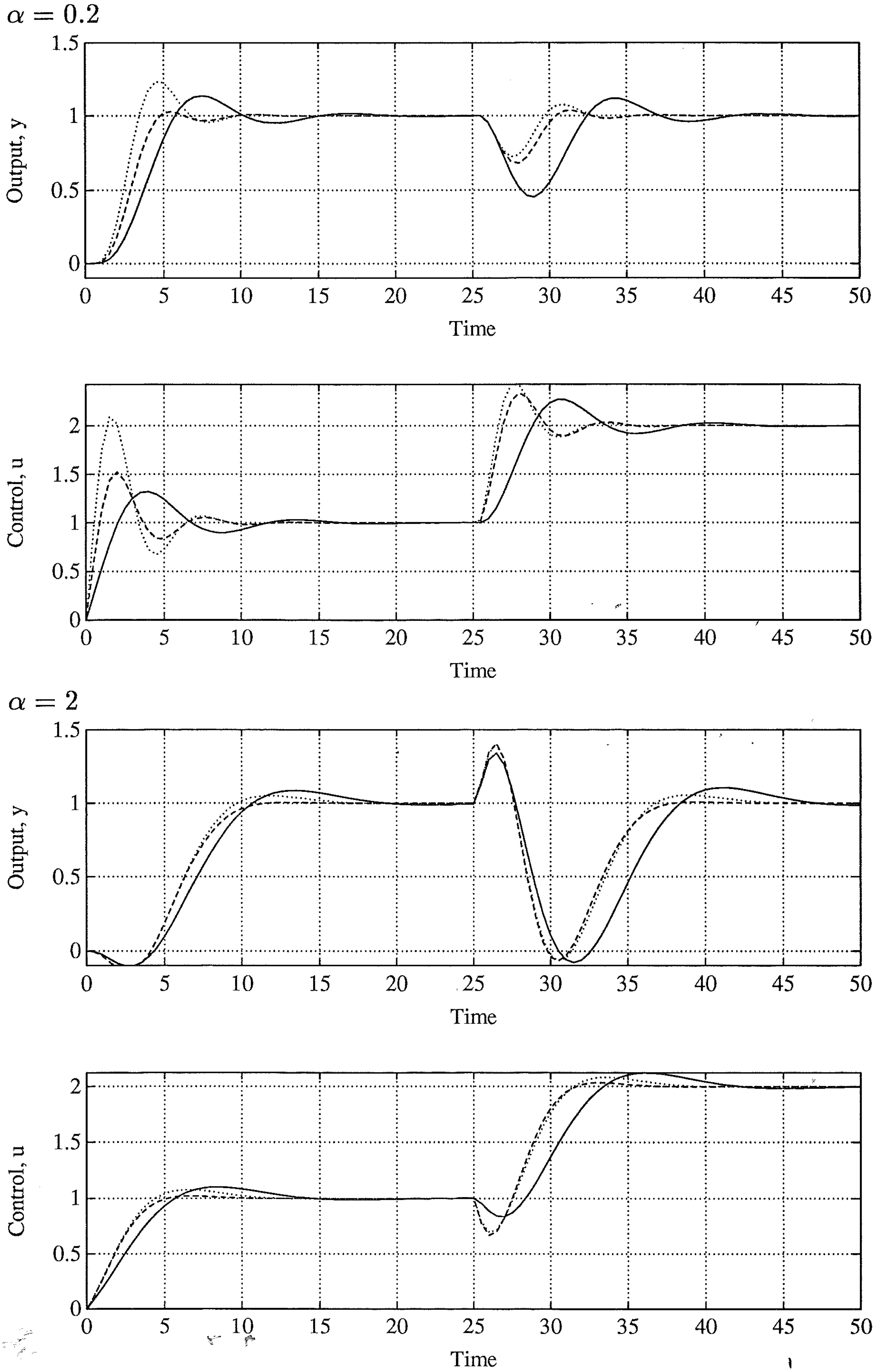



**Figure 7.12** Simulations of S4, with $\alpha = 0.2$, and with $\alpha = 2$. Controllers designed with 2PM (solid), designed with 3PM (dashed), and with MCM (dotted).

**System S5** – $G(s) = \alpha/(\alpha + s)(s^2 + 2\zeta_p s + 1)$

***Design with 2PM and 3PM*** In this case it proved to be almost impossible to get a decent PI or PID controller with 2PM or MCM, therefore only PID design with 3PM are shown. The design parameter for 2PM was $M_s = 3.7$, and for 3PM $\alpha_0 = 0.4$ and $M_s = 2.2$. The time responses of the controlled system are shown in Figure 7.13. In this example $\zeta_p = 0.3$ and in a) $\alpha = 0.3$ and in b) $\alpha = 3.3$. The difference between the two methods is where on the negative real axis the pole of the plant is located.

**Table 7.8** Results of designs and simulations of S5

| Case | $k$ | $T_i$ | $T_d$ | $\omega_0$ | $\zeta_0$ | IE | IAE |
|---|---|---|---|---|---|---|---|
| 5a 2PM | −0.0070 | −0.0160 | | 0.7789 | 0.1231 | 2.2792 | 8.3935 |
| 5a 3PM | 1.7330 | 2.7300 | 1.0069 | 1.5831 | 0.1790 | 1.5753 | 1.6031 |
| 5b 2PM | 0.2848 | 0.6513 | | 1.0384 | 0.0725 | 2.2865 | 7.4451 |
| 5b 3PM | 5.9079 | 1.7202 | 0.5829 | 3.0760 | 0.2795 | 0.2912 | 0.2920 |

This is a case where PID control is absolutely necessary. The case a) is the most difficult to handle. The controller coefficients become negative. In case b) the pole on the negative real axis is so far away that we almost have a second order system, which can easily be handled by a PID controller.

The designs gave for a) controller zeros with relative damping $\zeta = 0.82$ and $\omega = 0.60$ and for b) and $\zeta = 0.86$ and $\omega = 1.00$. It seems reasonable to try to compensate the oscillating poles with complex zeros. We also note that the design method does not try to cancel the oscillating poles. In these cases PID control gives good results. Note that the control signal is oscillating.

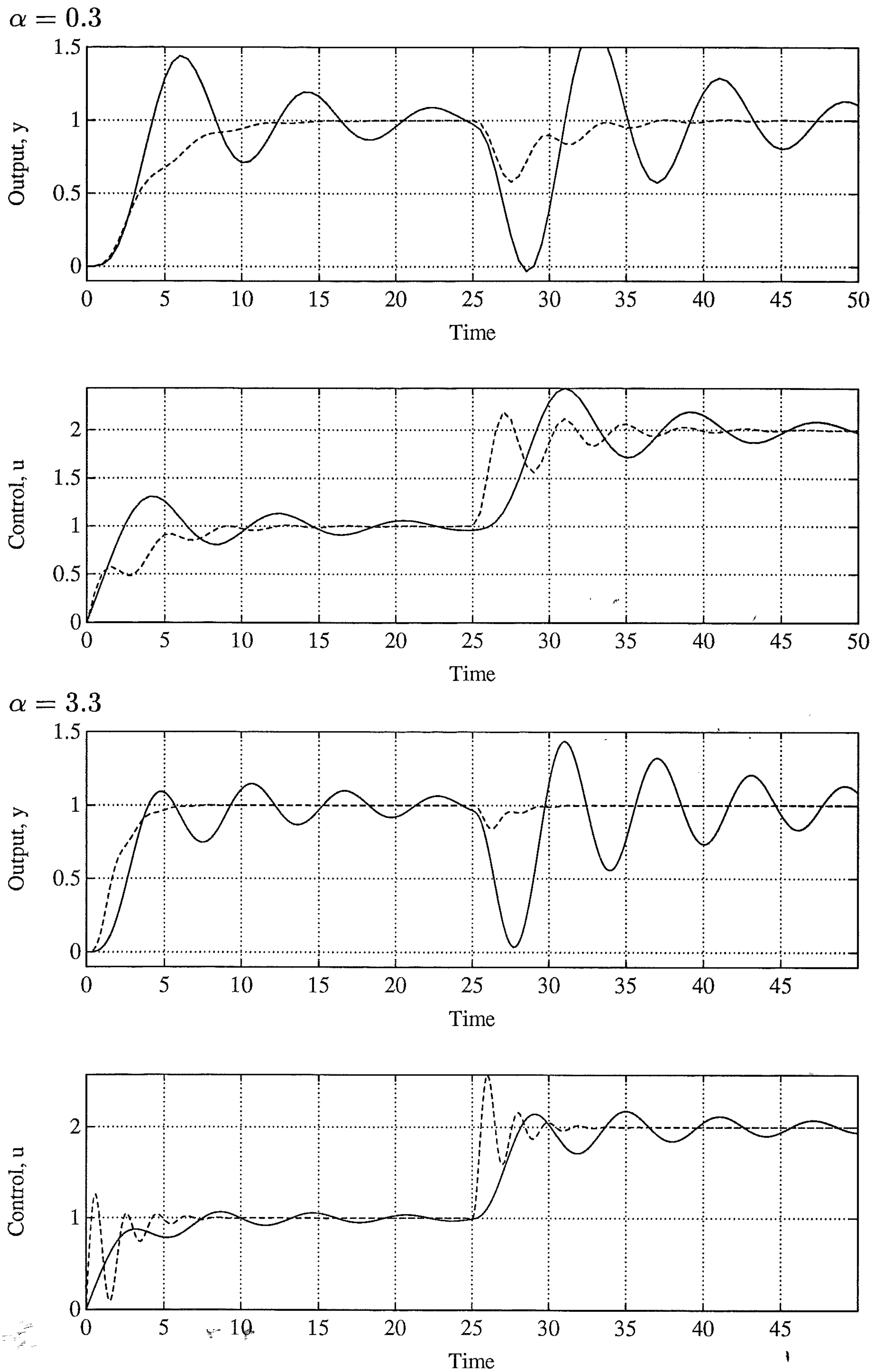



**Figure 7.13** Simulations of S5, with $\alpha = 0.3$, and with $\alpha = 3.3$. Controllers designed with 2PM (solid), designed with 3PM (dashed).

**System S6** – $G(s) = e^{-s}/s(1+Ts)$

***Design with 2PM, 3PM, and MCM*** In this section the system S6 will be investigated. Designs have been made with 2PM, 3PM, and MCM. In 2PM $M_s = 2.0$, in 3PM $\alpha_0 = 1$ and $M_s = 2.0$, and in MCM $M_{s1} = 1.8$ and $M_{s2} = 2.0$ have been used. Responses to set point and load changes when the system is controlled by PI and PID controllers are found in Figure 7.14. Two cases are shown, a) $T = 1$ and b) $T = 4$. Table 7.9 gives a summary of the simulations.

**Table 7.9** Results of designs and simulations of S6

| Case | $k$ | $T_i$ | $T_d$ | $\omega_0$ | $\zeta_0$ | IE | IAE |
|---|---|---|---|---|---|---|---|
| 6a 2PM | 0.2869 | 7.8612 | | 0.3427 | 0.5390 | 27.4033 | 29.6996 |
| 6a 3PM | 0.6716 | 5.3015 | 0.8221 | 0.7436 | 0.7069 | 7.8943 | 7.9257 |
| 6a MCM | 0.6525 | 4.8978 | 0.8150 | 0.5994 | 0.6600 | 7.5067 | 8.0256 |
| 6b 2PM | 0.1435 | 18.7309 | | 0.1507 | 0.4264 | 130.5490 | 140.3130 |
| 6b 3PM | 1.0303 | 6.9627 | 2.0152 | 0.5561 | 0.7126 | 6.7579 | 6.8322 |
| 6b MCM | 0.9764 | 5.8697 | 1.9689 | 0.4386 | 0.5160 | 6.0114 | 6.7485 |

This is a case where derivative action improves the controller performance radically.

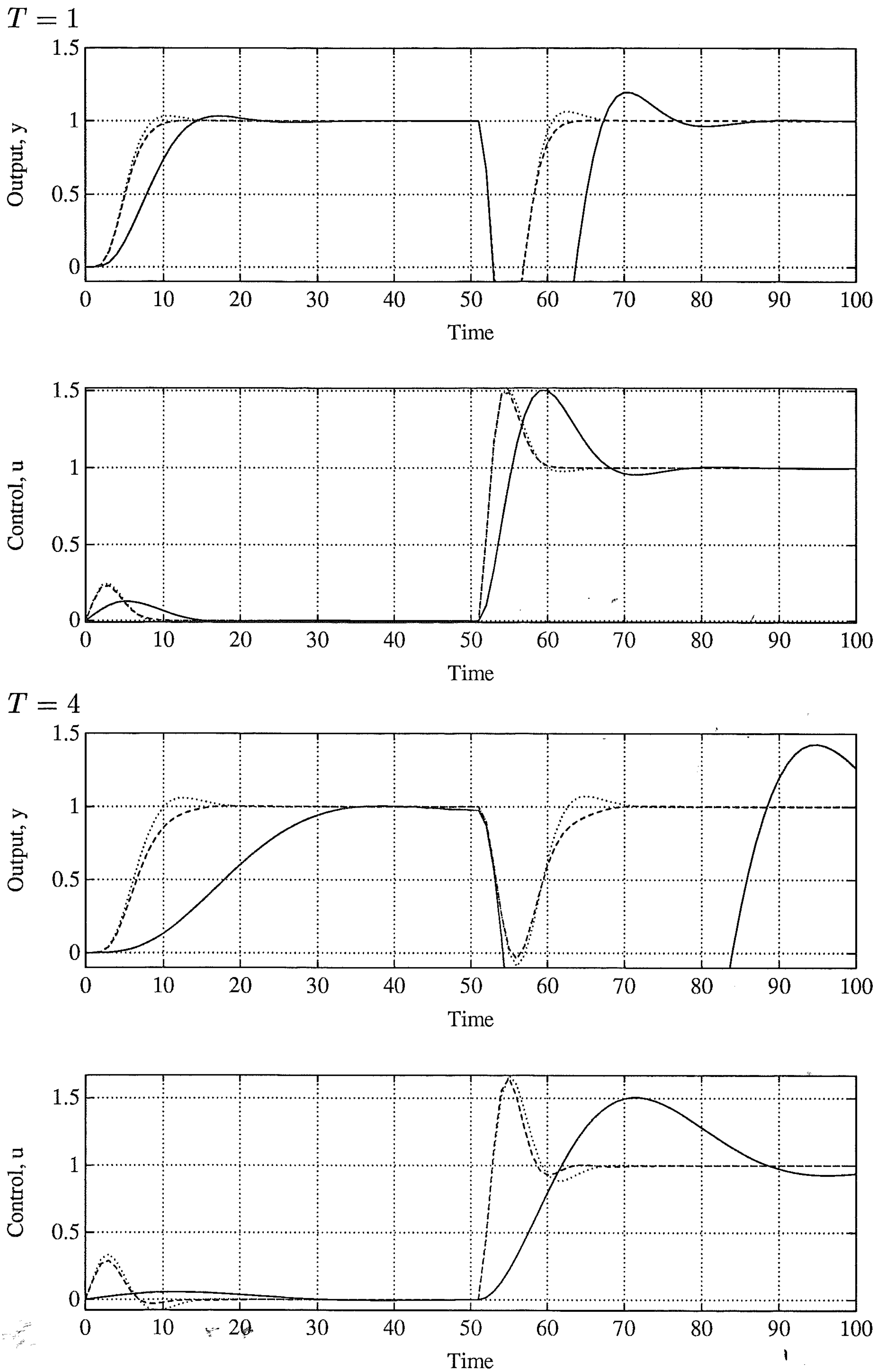



**Figure 7.14** Simulations of S6, with $T = 1$, and with $T = 4$. Controllers designed with 2PM (solid), designed with 3PM (dashed), and with MCM (dotted).

## 7.2 Achievable performance

A natural question when using PI and PID controllers is: What can we achieve in a closed loop, e.g., in terms of settling time after a load disturbance? It is often desired to express the achievable performance in terms of the plant dynamics. Unfortunately there is no simple answer to this question. It is the dynamics of the plant that puts limits on the achievable performance. To illustrate the problem, the settling times of the controlled systems, S1 to S6 were crudely estimated from Figures 7.2, 7.8, 7.11, 7.12, 7.13, and 7.14. The results are collected in Table 7.10.

S3 has open loop time constant $\approx 1$, S2a has open loop time constant $\approx 4$, and S2b has open loop time constant $\approx 8$. The settling time for S3 is approximately the same as the open loop time constant or less, while for S2 the settling time is about 4 to 6 times the open loop time constant. Thus, to determine the settling time of the closed system, more information than the apparent time constant of the open system is needed. From Table 7.10 we see that the performance is improved dramatically for

- processes with one very dominating pole, and little time delay
- integrating processes
- resonant processes.

These are application areas where much is gained by using derivative action. From Table 7.10 we also see that the product $t_s\omega_0 \approx 11$ for all processes, except for the resonant and the one with widely spread poles. This can be used to assess the settling time from data obtained from the design.

**Table 7.10** Settling times for the test processes

| | PI | | PID | | |
|---|---|---|---|---|---|
| | $t_s$ | $\omega_0 t_s$ | $t_s$ | $\omega_0 t_s$ | $t_{s,\text{PID}}/t_{s,\text{PI}}$ |
| S1a | 10 | 11.2 | 7 | 11.9 | 0.70 |
| S1b | 10 | 8.5 | 7 | 9.9 | 0.70 |
| S2a | 25 | 12.9 | 15 | 12.2 | 0.60 |
| S2b | 50 | 11.8 | 40 | 13.6 | 0.80 |
| S3a | 0.7 | 6.0 | 0.1 | 4.3 | 0.14 |
| S3b | 2 | 8.0 | 0.5 | 5.8 | 0.25 |
| S4a | 20 | 13.3 | 10 | 11.5 | 0.50 |
| S4b | 25 | 10.6 | 15 | 10.2 | 0.60 |
| S5a | 50 | 39 | 15 | 24.0 | 0.30 |
| S5b | 100 | 104 | 5 | 15.0 | 0.05 |
| S6a | 40 | 13.7 | 15 | 11.1 | 0.38 |
| S6b | 80 | 12.1 | 20 | 11.1 | 0.25 |

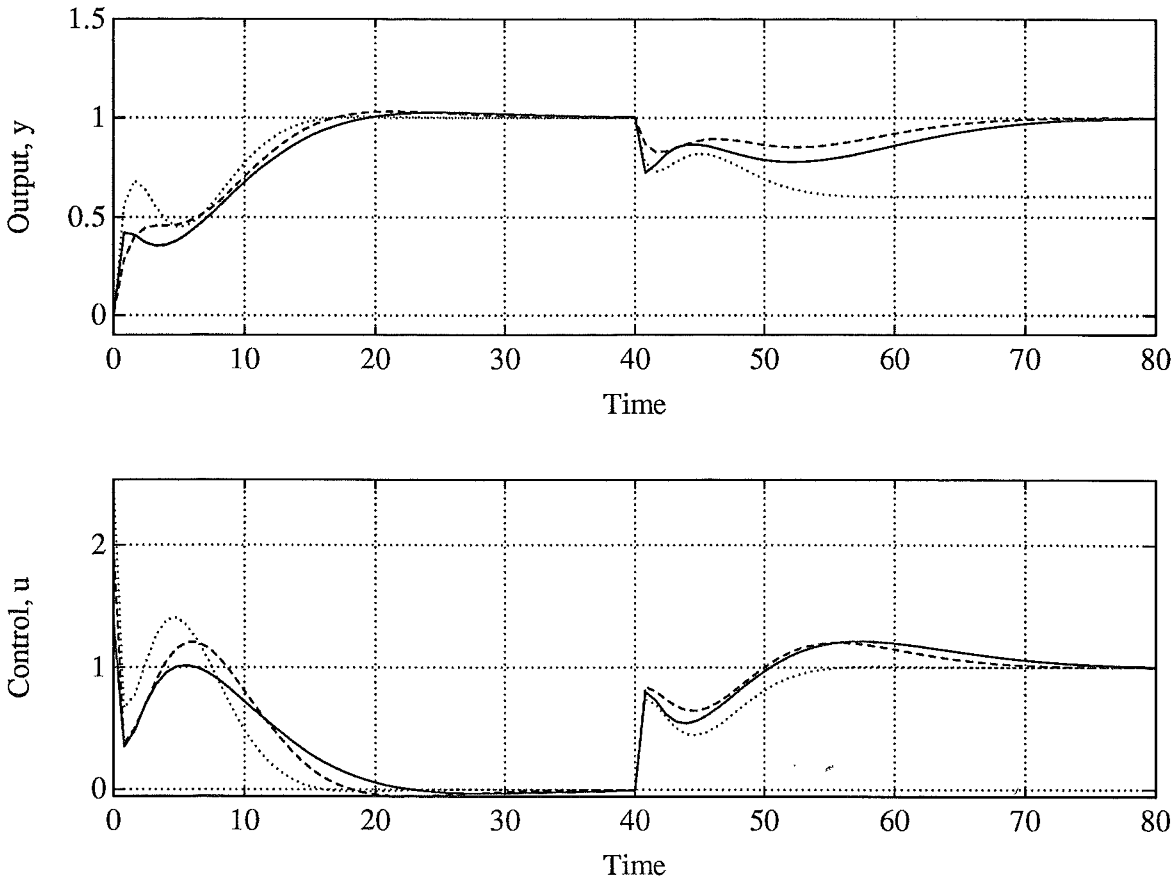



**Figure 7.15** Control of system (7.5). PI control (solid line), PID control (dashed line), and PD control (dotted line).

## 7.3 Special systems

Controllers for a number of special systems have also been designed with methods presented in this thesis. In these cases different values of $M_s$ and $\alpha_0$ have had to be chosen.

### A model for a motor drive

The model

$$G(s) = \frac{(s^2 + 2\zeta_1\omega_1 s + \omega_1^2)}{s(s^2 + 2\zeta_2\omega_2 s + \omega_2^2)} \tag{7.5}$$

is often used as a description of a flexible motor drive. Typically $\omega_1 = 0.3\omega_2$. In this example we set $\zeta_{1,2} = 0.1$ and $\omega_2 = 1$. This model has a pole excess of 1. This may cause rather low values of $M_s$ for a system controlled by a PI or PID controller, which do not have much significance for the closed loop behaviour. For this reason a PID design with direct specification of $\alpha_0$ and $\zeta_0$ was tried. PI and PD controllers were also designed with a fixed $\zeta_0$. In this case $\zeta_0 = 0.8$ and $\alpha_0 = 1$. The simulation of the system is shown in Figure 7.15. This is a case when PID control is not successful. More complicated controllers are required.

### A standard test example

Consider

$$G_T(s) = \frac{\omega_p^2(1-s)}{(1+s)(s^2+2\omega_p\zeta_p s+\omega_p^2)}e^{-s\tau} \tag{7.6}$$

with $\omega_p = 15$, $\zeta_p = 0.5$, and $\tau = 0.4$. The process $G_T(s)$ has been given in [M'Saad, 1991] as a benchmark process for adaptive control strategies. As have been pointed out in [Åström *et al.*, 1991] this is a process dominated by the dynamics

$$G(s) = \frac{(1-s)}{(1+s)}e^{-s\tau}. \tag{7.7}$$

Hence one can assume that a PI controller will work well, but a PID controller will probably have too much high frequency gain to do well. Controllers were designed with 2PM with $M_s = 2.0$ and with MCM with $M_{s1} = 1.7$ and $M_{s2} = 1.9$. The design parameters were chosen to get no overshoot in the set point responses. Due to little amplitude roll-off the controller can tolerate very little derivative action before the condition on $M_{s2}$ is met. Hence the results of control with PI and PID control is similar. The parameters for PI control were $k = 0.3618$ and $T_i = 0.9794$, and for PID control $k = 0.3611$, $T_i = 1.0590$, and $T_d = 0.0568$. The PI controller has IAE = 1.7519 and the PID controller has IAE = 1.8490, and we see that very little is gained by going from PI to PID control. The peak error is somewhat smaller in the case of PID control.

The results of the designs are found in Figure 7.16. Design results from, e.g., [Lundh, 1991] produce similar time responses.

### Systems with long time delays

Systems with long time delays are often controlled with little or no derivative action. Therefore we will often use MCM as design method to have direct control over the derivative action.

***The standard system*** S1 Let $\omega_1$ and $\omega_2$ be the gain and phase crossover frequencies, i.e., $|L(i\omega_1)| = 1$ and $\arg L(i\omega_2) = -\pi$. The behaviour of the loop transfer function for frequencies round the bandwidth of the system is most important for the closed loop system, and $\omega_0$ and $\omega_b$ are normally in the frequency interval $[\omega_1, \omega_2]$. One way of getting insight into the behaviour of a PID controller is to compare the magnitudes of the derivative and the integral parts to each other for the frequency $\omega_0$. Introduce $Q$ defined by

$$Q = \frac{k\omega_0 T_d}{k/(\omega_0 T_i)} = \omega_0^2 T_i T_d. \tag{7.8}$$

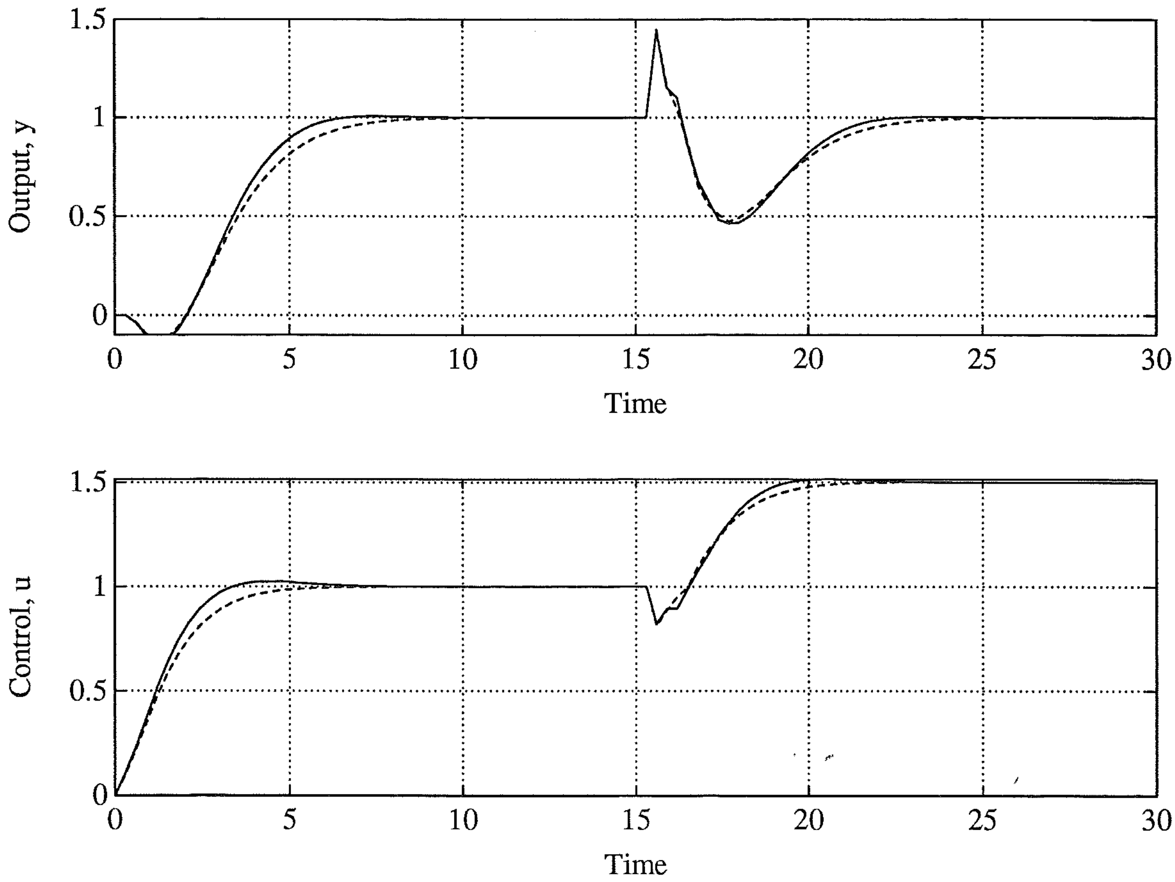



**Figure 7.16** The process $G_T$ controlled by PI (solid line) and PID (dashed line) controllers.

PID controllers were designed for

$$G(s) = \frac{e^{-sL}}{s+1} \tag{7.9}$$

with MCM with $M_{s1} = 1.8$ and $M_{s2} = 2.0$. The quantity $Q$ is shown in Figure 7.17. As can be seen the relative influence of the derivative part decreases when $L$ is increased. Similar results were obtained for systems of type S2, where the influence of the derivative action decreases when more dynamics is added. Design has been carried out with the standard PID controller $G_{\text{PID}}(s)$. In the case of (7.9) this means that $k_d$ is limited to $k_d \leq 1 - 1/M_{s2}$. The elbow in Figure 7.17 occurs when $k_d = 1 - 1/M_{s2}$, i.e., when the high frequency gain limits the robustness. Derivative action makes most difference for systems with relatively little dead time. The performance of a controlled system is always better with PID control than with PI control.

***Pure time delay*** A pure time delay has the transfer function

$$G(s) = e^{-sL}. \tag{7.10}$$

In [Haalman, 1965] it is claimed that a pure time delay is best controlled by

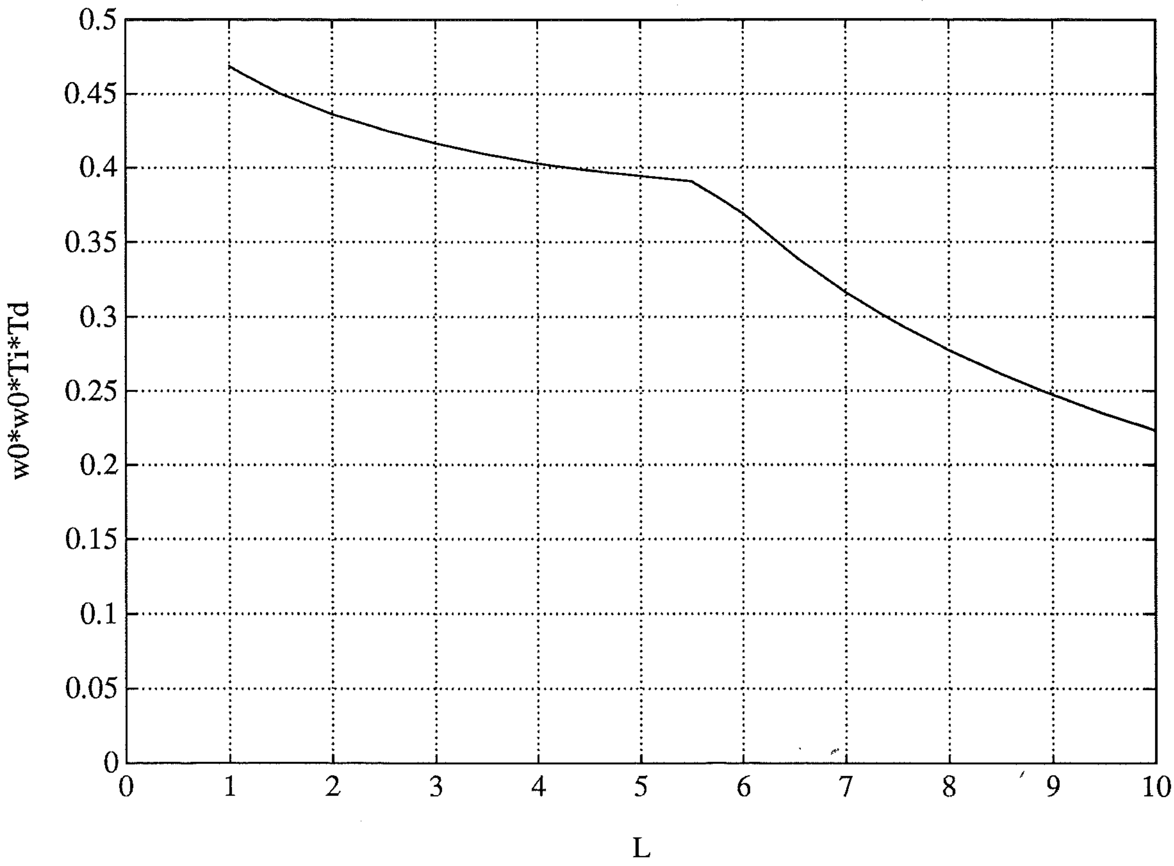



**Figure 7.17** The ratio between integral and derivative action.

an integrating controller

$$G_c(s) = \frac{2}{3Ls}. \tag{7.11}$$

We design a PI controller with 2PM with $M_s = 1.917$, which is the value obtained by Haalman's design. The controller parameters were $k = 0.235$ and $T_i = 0.288$. As can be seen from Figure 7.18 responses to both set point changes and load disturbances are better with the 2PM design. The IAE values were 2.125 for Haalman and 1.525 for 2PM. Note that the control signals have almost identical magnitude. The system designed with 2PM has faster response, but the same overshoot.

The process (7.10) is special because the process gain is constant for all frequencies. If a PID controller is used it is necessary to limit the derivative gain. The design method will not be able to handle the design of a PID controller for a pure time delay, since the loop transfer function

$$L(s) = k(1 + \frac{1}{sT_i} + sT_d)e^{-s} \tag{7.12}$$

has infinite gain for high frequencies. If the derivative part has a filter the high frequency gain of the loop transfer function is $k(1 + N)$. This means that the Nyquist curve approaches a circle with radius $k(1 + N)$. Thus we

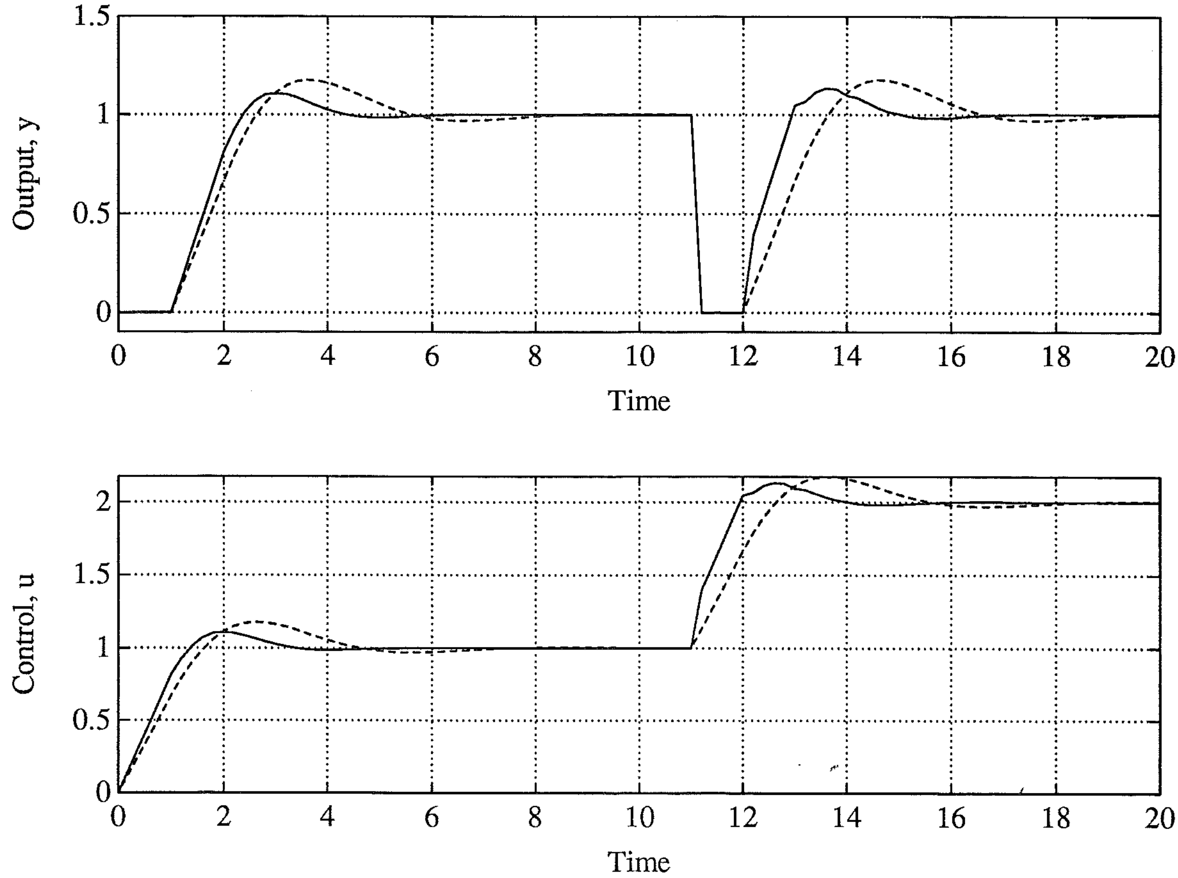



**Figure 7.18** Time response for a pure time delay controlled by a controller designed with 2PM with $M_s = 1.917$ (solid line), and according to Haalman (dashed line).

must have $k(1+N) < 1$ to get stability, and $k(1+N) < 1 - 1/M_s$ to fulfill the robustness criterion. This makes it necessary to take the derivative filter into account in the design. If we design with MCM and specify $M_{s1}$ as in the Haalman case we get $k = 0.235$ for the initial PI controller. The condition on the PID controller gain is $k < 0.478/(1+N)$. If $k_d$ is chosen very small the $k$ and $T_i$ of the PID controller will be approximately the same as for the PI controller and the inequality cannot be fulfilled with controllers from MCM.

***Two lags and time delay*** Another example of systems with long time delays will be studied. This time we choose a process which has more high frequency roll-off. Consider

$$G_L(s) = \frac{e^{-sL}}{(1+sT)^2} \tag{7.13}$$

for $L = 1$ and small values of $T$. PI controllers were designed with 2PM with $M_s = 2.0$ and PID controllers with 3PM with $M_s = 2.0$ and $\alpha_0 = 1.0$. The results of investigations for $L/T = 1$, 5, and 15 are shown in Table 7.11.

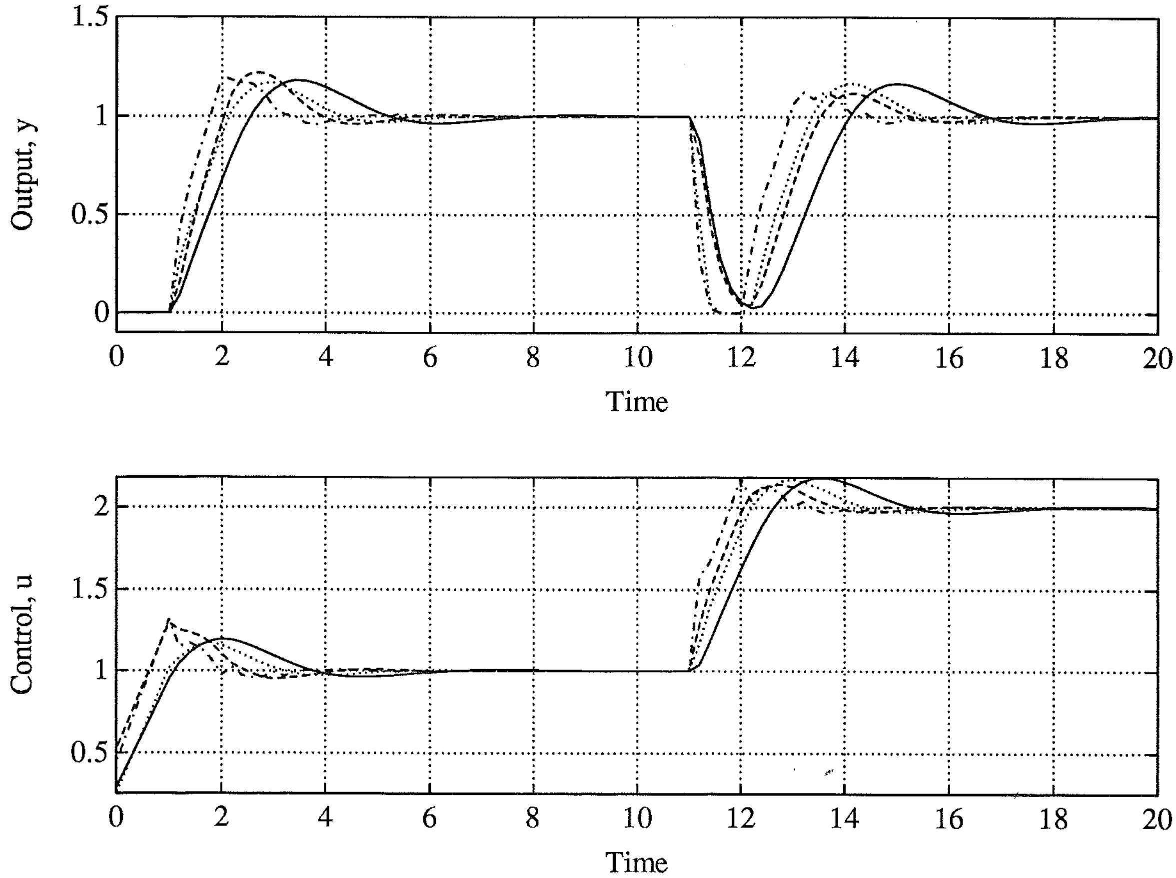



**Figure 7.19** The time responses of $G_L(s)$ for long relative time delays. $L/T = 5$, PI (solid line), $L/T = 5$, PID (dashed line), $L/T = 15$, PI (dotted line), and $L/T = 15$, PID (dashed-dotted line).

**Table 7.11**

| Control | L/T | $k$ | $T_i$ | $T_d$ | $\omega_0$ | $\zeta_0$ | IE | IAE |
|---|---|---|---|---|---|---|---|---|
| 2PM | 1 | 0.6151 | 1.3318 | | 0.6986 | 0.4027 | 2.1652 | 2.9924 |
| 3PM | 1 | 1.3365 | 1.8628 | 0.4679 | 1.1019 | 0.4630 | 1.3938 | 1.5768 |
| 2PM | 5 | 0.2902 | 0.4420 | | 1.3364 | 0.4700 | 1.5231 | 2.0550 |
| 3PM | 5 | 0.5241 | 0.6796 | 0.1773 | 1.9157 | 0.4692 | 1.2966 | 1.5420 |
| 2PM | 15 | 0.2526 | 0.3314 | | 1.6430 | 0.4884 | 1.3120 | 1.7393 |
| 3PM | 15 | 0.4444 | 0.5108 | 0.1363 | 2.3443 | 0.4849 | 1.1493 | 1.3576 |

We see that the relative improvement in IAE in using PID control over using PI control decreases from 0.47 to 0.22 as $L/T$ increases from 1 to 15. The time responses for $L/T = 5$ and 15 are shown in Figure 7.19. The improvement in IAE does not motivate the dramatic increase in high frequency gain when going from PI to PID control.

Better results of controller design for processes with long dead times can be obtained with special dead time controllers. As comparison we will show the design with the PIP controller.

***The PIP controller*** The PIP controller is a dead time controller and is

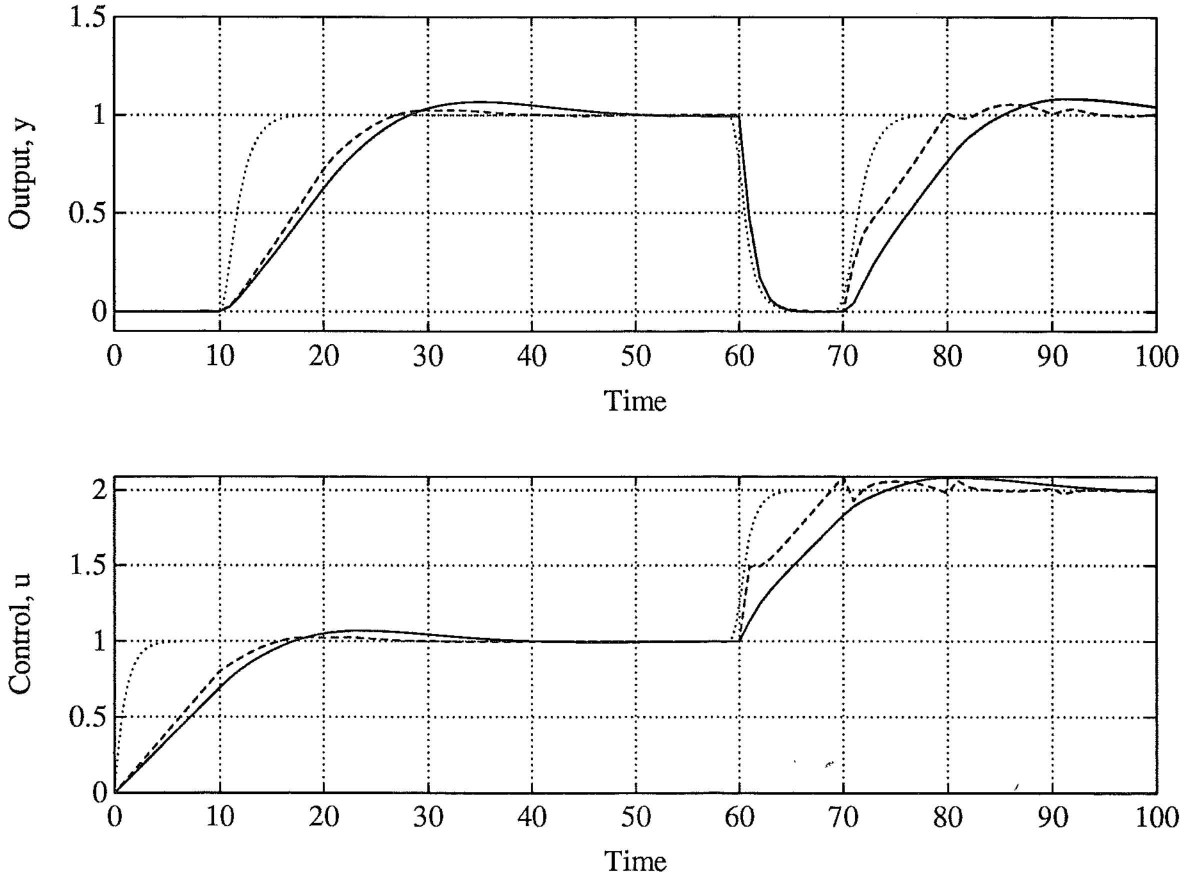



**Figure 7.20** Systems with a PI controller designed with 2PM (solid line), a PID controller designed with 3PM (dashed line), and with the PIP deadtime controller (dotted line).

described in [Hägglund, 1991]. The controller has the transfer function

$$G_{\mathrm{PIP}}(s) = \frac{K(1 + sT_i)}{sT_i + 1 - e^{-sL}}, \tag{7.14}$$

and the time domain relation

$$\begin{aligned} u(t) &= K\left(1 + \frac{1}{pT_i}\right)\left(e(t) - \frac{K_p}{1+pT}[u(t) - u(t-L)]\right) \\ &= K\left(1 + \frac{1}{pT_i}\right)e(t) - \frac{\kappa(1 + pT_i)}{pT_i(1 + p\tau T_i)}[u(t) - u(t-L)]. \end{aligned} \tag{7.15}$$

EXAMPLE 7.3—Control with the PIP controller
The PIP controller was tested on

$$G(s) = \frac{e^{-10s}}{s+1}. \tag{7.16}$$

In Figure 7.20 the PIP controller is compared to PI and PID controllers designed with 2PM with $M_s = 1.8$ and 3PM with $M_s = 1.8$ and $\alpha_0 = 1.0$.

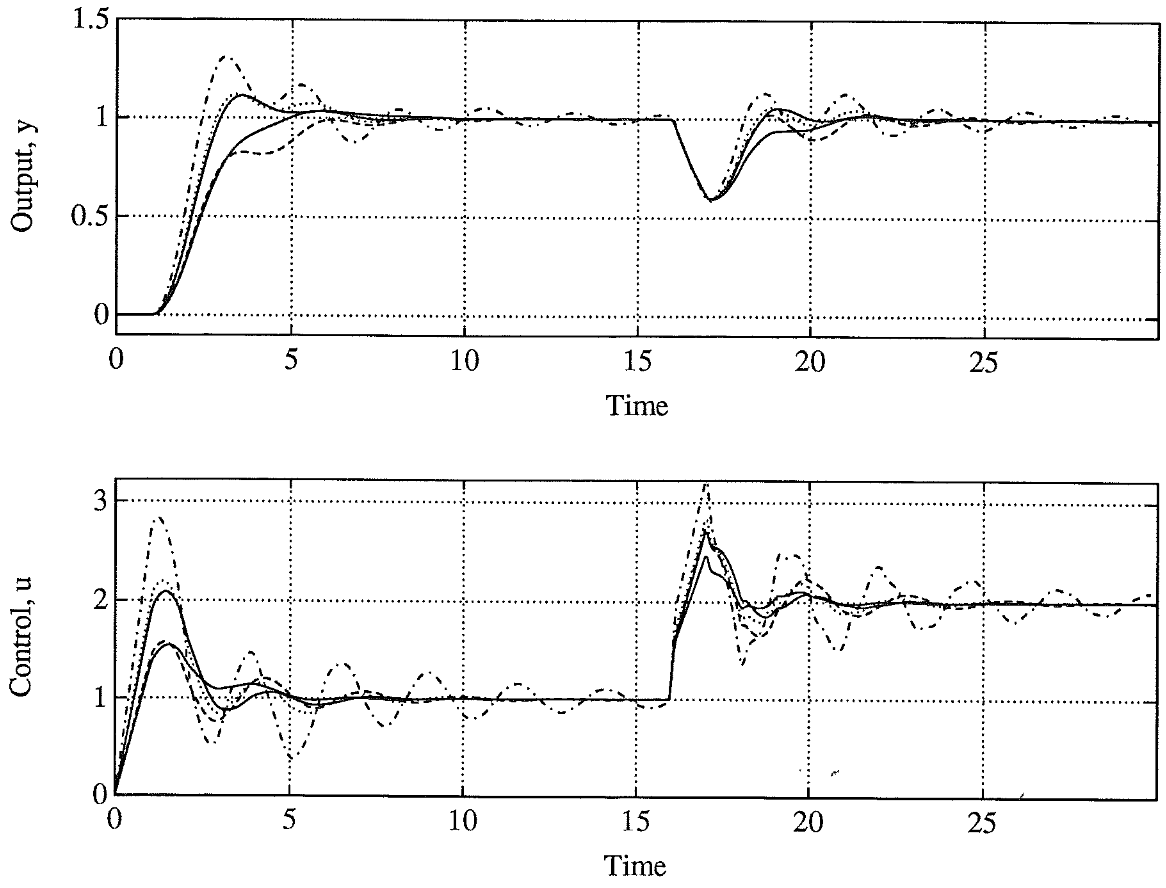



**Figure 7.21** Set point and load responses for S1 with $T = 2$ with controllers designed with ZN (solid line, little step response overshoot), CC (dashed line), IAE (dotted line), ISE (dashed-dotted line), and ITAE (solid line larger overshoot).

Low values of $M_s$ were chosen to get small overshoots. The PIP controller gives the best performance, but a well tuned PID controller comes close. □

## 7.4 Other design methods

In this section a few other design methods will be compared with the methods developed in this thesis.

### The standard design methods

Comparisons of the standard integral criteria can be found in many textbooks and articles, see e.g. [Seborg *et al.*, 1989] and [Miller *et al.*, 1967]. To show how systems behave when controlled by controllers obtained from the standard design rules, we will study S1 with $T = 2$. PID controllers were computed with ZN, CC, IAE, ISE, and ITAE, and the simulated set point and load responses are shown in Figure 7.21.

In this case the IAE and ITAE methods yield very similar responses. As can be seen from the figure a wide variety of behaviour can be obtained from the standard criteria.

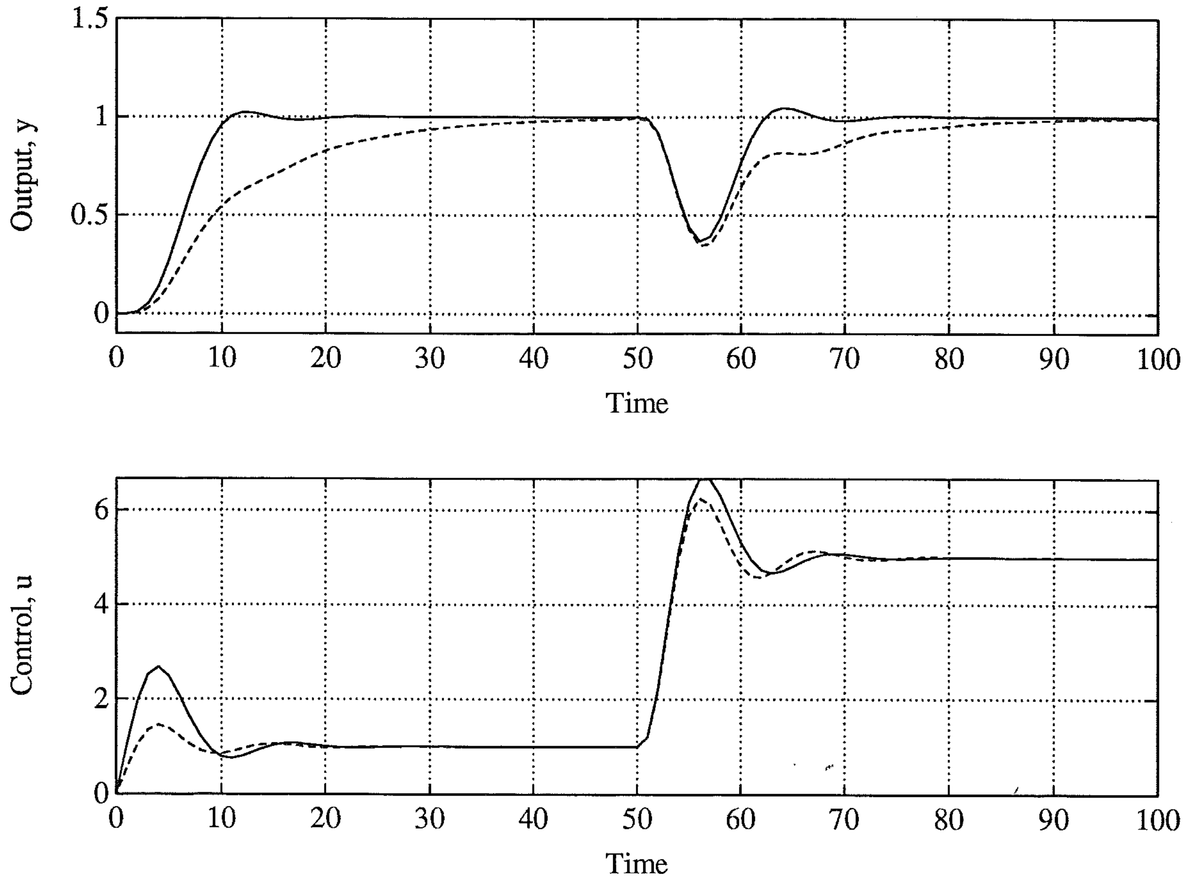



**Figure 7.22** Time responses with controller designed with 3PM (solid line) and with pole cancelling (dashed line).

## Pole cancelling methods

Consider the plant

$$G(s) = \frac{1}{(s+1)^3(10s+1)}. \tag{7.17}$$

PID controllers were designed with two methods. First with 3PM with $M_s = 2.0$ and $\alpha_0 = 1.0$. In the second method $T_i$ and $T_d$ were chosen to cancel two process poles in $-0.1$ and $-1$, this gave $T_i = 11$ and $T_d = 10/11$. The controller gain $k$ was then chosen to get $M_s = 2$. The time responses obtained with controllers designed with the two methods are shown in Figure 7.22.

The 3PM design corresponds to controller zeros in $-0.66$ and in $-0.25$. We get controller zeros between the process poles. The controller gain with 3PM was 5.9 and in the cancellation case 6.3. The main drawback of the cancellation method is the slow recovery from a load disturbance. The control signal becomes larger in the 3PM case. Basing a design method on pole cancelling is not a good idea. The results get even worse if we try to cancel poles closer to the origin. References to discussions of this problem can be found in Chapter 3.

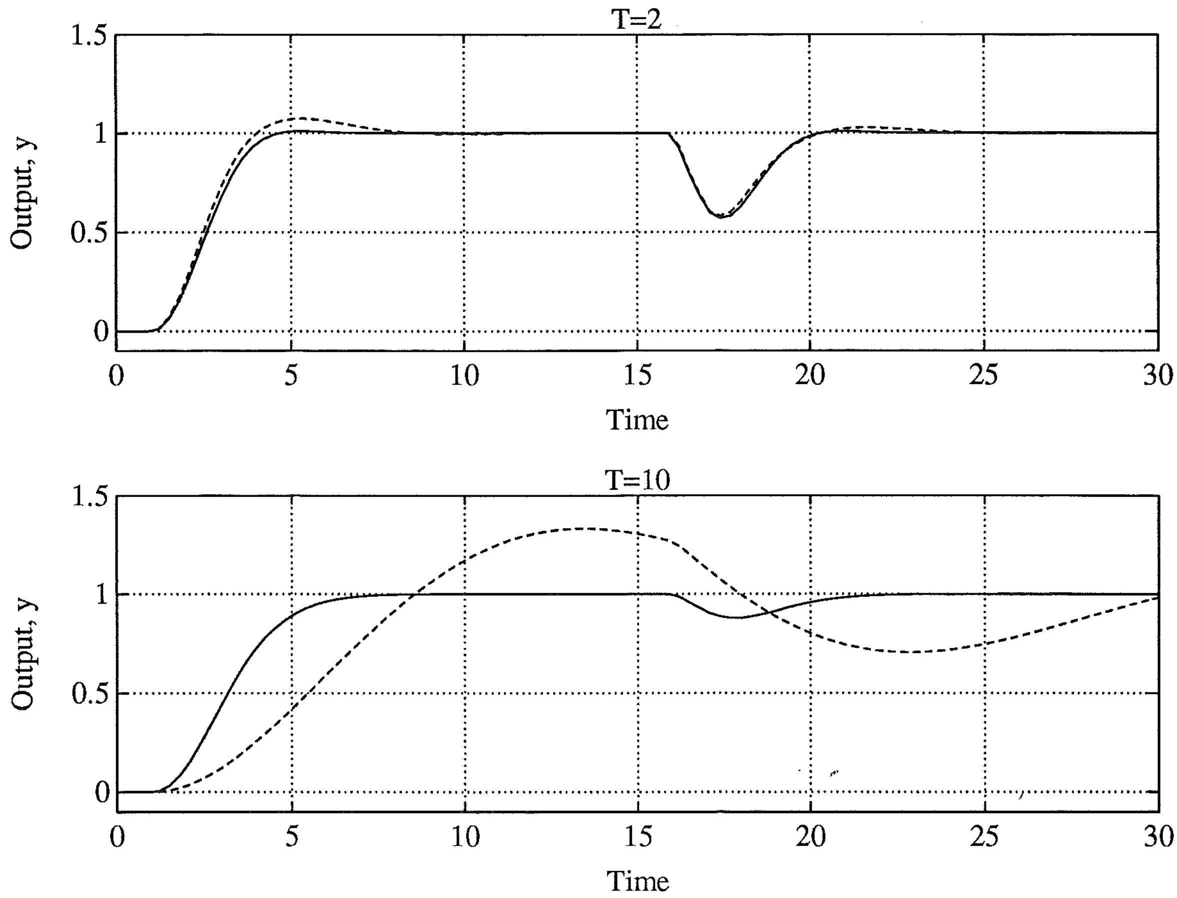



**Figure 7.23** Time responses for $G(s)$ with controllers designed with 3PM (solid line) and Thomas' method (dashed line) for $T = 2$ and $T = 10$. Only the output signals are shown.

## The Thomas method

A method for PID controller tuning is presented in the thesis [Thomas, 1990]. The method is based on knowledge of the frequency response of the plant. PI controller design is not considered. For details of the method, see Thomas' thesis.

The method has been tested on

$$G(s) = \frac{e^{-s}}{1 + sT}, \quad T = 2, 10. \tag{7.18}$$

Controllers were designed with 3PM with $M_s = 2$ and with Thomas' method. The time responses for the closed systems are shown in Figure 7.23. Thomas' method gives good results in many cases, but $T = 10$ is a case where it fails. A short relative time delay $(L/T)$ gives a far too slow system.

## The PIDWIZ method

The American company BST Control delivers a program for hand-held computers (Texas Instruments TI-74), which gives PID controller settings from data obtained from a step response. For further information on PIDWIZ, see [Blickley, 1988].

The following systems were given from BST Control as a test batch for dominant pole design, [Sanathan, 1989]. PID controller settings according to PIDWIZ were also given.

$$G_1(s) = 0.5e^{-10s} \prod_{m=1}^{10} \frac{1}{1+ms} \tag{Z1}$$

$$G_2(s) = 1.5\frac{e^{-2s}}{(1+0.2s)^5} \tag{Z2}$$

$$G_3(s) = 0.75\frac{e^{-10s}}{1+s}. \tag{Z3}$$

The controller parameters obtained with PIDWIZ and with 3PM for $\alpha_0 = 1$ and $M_s = 1.6$ are given in Table 7.12. The 3PM design were used with $M_s = 1.6$ to avoid overshoots.

**Table 7.12** PIDWIZ and 3PM controller settings

| Case | $k$ | $T_i$ | $T_d$ | IE | IAE | $M_s$ |
|---|---|---|---|---|---|---|
| Z1 PIDWIZ | 0.934 | 39.244 | 6.156 | 42.017 | 42.017 | 1.564 |
| Z1 3PM | 0.941 | 32.917 | 8.698 | 34.984 | 35.480 | 1.600 |
| Z2 PIDWIZ | 0.150 | 1.220 | 0.167 | 8.130 | 8.130 | 1.433 |
| Z2 3PM | 0.212 | 1.254 | 0.342 | 5.917 | 5.923 | 1.600 |
| Z3 PIDWIZ | 0.202 | 3.321 | 0.000 | 16.474 | 16.474 | 1.428 |
| Z3 3PM | 0.392 | 4.400 | 1.204 | 11.236 | 11.248 | 1.600 |

As can be seen from Table 7.12 PIDWIZ makes rather conservative designs in the sense that the $M_s$ values of the controlled systems are low. 3PM gave larger values than PIDWIZ of all controller parameters.

In Figures 7.24 and 7.25 are shown time responses for Z1, Z2, and Z3 with 3PM designed controllers (solid lines) and PIDWIZ controllers (dashed lines). In all cases 3PM performed better than PIDWIZ.

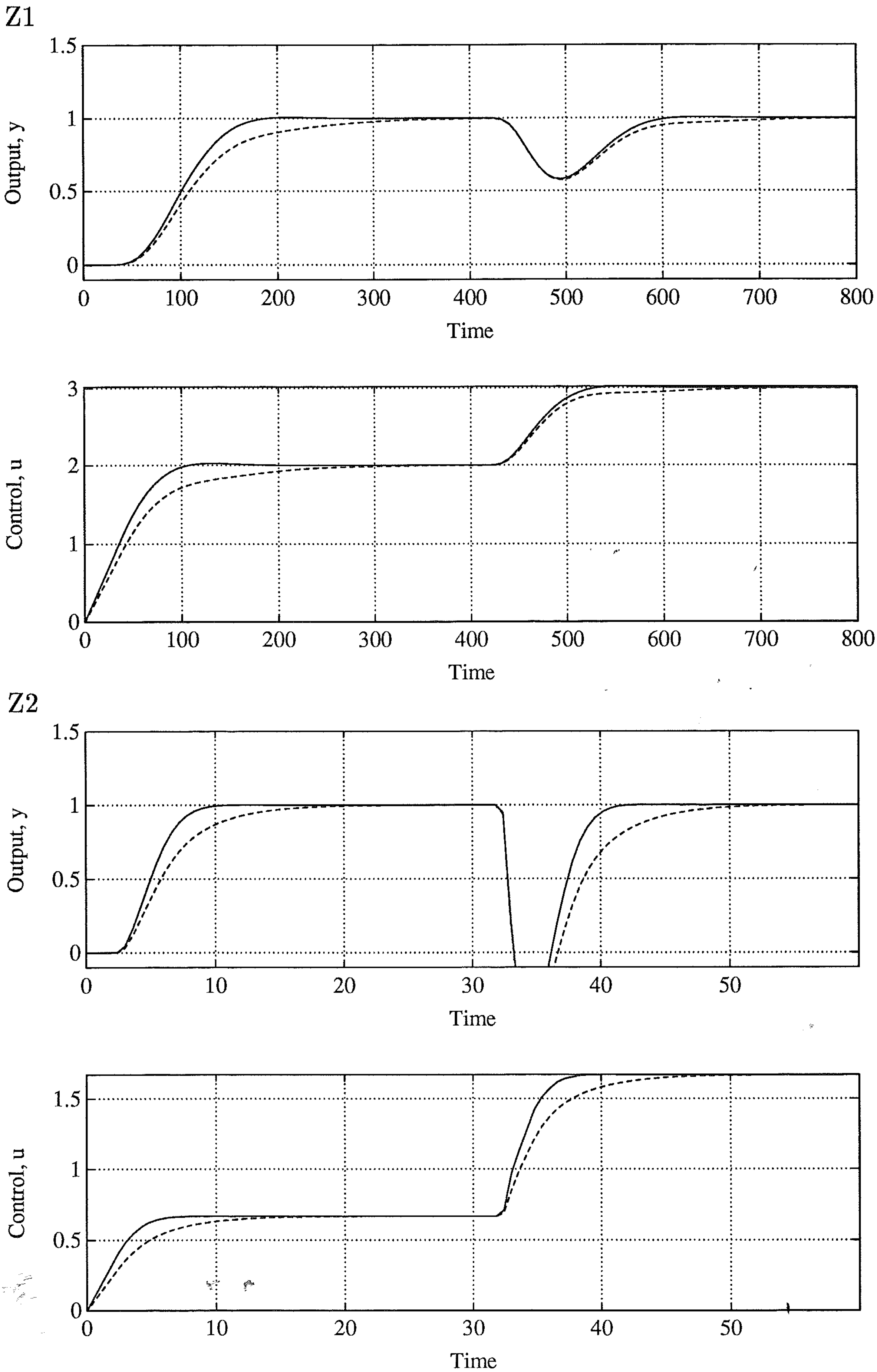



**Figure 7.24** Time response of 3PM design (solid line) and PIDWIZ design (dashed line) for Z1 and Z2.

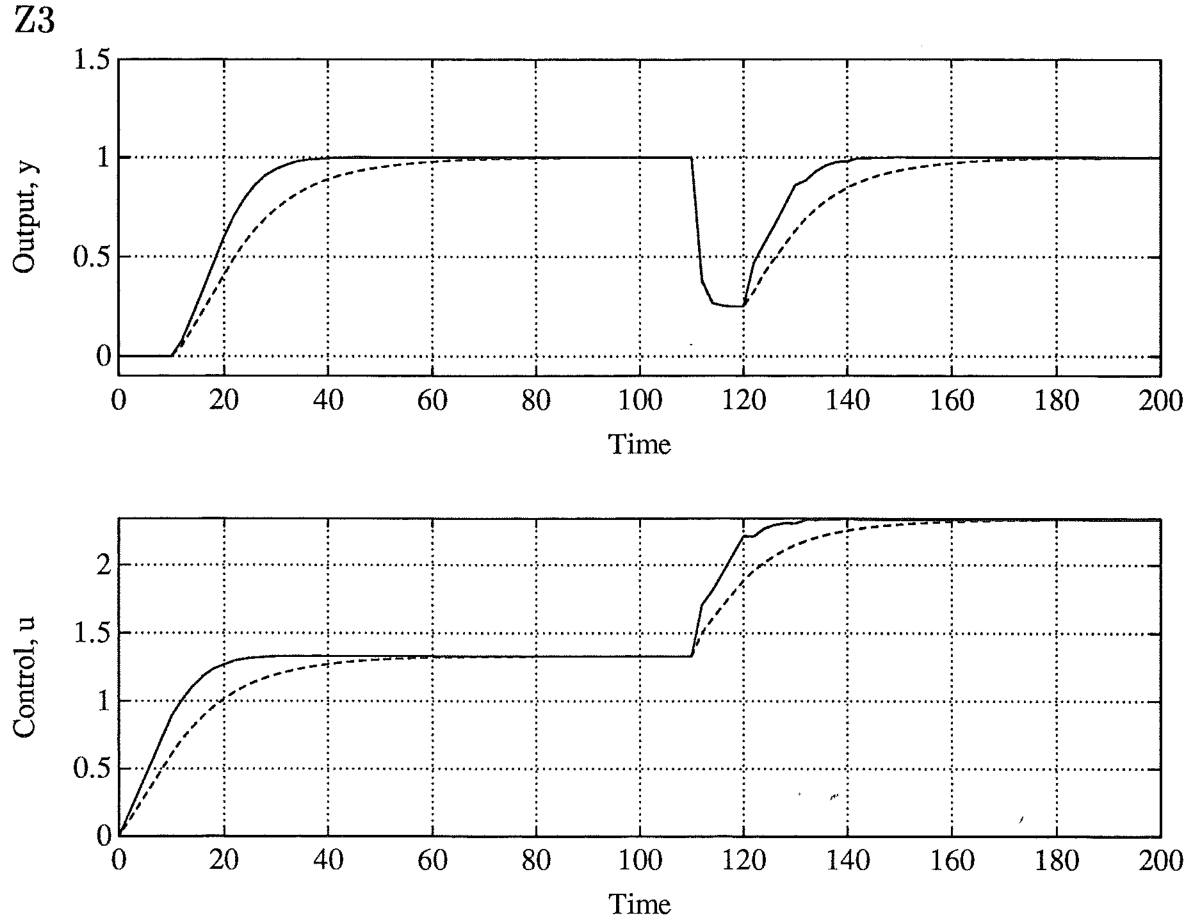



**Figure 7.25** Time response of 3PM design (solid line) and PIDWIZ design (dashed line) for Z3.

## 7.5 Interpretation of other methods as DPD

Design of PID controllers with integral criteria has been in use for a long time. This design method usually give quite good time responses, but has its drawbacks. In this section we will compare design with integral criteria with dominant pole design, as presented in the previous chapters. Since the system S1 is the one most frequently studied, this will be our main example.

### PI control

We have shown how the stability region for PI controllers can be computed from the transfer function. All controller settings corresponding to a stable closed loop system must lie inside this stability region. In Figures 7.26, 7.27, 7.28, and 7.29 a number of controller settings will be shown in the $(k, k_i)$ plane of PI controllers. Curves for constant $M_s$ and constant $\zeta_0$ will also be shown in the figures. The controller settings have been computed from the formulas in [Miller *et al.*, 1967].

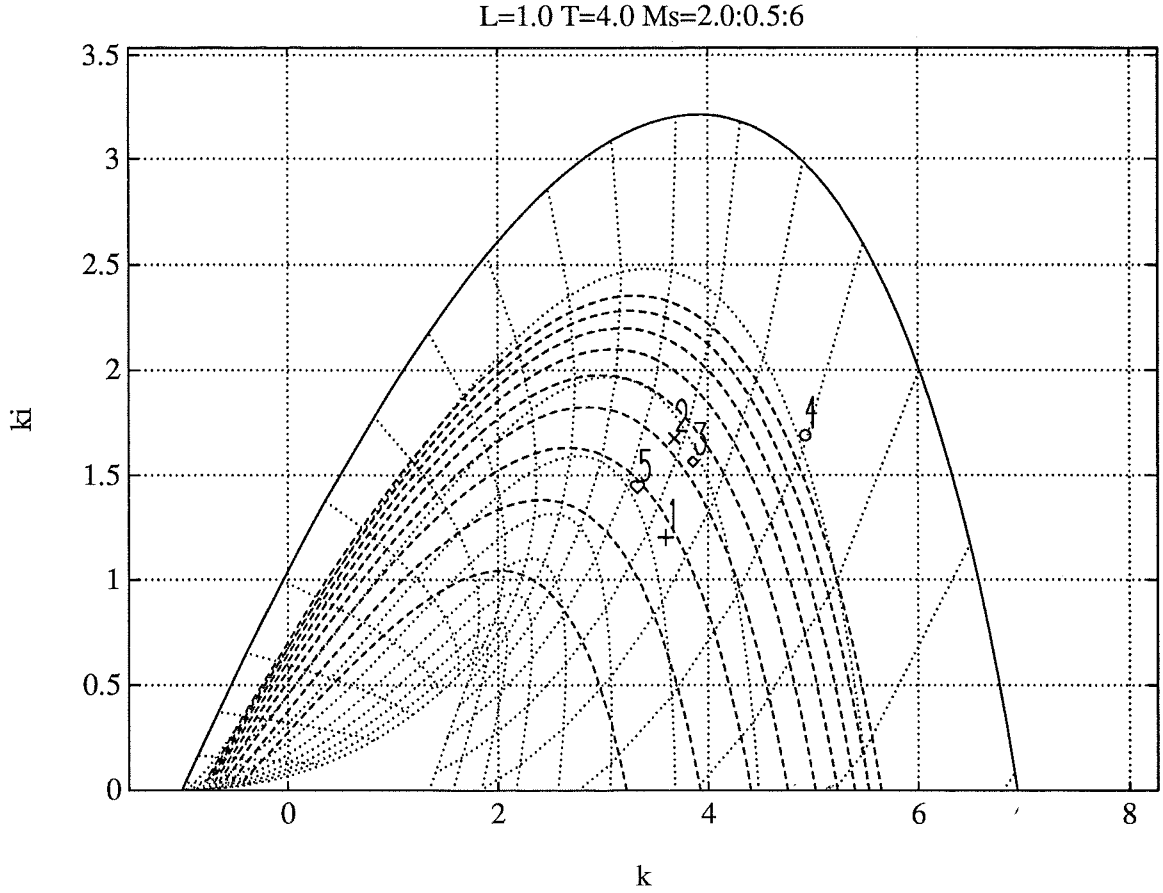



**Figure 7.26** Controller settings for S1. $T = 4$. Settings from different tuning rules are marked in the figure Ziegler-Nichols: 1, Cohen-Coon: 2, IAE: 3, ISE: 4, and ITAE: 5.

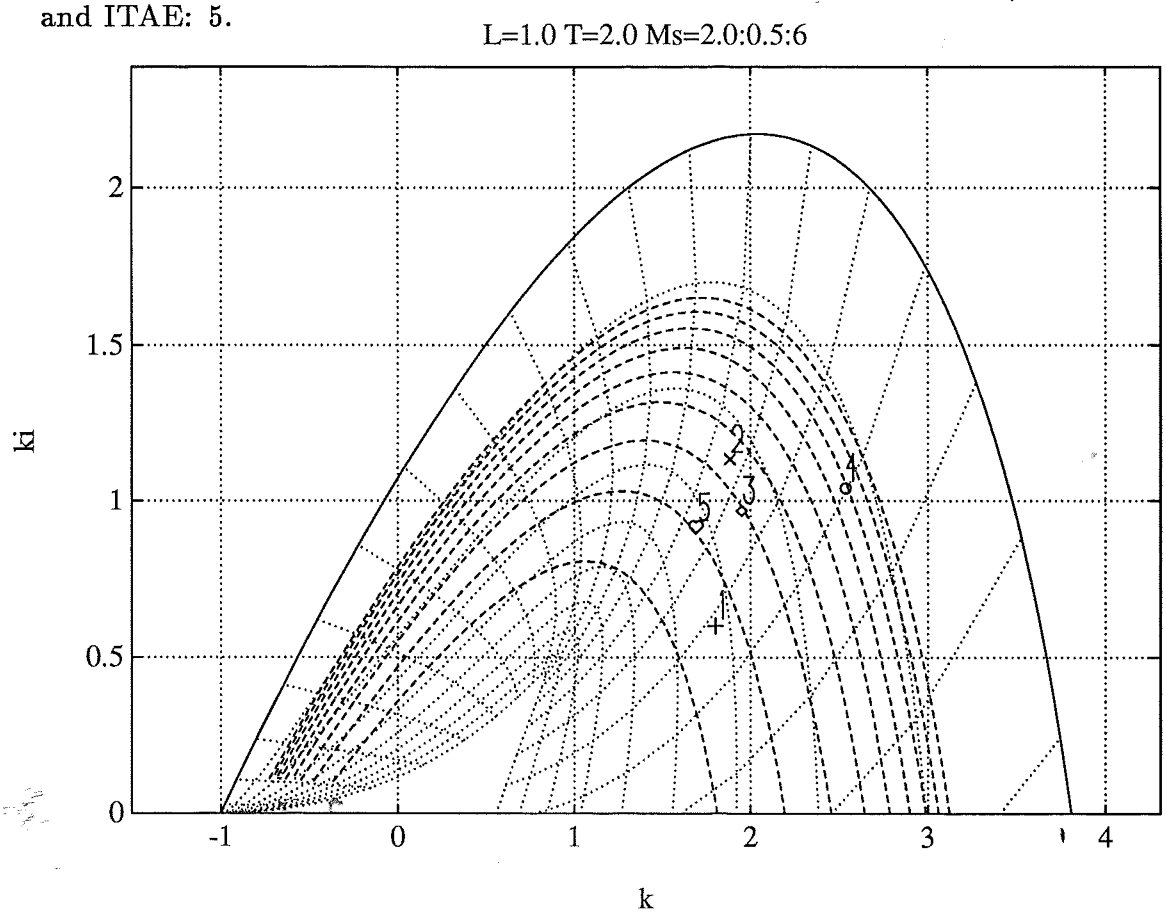



**Figure 7.27** Controller settings for S1. $T = 2$. Settings from different tuning rules are marked in the figure Ziegler-Nichols: 1, Cohen-Coon: 2, IAE: 3, ISE: 4, and ITAE: 5.

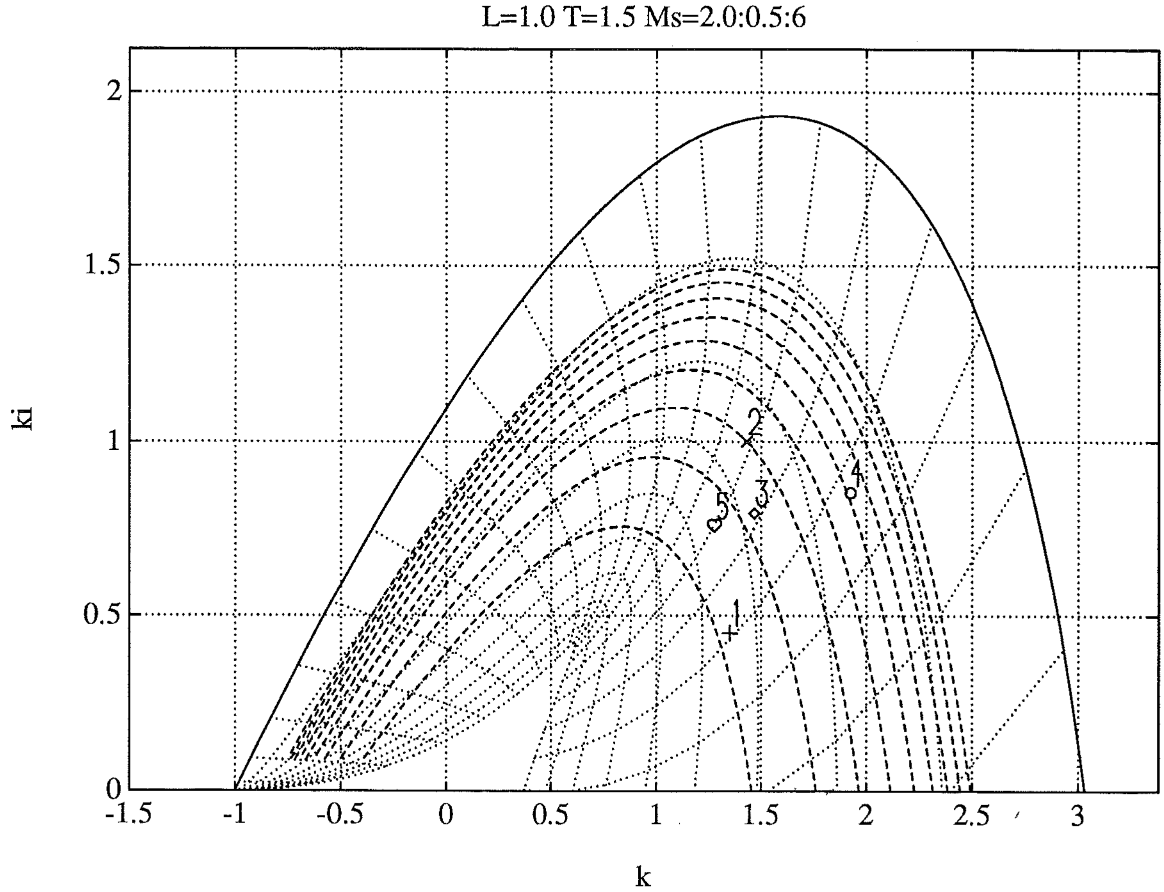



**Figure 7.28** Controller settings for S1. $T = 1.5$. Settings from different tuning rules are marked in the figure Ziegler-Nichols: 1, Cohen-Coon: 2, IAE: 3, ISE: 4, and ITAE: 5.

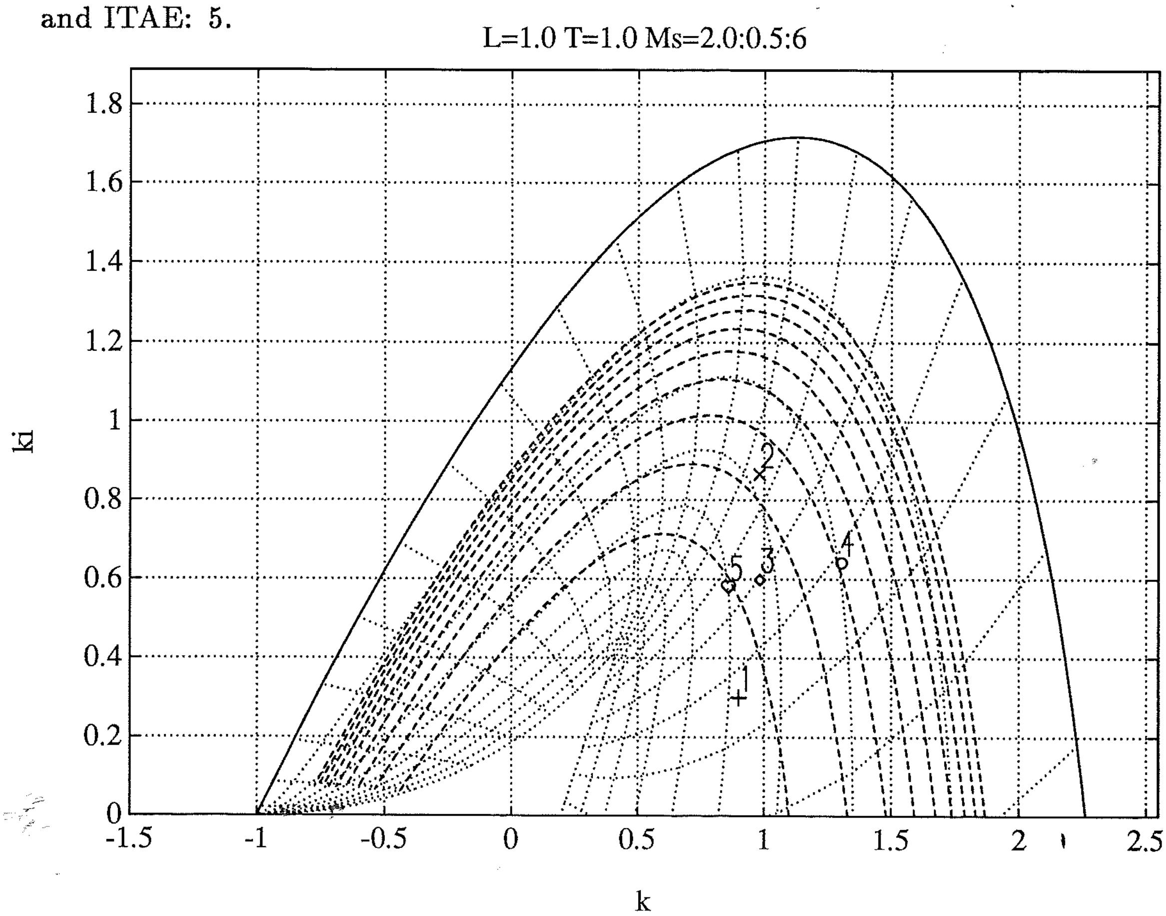



**Figure 7.29** Controller settings for S1. $T = 1$. Settings from different tuning rules are marked in the figure Ziegler-Nichols: 1, Cohen-Coon: 2, IAE: 3, ISE: 4, and ITAE: 5.

In Figures 7.26, 7.27, 7.27, and 7.27 the stability limit, $\zeta_0 = 0$, is marked by a solid line, the dotted lines parallel to the stability limit are curves for constant $\zeta_0$. The curves are for $\zeta_0 = 0.1$ to 0.9 in steps of 0.1. The other dotted curves are loci for constant $\omega_0$. The dashed curves are curves for constant $M_s$. The curves are for $M_s = 2$ to 6 in steps of 0.5.

Statements have been made about of the behaviour of controllers tuned with IE, IAE, ISE, and ITAE, see [Seborg *et al.*, 1989]. When interpreted in the dominant pole framework it is clear what happens.

***Ziegler-Nichols*** Systems designed with the Ziegler-Nichols rule have approximately the same $M_s$ value as systems designed with the ITAE rule. However the $\omega_0$ parameter is larger for ZN. This corresponds to an pole on the negative real axis closer to the origin than for ITAE. This can make ZN responses rather sluggish. Especially for long time delays ZN does not do very well. In these cases the $\omega_0$ is larger than for the other rules.

***Cohen-Coon*** The Cohen-Coon rules are based on systems described by a first order lag and a time delay. The dominant poles are placed with a fixed relative damping, $\zeta_0 = 0.2155$, and maximal $k_i$. The Cohen-Coon rule thus corresponds to a high $M_s$. The fact that the settings in the figures does not correspond to a maximal $k_i$, is due to the approximations made to get the simple CC formulas.

***IAE*** The IAE criterion produces time responses relative similar to those of CC or ITAE. Because of its physical significance the IAE criterion is recommended in process control literature, see [Shinskey, 1988] and [Shinskey, 1990]. However, the robustness of IAE controllers can be poor.

***ISE*** The ISE criterion is known to produce rather oscillatory systems. The figures show that the ISE criterion give the largest $M_s$ value for all investigated systems. The ISE criterion is unrealistic for practical use. This means poor robustness.

***ITAE*** It is generally agreed on that of all these criteria the ITAE criterion produces the most robust and well damped systems. The $M_s$ values are between 2 and 3. In [Miller *et al.*, 1967] ITAE is judged to be the best criterion. We see that ITAE correspond to a moderate $M_s$ value, and does not push $\omega_0$ too high.

Notice that all these design methods lead to high $M_s$ values and poor robustness of the closed loop system.

## PID control

It is more difficult to visualize the controller settings for a PID controller, since this requires three dimensions. Stability regions for System S1 as functions of PID controller parameters can be found in [Oppelt, 1964]. Because of the difficulties of showing the PID settings graphically the results of the investigations of S1 are displayed in Table 7.13.

**Table 7.13**

| | ZN | | | CC | | | IAE | | |
|---|---|---|---|---|---|---|---|---|---|
| $L/T$ | $\omega_0$ | $\zeta_0$ | $M_s$ | $\omega_0$ | $\zeta_0$ | $M_s$ | $\omega_0$ | $\zeta_0$ | $M_s$ |
| 0.10 | 23.95 | 0.21 | 3.42 | 20.27 | 0.21 | 4.12 | 19.08 | 0.39 | 3.05 |
| 0.20 | 12.26 | 0.20 | 3.34 | 10.40 | 0.20 | 4.02 | 10.59 | 0.32 | 3.15 |
| 0.30 | 8.36 | 0.20 | 3.26 | 7.10 | 0.20 | 3.91 | 7.49 | 0.28 | 3.19 |
| 0.40 | 6.40 | 0.19 | 3.18 | 5.44 | 0.19 | 3.81 | 5.87 | 0.24 | 3.23 |
| 0.50 | 5.22 | 0.19 | 3.10 | 4.44 | 0.19 | 3.71 | 4.87 | 0.22 | 3.25 |
| 0.60 | 4.43 | 0.19 | 3.03 | 3.76 | 0.19 | 3.61 | 4.18 | 0.20 | 3.27 |
| 0.70 | 3.86 | 0.18 | 2.96 | 3.28 | 0.19 | 3.51 | 3.68 | 0.19 | 3.29 |
| 0.80 | 3.44 | 0.18 | 2.89 | 2.92 | 0.19 | 3.42 | 3.29 | 0.18 | 3.30 |
| 0.90 | 3.10 | 0.18 | 2.82 | 2.63 | 0.19 | 3.33 | 2.99 | 0.17 | 3.30 |
| 1.00 | 2.83 | 0.18 | 2.76 | 2.40 | 0.19 | 3.25 | 2.74 | 0.16 | 3.30 |
| | ISE | | | ITAE | | | | | |
| $L/T$ | $\omega_0$ | $\zeta_0$ | $M_s$ | $\omega_0$ | $\zeta_0$ | $M_s$ | | | |
| 0.10 | 24.39 | 0.13 | 4.92 | 20.94 | 0.33 | 3.02 | | | |
| 0.20 | 12.38 | 0.12 | 5.19 | 10.62 | 0.32 | 3.06 | | | |
| 0.30 | 8.39 | 0.12 | 5.25 | 7.25 | 0.32 | 3.02 | | | |
| 0.40 | 6.40 | 0.11 | 5.22 | 5.58 | 0.31 | 2.94 | | | |
| 0.50 | 5.21 | 0.11 | 5.13 | 4.58 | 0.30 | 2.85 | | | |
| 0.60 | 4.42 | 0.11 | 5.02 | 3.91 | 0.30 | 2.77 | | | |
| 0.70 | 3.85 | 0.11 | 4.89 | 3.43 | 0.29 | 2.68 | | | |
| 0.80 | 3.42 | 0.11 | 4.76 | 3.07 | 0.29 | 2.60 | | | |
| 0.90 | 3.08 | 0.11 | 4.62 | 2.79 | 0.29 | 2.53 | | | |
| 1.00 | 2.81 | 0.11 | 4.48 | 2.56 | 0.28 | 2.46 | | | |

We see the same patterns for PID controllers as for PI controllers. The ITAE controllers shows the least $M_s$ values and ISE controllers show the highest values. In all cases shown in Table 7.13 the $M_s$ values are too high to fulfill requirements on robustness.

## 7.6 Summary

PI and PID controllers have been designed for a number of systems. The design methods were the methods developed in Chapter 4 and Chapter 5.

The proposed method offers control of the robustness of the controlled system by the choice of the design parameter $M_s$. Furthermore, the Nyquist plots of the loop transfer functions, and time responses of the closed loop systems look surprisingly similar for different processes. This is not the case for the standard tuning rules. As a consequence of this, the shape of the time responses for different systems is also very similar for different systems designed with the same $M_s$. The great advantage with the proposed methods is the predictability of the results. For all systems examined in this chapter it was possible to compute a reasonable PI or PID controller.

If we want optimality in the sense of IAE, ITAE, etc., the only way to do the design is to determine the controller parameters by numerical optimization. The controllers obtained in this manner tend to have rather high $M_s$ values and thus to be less robust, without giving significantly better time responses.

# 8

# Software Tools

This chapter describes some of the software tools used in the work with this thesis. The implementation of the design routines will also be discussed.

Computer assistance is absolutely necessary in the kind of research that is presented in this thesis. Since it is possible to obtain analytical results only in the very simplest cases most of the investigations have been done numerically. All design algorithms have been implemented in Matlab, see [MathWorks, 1990b]. All simulations have been done with Simnon, see [SSPA, 1990]. The computer algebra program Maple, see [Char *et al.*, 1988], has been used to check formulas and to explore what can be done analytically.

## 8.1 Matlab

Matlab has been the main computer tool in this thesis. The matrix is the only data structure Matlab can handle. With such a simple structure it can be difficult to handle general transfer functions. Transfer functions have therefore been represented as text strings.

EXAMPLE 8.1—Transfer function definition in Matlab
The transfer function

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

could be defined in Matlab as the string 'gst'

```
gst = 'exp(-sqrt(s))./s./(sT + 1)';
```

Note that the transfer function should be defined such that it can be evaluated when 's' is a vector. □

The evaluation of a transfer function represented as a string is done with the Matlab function 'evals', which is defined as

```
function r = evals(s, str)
%r = evals(s, str)
%Evaluates the function represented by the
%string str for the argument s.
r = eval(str);
```

By using this routine we have a safe way of evaluating a function represented as a string. It can also be done by the following commands

```
s = sarray;
eval(str);
```

Here 'sarray' is the array of arguments for which we want to evaluate 'str'. Here we must remember to assign the argument to 's' before the evaluation. This may destroy the value of the global variable 's', if there exists one. This problem can be handled automatically and safely by using Matlab's lexical scoping in the functions.

Other operations which are needed for the design routines are a routine for optimization of functions of one variable and a routine for the solution of nonlinear equations with respect to one unknown. These routines are called 'opt' and 'solve'.

The routine 'opt' is called as follows

```
[x, f] = opt(x0, str, tol);
```

The input parameters are: 'x0', the interval in which the maximum lies, 'str', the function to be maximized (it must be written as a function of 'x'), and 'tol' is the relative accuracy of the solution. The output parameters are the $x$ value of the maximum and the value in that point.

The routine 'solve' is called as follows

```
x = solve(y, x0, str, tol);
```

The input parameters are: 'y', the value we are solving the function for, 'x0', the interval in which the solution lies, 'str', the function to solve for (it must be written as a function of 'x'), and 'tol' is the relative accuracy of the solution. The output parameter is the $x$ value of the solution.

The routines have been implemented by the simplest possible methods. The optimizer 'opt' was implemented by a golden section algorithm and the nonlinear equation solver 'solve' was implemented with the bisection method. This was done to be able to handle as general transfer functions as possible, without having to compute any derivatives, Jacobians, etc.

### The design routines

The basic routine in the design system is 'dptable'. This routine is called with

```
res = dptable(str, regtype, warray, zeta, alpha)
```

The input parameters are: 'str', the transfer function, 'regtype', the type of controller we design (it may be 'pi', 'pd', 'pid', etc.), 'warray' a vector of $\omega_0$, 'zeta' is the parameter $\zeta_0$, and 'alpha' is $\alpha_0$. Depending on the value of 'regtype' this routine computes values of the controller parameters from the formulas (4.32), (4.33), or (4.53), (4.54), or (5.5), (5.6), (5.7). The values of $\omega_0$, $k$, $k_i$, $T_i$, $k_d$, and $T_d$ are returned in the array 'res'. Effort has been put into coding this routine efficiently.

EXAMPLE 8.2—Routine for computation of controllers from $\omega_0$ and $\zeta_0$
The PI controller parameters are computed by (4.32) and (4.33) given a process, $\omega_0$, and $\zeta_0$. The routine for doing it is implemented as follows.

```
function [k, ki] = dppi(str, w0, z)
zp = sqrt(1 - z*z);
p1 = w0*(-z + i*zp);
tmp = evals(p1, str);
a = real(tmp);
b = imag(tmp);
n = zp*(a.^2+b.^2);
k = -(b*z+a*zp)./n;
ki = -b.*w0./n;
```

These routines must accept a vector as 'w0'. □

The actual design routines now become very simple.

EXAMPLE 8.3—Routines for PI design with specified $\zeta_0$
To design a PI controller with a given $\zeta_0$, and $\omega_0$ chosen to maximize $k_i$, the following routine is needed

```
function r = pidesign(str, warray, zeta)
tmp = ['nthcol(dptable(''' str, ''', ''pi'', x, ' ..
        num2str(zeta) '), 3)'];
w = opt(warray, tmp, 1e-5);
res = dptable(str, 'pi', w, zeta);
```

In the routine 'pidesign' a string describing what we want to optimize is built, and the optimizer does the rest of the job. The routine 'nthcol' picks out the $n$th column of a matrix, in this case the third column which contains $k_i$. □

EXAMPLE 8.4—Routines for PI design with specified $M_s$
To design a controller such that the system gets a specified $M_s$ value we need a help routine

```
function ms = mshelp(str, warr, zs)
ms = [];
for ix=zs,
  r = pidesign(str, warr, ix);
  ws = linspace(0.1*r(1), 10*r(1), 50);
  [w, m] = mscl(ws, setgr(r), str);
  ms = [ms; m];
end;
```

This routine computes the $M_s$ values for controllers with maximal $k_i$ and for different $\zeta_0$.

The routine 'mscl' computes the $M_s$ value, given a frequency interval, a controller, and a process. The function 'setgr' converts the controller parameters to a transfer function string. The routine 'pidesignms' now becomes

```
function r = pidesignms(str, warray, ms)
w1 = warray(1); nw = length(warray); w2 = warray(nw);
s1 = ['linspace(' num2str(w1) ',' num2str(w2) ',' ..
       num2str(nw) ')'];
tmp = ['mshelp(''' str, ''', ' s1 ', x)'];
zguess = [0.1:0.1:0.9];
z = solve(ms, zguess, tmp, 1e-5);
r = pidesign(str, warray, z);
```

Each time the function 'mshelp' is called two optimizations are carried out for each element in the 'zs' vector, one to do the PI controller design and one to find $M_s$. Depending on 'warray' in 'pidesignms' the function 'mshhelp' may be called many times to get the solution with desired accuracy. □

The code for the other design methods looks very similar. To get readable code, the safety net and the default handling normally used has been stripped off the functions presented here. This way of writing code is very convenient, but can also be very inefficient numerically. The great advantage is that it is extremely simple to test new ideas.

## Computing statistics

To indicate how computationally demanding the design routines are, some computing statistics were made. All programs were run on a SPARCstation ELC. Four design algorithms were compared.

1. PI design with a fixed $\zeta_0$
2. PI design with 2PM

3. PID design with 3PM
4. PID design with MCM

The plant $G(s) = e^{-s}/(s+1)$ was used for the design computations. The results of the computations are shown in Table 8.1.

**Table 8.1** Statistics for a different design methods

| | PI | | 2PM | | 3PM | | MCM | |
|---|---|---|---|---|---|---|---|---|
| Tolerance | # of Mflops | Time (s) | # of Mflops | Time (s) | # of Mflops | Time (s) | # of Mflops | Time (s) |
| $10^{-2}$ | 0.0016 | 1.7 | 0.31 | 52 | 0.37 | 74 | 0.52 | 92 |
| $10^{-3}$ | 0.0024 | 2.3 | 0.39 | 82 | 0.48 | 120 | 0.69 | 150 |
| $10^{-4}$ | 0.0032 | 3.2 | 0.51 | 120 | 0.63 | 180 | 0.90 | 230 |
| $10^{-5}$ | 0.0040 | 4.1 | 0.63 | 170 | 0.77 | 230 | 1.10 | 330 |
| $10^{-6}$ | 0.0047 | 4.5 | 0.76 | 230 | 0.92 | 300 | 1.40 | 440 |
| $10^{-7}$ | 0.0055 | 5.3 | 0.92 | 290 | 1.10 | 380 | 1.70 | 560 |
| $10^{-8}$ | 0.0063 | 6.2 | 1.10 | 350 | 1.30 | 480 | 2.00 | 760 |

The MCM is the most demanding. The computation times and number of flops are roughly proportional to the logarithm of the tolerance.

Much of these investigations could not have been done without the computing power of the latest generation of workstations. The speed of a standard VAX 11/780 is about 1/20 of the speed of an SPARCstation ELC. A tolerance of $10^{-3}$ is sufficient for the design computations. Smaller tolerances were needed when computed parameters were plotted against some parameter in the design algorithms or in the plant. Too high a tolerance makes the plots irregular. To wait a couple of minutes to compute the controller parameters is tolerable. On a VAX 11/780 this would mean waiting for about half an hour, which is *not* tolerable. If the design methods were implemented with concern for time optimality, and were written in a compilable language the execution times would be reduced.

## 8.2 Simnon

Matlab is an excellent tool for doing matrix calculations of all kinds, computing frequency responses, plotting etc., but it is limited when it comes to simulation of dynamical systems. For example, Matlab cannot handle time delays in simulations. For these reasons the simulation package Simnon, see [SSPA, 1990], has been used. Simnon is a general simulation package, but in this work its capability of handling nonlinear equations have not been used.

One major disadvantage with Simnon is its low accuracy in the simulations. At best it is possible to get results with six digits accuracy from Simnon. Some attempts were made to use Simnon as a computing engine for optimization algorithms implemented in Matlab. It may work, but for systems with flat minima, as those of the integral criteria, there are convergence problems since the gradient computations may give irrelevant results. Simnon has also been used for optimization of dynamic system, see [Glad, 1974]. This may work for some systems, but the accuracy of the results is usually low.

### Matlab–Simnon connection

To be able to use the controller parameters computed in Matlab conveniently in Simnon a routine for sending commands from Matlab to Simnon was written. Simnon is started with the following Perl script.

```
#!/usr/local/bin/perl
$pipename = shift;
open(SIMNON, "|simnon") || die "cannot pipe\n";
select(SIMNON);
$|=1;
print "x\\n";
while(open(PIPE, $pipename)){
  while(<PIPE>){
    if (eof(PIPE)) {close(PIPE);}
    print;
    if ($_ eq "stop\n") {exit;};
  }
}
```

The programming language Perl is described in [Wall and Schwartz, 1991]. The programs run under a UNIX window system. Simnon is started in one window with the Perl script, and Matlab communicates with Simnon via two named pipes. For some reason Simnon refuses to accept commands directly from a named pipe. This is the reason for communicating via the Perl script.

The basic Matlab function for interacting with Simnon is 'simcommand', as defined by

```
function simcommand(comstr, pipe)
eval(['!echo ' '''' comstr '''' ' >> ' pipe]);
```

This routine writes the string 'comstr' into a named pipe, which is then read by the Perl script and sent to Simnon

There are also routines for getting parameter values from Simnon and assigning them to Matlab variables, 'getsimval'. This is done by forcing Simnon to write the values into a file and then reading the file in Matlab. The routine for getting parameter values from Simnon also synchronizes the two processes. If Matlab has issued a number of commands to Simnon which do not give any values in return (e.g., plotting commands), and then a call to 'getsimval', Matlab must wait until Simnon has processed the command and has sent the return value back to Matlab.

The Matlab-Simnon communication is handled by ten Matlab functions. Although these routines consist of only a few lines of code they extend the functionality of Matlab considerably. Simnon can in this way be used transparently from Matlab.

This way of using Simnon is good for 'production' simulation. It requires well tested models. Running Simnon from Matlab while developing models is pointless, in that case the interactive capabilities of Simnon are needed. The main drawback with this way of using Simnon and Matlab together is that there is no way of interrupting Simnon from Matlab without terminating Simnon.

## 8.3 Maple

Although numerical calculations are useful and illuminating we can get even more insight from analytical solutions that show parameter dependencies explicitly. This has been possible in a few cases. The computer algebra package Maple, see [Char *et al.*, 1988], has been used to compute some of the analytical expressions and check the analytical examples.

Maple is very simple to use, but sometimes it may be difficult to get simple and esthetically appealing results. One easily gets many pages of meaningless formulas without any structure.

EXAMPLE 8.5—Use of Maple
As an example the Maple code for computing (4.32) and (4.33) is given.

```
Gr := proc(s) k + ki/s; end;      # Define the controller.
p1 := w0*(-z0 + I*sqrt(1-z0^2)); # Define the closed
p2 := w0*(-z0 - I*sqrt(1-z0^2)); # loop poles.
G1 := A + I*B;                   # Define G(p1) and G(p2).
G2 := A - I*B;
tmp1 := evalc(1 + G1*Gr(p1)); # Compute the characteristic
tmp2 := evalc(1 + G2*Gr(p2)); # equation.
solve({tmp1, tmp2}, {k, ki}); # Solve the equations for k
                              # and ki.
```

As results of these commands we get for $k$

```
                2 1/2
      A (1 - z0 )      + B z0
    - ----------------------

               2 1/2    2     2
       (1 - z0 )     (A  + B )
```

and for $k_i$

```
                 w0 B
 - ---------------------
           2 1/2    2    2
   (1 - z0 )     (A   + B )
```

□

Maple have been used to compute and check most of the formulas in the text of Chapter 4 and Chapter 5. The analytical examples at the end of Chapter 4 and Chapter 5 have also been computed with Maple, and so has the examples of SO for PI control in Chapter 3.

## 8.4 Cooperating programs

Using many programs together requires that the output from one program should be usable as input to the next program without any manual editing. It is also desirable to be able to write variables in a program to a file without any loss of accuracy. These problems have been encountered in using Matlab and Simnon together. Simnon cannot write its variables on files with full precision. At least one decimal is lost when a variable is written on the screen or to a file. This can cause problems if Simnon is used for computings in an optimization. It should also be possible to signal errors from one program to another.

# 9

# Conclusions

The problem of synthesizing PID controllers has been considered. A number of different approaches to PID controller design have been reviewed. None of them have been entirely satisfactory. A new method for tuning simple controllers, such as PI, PD, and PID controllers has been presented. The method uses a frequency domain model of the plant. It is based on the assumption that the behaviour of the system can be characterized by the closed loop poles closest to the origin. We call this the Dominant Pole principle. The dominant poles are denoted $p_{1,2} = \omega_0(-\zeta_0 \pm i\sqrt{1-\zeta_0^2})$ and $p_3 = -\alpha_0\omega_0$. Different methods for assigning two or three poles of the closed system have been investigated. The criteria have been to fulfill requirements on performance and robustness of the closed system. The design methods are frequency domain methods. They do not assume anything about the structure of the model of the plant. This means that we are not restricted to rational polynomial transfer functions, e.g., time delays can easily be handled. Few simple methods for designing PID controllers with no specified structure on the transfer function has been published. An example is the Ziegler-Nichols self oscillation method, see [Ziegler and Nichols, 1942].

Various performance criteria have been discussed. One that is computationally easy to handle and has a sound physical interpretation is IE, the integrated error of the process output after a load disturbance at the process input. The minimization of IE, under certain restrictions, has been chosen as the performance criterion. IE is minimized with respect to $\omega_0$.

To have control over the robustness of the system, a value of the maximum of the sensitivity function is specified in the design. This is the $M_s$ value. The number $1/M_s$ can be interpreted as the smallest distance from the Nyquist curve of the loop transfer function to $-1$. To obtain this, $\zeta_0$ is chosen to give the specified $M_s$ of the controlled system. This way, the parameter $M_s$ becomes the only design parameter of the method.

Several variations of the design method have been tested. As conclusion two methods are recommended. For PI controller design, $\omega_0$ is chosen to maximize $k_i$. The parameter $\zeta_0$ is chosen to get a specified $M_s$ value of the loop transfer function. For PID controller design the same choices of $\omega_0$ and $\zeta_0$ can be made. The third parameter, $\alpha_0$, necessary to specify PID controllers can be set to 1, except for resonant plants where $\alpha_0$ should be set to a lower value. Normally $\alpha_0 = 0.5$ works well. Setting $M_s = 1.6$ gives in most cases a non-oscillating system, with set point and load responses without overshoots. If $M_s = 2.0$ we get a somewhat faster systems, which may have overshoots.

Given the frequency response of a plant, it is natural to ask which parts of the frequency response are used in a certain design method. Some numerical investigations on this problem have been made. The results are that for PI and PID control nothing is gained by using information of the model in a frequency range where the phase lag is more than 180°. Normally four points on the Nyquist curve, corresponding to the phase lags 0, 45°, 135°, and 180° are enough information to design a good PID controller. To design controllers for integrating processes information is needed up to the frequency corresponding to the phase lag $\approx 225°$.

The methods can handle a broad spectrum of dynamics uniformly, in the sense that the time and frequency responses of the controlled systems will look very similar.

The parametrizations studied are very flexible. It is easy to implement and test various design criteria. The methods do not require any sophisticated numerical software, such as differential equation solvers. Of course, simulation of the closed systems is advised, but it is not part of the design method. The methods are moderately computationally demanding, e.g., we do not have to solve any differential equations numerically. The design routines have been implemented in Matlab. The simplest possible numerical methods for optimization and equation solving have been used to be able to handle as wide a variety of plants as possible.

It has been shown that in the case of PI control all other design rules can be interpreted as DPD. Study of the standard design rules, ZN, IAE, ISE, and ITAE, for standard systems has given some guidance for design in the general case. If we want optimal controllers in the sense of IAE, ISE, etc, the optimization must be done numerically by solving differential equations. These criteria may give closed systems with very poor robustness. The criteria are not always physically relevant.

There are a few problems with the methods. Designing by maximizing $k_i$ can be dangerous, since a maximal $k_i$ does not necessarily mean that the system will be stable. In particular, design of systems with resonances is problematic. This is, however, not the prime application area of PID controllers. Moreover, the parameter function $k_i(\omega_0)$ may have several maxima, to get a stable closed loop system the numerical routines must pick the first maximum.

It is not possible to solve anything but the simplest cases analytically. Therefore it is difficult to give any proofs of the properties of the design methods for more realistic plants.

**Further research**

It has been suggested to use on-line expert systems to tune and supervise large plants with hundreds of control loops. The presented controller tuning methods are suitable for such a system. The supervisor system may come up with several different kinds of process models, all which can be handled by the proposed methods. We also have a good design parameter, $M_s$, by which we can control the behaviour of the closed loop system.

This design method could also be used in an adaptive PID controller. The models estimated by a recursive estimator could very well be used as models for PID controller design. With the software system developed for this thesis it would be easy to make a simulation study. The simulation and parameter estimator in Simnon could be connected with the design procedures in Matlab. In such a case it would be interesting to study a larger process with several interconnected control loops.

If the design methods are to be used on a routine basis, a better and more efficient numerical implementation will probably be needed. It would be interesting to see how much could be gained by coding the design routines in a compiled language.

# 10

# References


ATHANS, M. (1971): "On the design of P-I-D controllers using optimal linear regulator theory." *Automatica*, **7**, pp. 643–647.

BLICKLEY, G. J. (1988): "PID tuning made easy with hand-held computer." *Control Engineering*, November.

BLICKLEY, G. J. (1990): "Modern control started with Ziegler-Nichols tuning." *Control Engineering*, October, pp. 11–17.

BUCKLEY, P. S. (1964): *Techniques of Process Control.* John Wiley & Sons, New York, NY.

CALLENDER, A., D. R. HARTREE, and A. PORTER (1936): "Time lag in a control system." *Philos. Trans. A.*, **235**, pp. 415–444.

CHAR, B. W., K. O. GEDDES, G. H. GONNET, M. B. MONAGAN, and S. M. WATT (1988): *Maple – Reference Manual.* Symbolic Computation Group, Department of Computer Sciences, University of Waterloo, Waterloo, Ontario, Canada, fifth edition.

CLARK, R. N. (1988): "Another reason to eschew pole-zero cancellation." *IEEE Control Systems Magazine*, April, pp. 87–88.

COHEN, G. H. and G. A. COON (1953): "Theoretical consideration of retarded control." *Transactions of the ASME*, **75**, July, pp. 827–834.

EITELBERG, E. (1987): "A regulating and tracking PI(D) controller." *International Journal of Control*, **45:1**, pp. 91–95.

FOXBORO, INC. (1979): *Controller Tuning Guide*, PUB 342A.

FRANK, P. M. (1968): "Das $PIS_m$-Regelungssystem, eine Erweiterung des PI-Regelungssystems für Totzeitstrecken." *Regelungstechnik*, **16:7**, pp. 306–313.

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.

Glad, T. (1974): "A program for the interactive solution of parametric optimization problems in dynamic systems." Technical Report TFRT-3084, Dept. of Automatic Control, Lund Inst. of Technology, Lund, Sweden.

van der Grinten, P. M. E. M. (1963): "Finding optimum controller settings." *Control Engineering*, December, pp. 51–56.

Haalman, A. (1965): "Adjusting controllers for a deadtime process." *Control Engineering*, July, pp. 71–73.

Hägglund, T. (1991): "A dead-time compensating three-term controller." In *9th IFAC/IFORS Symposium on Identification and System Parameter Estimation*, Budapest, Hungary.

Hang, C. C. (1989): "The choice of controller zeros." *IEEE Control Systems Magazine*, January, pp. 72–75.

Hang, C. C., K. J. Åström, and W. K. Ho (1991): "Refinements of the Ziegler-Nichols tuning formula." *IEE PROCEEDINGS-D*, **138:2**, pp. 111–118.

Harris, S. L. and D. A. Mellichamp (1985): "Controller tuning using optimization to meet multiple closed-loop criteria." *AIChE Journal*, **31:3**, pp. 484–487.

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.

Hazebroek, P. and B. L. van der Waerden (1950a): "The optimal adjustment of regulators." *Transactions of the ASME*, **72**, April, pp. 317–322.

Hazebroek, P. and B. L. van der Waerden (1950b): "Theoretical considerations on the optimum adjustment of regulators." *Transactions of the ASME*, **72**, April, pp. 309–315.

Hippe, P., C. Wurmthaler, and F. Dittrich (1987): "Comments on 'A regulating and tracking PI(D) controller'." *International Journal of Control*, **46:5**, pp. 1851–1856.

HOLMBERG, U. (1991): *Relay Feedback of Simple Systems*, Ph. D. Thesis. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

HOROWITZ, I. (1963): *Synthesis of Feedback Systems*. Academic Press, New York, NY.

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.

KALMAN, R. E. (1960): "Contributions to the theory of optimal control." *Boletin de la Sociedad Mathematica Mexicana*, **5**, pp. 102–119.

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.

LARSSON, J. E. and P. PERSSON (1986a): "Ett expertsystemsnitt för Idpac." In *SAIS '86 Workshop*, Linköping, Sweden.

LARSSON, J. E. and P. PERSSON (1986b): "Knowledge representation by scripts in an expert interface." In *American Control Conference*, volume 2, pp. 1159–1162, Seattle, Washington.

LARSSON, J. E. and P. PERSSON (1987a): "Ett intelligent hjälpsystem för systemidentifiering." In *SAIS '87 Workshop*, Uppsala, Sweden.

LARSSON, J. E. and P. PERSSON (1987b): *An Expert Interface for Idpac*, Licentiate Thesis. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

LARSSON, J. E. and P. PERSSON (1987c): "The expert interface strikes back." *ORDO*, **21:1**, pp. 26–31. In Swedish.

LARSSON, J. E. and P. PERSSON (1987d): "An expert system interface for Idpac." In *Joint SERC–STU Workshop on Expert Systems and Data Bases for Control System Design and Application*, Cambridge, England.

LARSSON, J. E. and P. PERSSON (1988a): "An intelligent help system for Idpac." In *8th European Conference on Artificial Intelligence*, pp. 119–123, München, Germany.

LARSSON, J. E. and P. PERSSON (1988b): "The knowledge database used in an expert system interface for Idpac." In *IFAC Workshop on Artificial Intelligence in Real-Time Control*, pp. 107–112, Swansea, Wales.

LARSSON, J. E. and P. PERSSON (1991): "An expert system interface for an identification program." *Automatica*, **27:6**, pp. 919–930.

LEE, J. (1989): "On-line PID controller tuning from a single, closed-loop test." *AIChE Journal*, **35:2**, pp. 329–331.

LEE, J., W. CHO, and T. F. EDGAR (1990): "An improved technique for PID controller tuning from closed-loop tests." To appear.

LILJA, M. (1989): *Controller Design by Frequency Domain Approximation*, Ph. D. Thesis. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

LJUNG, L. and T. SÖDERSTRÖM (1983): *Theory and Practice of Recursive Identification*. The MIT Press, Cambridge, MA.

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.

LUENBERGER, D. G. (1984): *Linear and nonlinear programming*. Addison-Wesley, Reading, MA.

LUNDH, M. (1991): *Robust Adaptive Control*, Ph. D. Thesis. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

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.

MARSILI-LIBELLI, S. (1981): "Optimal design of PID regulators." *International Journal of Control*, **33:4**, pp. 601–616.

MATHWORKS (1990a): *Optimization Toolbox User's Guide*. MathWorks, Cochituate Place, 24 Prime Parkway, Natick, MA 01760, USA.

MATHWORKS (1990b): *PRO-MATLAB – User's Guide*. MathWorks, Cochituate Place, 24 Prime Parkway, Natick, MA 01760, USA.

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.

MOORE, R. L., H. ROSENOF, and G. STANLEY (1990): "Process control using a real time expert system." In *11th IFAC World Congress*, volume 7, pp. 234–239, Tallinn, Estonia.

M'SAAD, M. (1991): "A showcase of adaptive control designs." In COMMAULT *et al.*, Ed., *Proceedings of the First European Control Conference, ECC 91,* Grenoble, France, volume 3, pp. 2374–2375, Paris. Hermes.

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.

PEMBERTON, T. J. (1972): "PID: The logical control algorithm." *Control*

*Engineering*, May, pp. 66–67.

PERSSON, P. and K. J. ÅSTRÖM (1992): "Dominant pole design – a unified view of PID controller tuning." In *IFAC Symposium on Adaptive Systems in Control and Signal Processing*, Grenoble, France.

RAD, A. B. and P. J. GAWTHROP (1991): "Explicit PID self-tuning control for systems with unknown time delay." In *IFAC International Symposium ITAC 91 Preprint*, volume 5, Singapore.

RAKE, H. (1980): "Step response and frequency response methods." *Automatica*, **16**, pp. 519–526.

RIVERA, D. E., M. MORARI, and S. SKOGESTAD (1986): "Internal model control. 4. PID controller design." *Industrial & Engineering Chemistry Process Design and Development*, **25**, pp. 252–265.

ROVIRA, A. A., P. W. MURRILL, and C. L. SMITH (1969): "Tuning controllers for setpoint changes." *Instruments and Control Systems*, December, pp. 67–69.

SANATHAN, C. K. (1989): Letter to K. J. Åström.

SEBORG, D. E., T. F. EDGAR, and D. A. MELLICHAMP (1989): *Process Dynamics and Control*. John Wiley & Sons, New York, NY.

SHIGEMASA, T., Y. IINO, and M. KANDA (1987): "Two degrees of freedom PID auto-tuning controller." In *ISA Conference*.

SHINSKEY, F. G. (1988): *Process Control Systems*. McGraw-Hill Book Company, New York, NY, 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. (1991): "Evaluating feedback controllers challenges users and vendors." *Control Engineering*, September, pp. 75–78.

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.

SOOD, M. and H. T. HUDDLESTON (1973): "Optimal control settings for random disturbances." *Instrumentation Technology*, March, pp. 47–49.

SSPA (1990): *SIMNON User's Guide for MS-DOS Computers*. SSPA Systems, Box 24001, S-400 22 Göteborg, Sweden. Version 3.0.

SUNDARESAN, K. R. and P. R. KRISHNASWAMY (1978): "Estimation of time delay time constant parameters in time, frequency, and Laplace domains." *The Canadian Journal of Chemical Engineering*, **56**, pp. 257–262.

TAKAHASHI, Y., M. J. RABINS, and D. M. AUSLANDER (1972): *Control and Dynamic Systems.* Addison-Wesley, Reading, MA.

THOMAS, B. (1990): *New PID Parameter Tuning Methods for Industrial Applications*, Ph. D. Thesis. School of Electrical and Computer Engineering, Control Engineering Laboratory, Chalmers University of Technology, Göteborg, Sweden.

TRUXAL, J. G. (1955): *Automatic Feedback Control System Synthesis.* McGraw-Hill Book Company, New York, NY.

UNBEHAUEN, H. and G. P. RAO (1987): *Identification of Continuous Systems.* North-Holland, Amsterdam.

WALL, L. and R. SCHWARTZ (1991): *Programming Perl.* O'Reilly & Associates, Inc., Sebastopol, CA.

WALTON, K. and J. E. MARSHALL (1984): "Closed form solution for time delay systems' cost functionals." *International Journal of Control*, **39:5**, pp. 1063–1071.

WEBER, T. W. and M. BHALODIA (1979): "Optimal behaviour of a third-order process under feedback control." *Industrial & Engineering Chemistry Process Design and Development*, **18**, pp. 217–223.

WILLS, D. M. (1962): "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.

WURMTHALER, C. and P. HIPPE (1974): "PID-Regler mit integriertem Führungsgrößenformer." *Regelungstechnik*, **22:9**, pp. 284–285.

YAMAMOTO, S. and I. HASHIMOTO (1991): "Present status and future needs: The view from Japanese industry." In *Chemical Process Control CPCIV*, pp. 1–28, Padre Island, TX.

YUWANA, M. and D. E. SEBORG (1982): "A new method for on-line controller tuning." *AIChE Journal*, **28:3**, pp. 434–440.

ZHUANG, M. (1991): "On PID controller settings using simple formulae." Technical Report CE/S/60, The University of Sussex, School of Engineering and Applied Sciences.

ZHUANG, M. and D. ATHERTON (1991): "Tuning PID controllers with integral performance criteria." In *Control '91*, Heriot-Watt University, Edinburgh, U. K.

ZIEGLER, J. G. and N. B. NICHOLS (1942): "Optimum settings for automatic controllers." *Transactions of the ASME*, **64**, pp. 759–768.

ZIEGLER, J. G. and N. B. NICHOLS (1943): "Process lags in automatic-control circuits." *Transactions of the ASME*, **65:5**, pp. 433–443.

ÅRZÉN, K.-E. (1987): *Realization of Expert System Based Feedback Control*, Ph. D. Thesis. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

ÅSTRÖM, K. J. (1970): *Introduction to Stochastic Control Theory*. Academic Press, New York, NY.

ÅSTRÖM, K. J. (1976): *Reglerteori*. AWE/Gebers, Stockholm, second edition.

ÅSTRÖM, K. J. (1988): "Dominant pole placement design of PI regulators." Technical Report TFRT-7381, Dept. of Automatic Control, Lund Inst. of Technology, Lund, Sweden.

ÅSTRÖM, K. J., J. J. ANTON, and K.-E. ÅRZÉN (1986): "Expert control." *Automatica*, **22:3**, pp. 277–286.

ÅSTRÖM, K. J., B. BERNHARDSSON, and A. RINGDAHL (1991): "Solution using robust adaptive pole placement." In COMMAULT *et al.*, Ed., *Proceedings of the First European Control Conference, ECC 91*, Grenoble, France, volume 3, pp. 2340–2345, Paris. Hermes.

ÅSTRÖM, K. J. and T. HÄGGLUND (1988): *Automatic Tuning of PID Controllers*. Instrument Society of America (ISA), Research-Triangle Park, NC.

ÅSTRÖM, K. J. and T. HÄGGLUND (1993): *Automatic Tuning of PID Controllers*. Instrument Society of America (ISA), Research-Triangle Park, NC, second edition. To appear.

ÅSTRÖM, K. J. and T. HÄGGLUND (1984): "Automatic tuning of simple regulators with specifications on phase and amplitude margins." *Automatica*, **20:5**, pp. 645–651.

ÅSTRÖM, K. J. and T. HÄGGLUND (1985): "Dominant pole design." Technical Report TFRT-7282, Dept. of Automatic Control, Lund Inst. of Technology, Lund, Sweden.

ÅSTRÖM, K. J., C. C. HANG, and P. PERSSON (1989): "Towards intelligent PID control." In *IFAC Workshop on Artificial Intelligence in Real-Time Control*, Shenyang, China.

ÅSTRÖM, K. J., C. C. HANG, P. PERSSON, and W. K. HO (1992): "Towards intelligent PID control." *Automatica*, **28:1**, pp. 1–9.