Advances in Industrial Control

Ramon Vilanova
Antonio Visioli *Editors*

PID Control in the Third Millennium

Lessons Learned and New Approaches

Springer

Advances in Industrial Control

For further volumes:
www.springer.com/series/1412

Ramon Vilanova · Antonio Visioli
Editors

PID Control in the Third Millennium

Lessons Learned and New Approaches

Springer

Editors

Ramon Vilanova
Dept of Telecommunic. & Systems Engin.
School of Engineering
Universitat Autònoma de Barcelona
Barcelona
Spain

Antonio Visioli
Facoltà di Ingegneria, Dipto. Elettronica per l'Automazione
Università Brescia
Brescia
Italy

ISSN 1430-9491 e-ISSN 2193-1577
Advances in Industrial Control
ISBN 978-1-4471-2424-5 e-ISBN 978-1-4471-2425-2
DOI 10.1007/978-1-4471-2425-2
Springer London Dordrecht Heidelberg New York

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2012931321

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Series Editors' Foreword

The series *Advances in Industrial Control* aims to report and encourage technology transfer in control engineering. The rapid development of control technology has an impact on all areas of the control discipline. New theory, new controllers, actuators, sensors, new industrial processes, computer methods, new applications, new philosophies…, new challenges. Much of this development work resides in industrial reports, feasibility study papers and the reports of advanced collaborative projects. The series offers an opportunity for researchers to present an extended exposition of such new work in all aspects of industrial control for wider and rapid dissemination.

The Editors of the *Advances in Industrial Control* series have long had a policy of supporting the publication of monographs about the different aspects of PID control and its implementation. This recognises the incontrovertible fact that PID control is *the* ubiquitous controller applied in industry. At the turn of the century and the beginning of the new millennium, the Editor-in-Chief of the Instrument Society of America Transactions conducted an informal survey of around 100 industrial control experts and academics to determine "The Century's Greatest Contributions to Control Practice" [1]. The results make interesting reading, for in Category 3: Controls, decision and communications instrumentation, the PID controller took second place, beaten only by the microprocessor. Nevertheless, boosted by industrial votes, the Zeigler–Nichols tuning method came top of Category 2: Techniques. These are some indicators of just how important the PID control field is to some industrial sectors, especially the process industries.

The *Advances in Industrial Control* Series Editors' interest in supporting publications on PID control theory, implementation, and applications has produced a good library of PID-focussed monographs for the series over the years:

1999 C.C. Yu: Autotuning of PID Controllers. ISBN 978-3-540-76250-8;
1999 A. Datta, M.T. Ho and S.P. Bhattacharyya: Structure and Synthesis of PID Controllers. ISBN 978-1-85233-614-1;
1999 K.K. Tan, Q.-G. Wang and C.C. Hang with T.J. Hagglund: Advances in PID Control. ISBN 978-1-85233-138-2;
2006 A. Visioli: Practical PID Control. ISBN 978-1-84628-585-1;

2007 A.W. Ordys, D. Uduehi, M.A. Johnson (eds.): Process Control Performance Assessment. ISBN 978-1-84628-623-0;

2010 A. Visioli and Q.-C. Zhong: Control of Integral Processes with Dead Time. ISBN 978-0-85729-069-4; and

2011 T. Liu and F. Gao: Industrial Process Identification and Control Design. ISBN 978-0-85729-976-5.

In making the above list, *Advances in Industrial Control* series volumes of related interest, for example, on controller performance assessment, windup in control, fragility of digital controllers and many industrial applications, have not been listed. So, just when it might be thought that there is little more to be written on PID controller design, implementation and applications, *PID Control in the Third Millennium* collated by Editors, Ramon Vilanova and Antonio Visioli provides a timely reminder that this research field is still vibrantly active.

At eighteen chapters, the book shows how the more recent developments in control systems theory have had an impact on PID control design and highlights areas where theory and technology have the potential to force changes on the software and hardware of the PID control device. The book's editors have managed to collect, collate and even contribute to some fascinating chapters, and each reader will find aspects from the different contributions that illuminate or provide insight for their own particular interests in this very extensive control field.

Historical Background

Aidan O'Dwyer is well known for his encyclopaedic volume [2], and he contributes an opening chapter that very usefully updates the reader on recent trends found in publications in this still growing field. In this chapter, the reader will also find some nicely balanced remarks about the industrial uptake of publications of this research.

Principles and Taxonomy

Having attempted to create a classification framework for PID control methods in the past [3], it was interesting to see structural principles converging to a two-branch tree that was found at various points in the volume. However, authors applied various names to the two branches, "model-based versus model-free", "parametric versus non-parametric", and borrowing terminology from adaptive control, "indirect versus direct methods". These terms reflected the importance of whether or not an explicit model identification step was involved or whether the PID controller parameters were going to be computed directly from the process data/knowledge but, whichever route was selected, the essential practical outcome was a careful prescription for the way process data was collected (details of the experimental framework to be used) and processed.

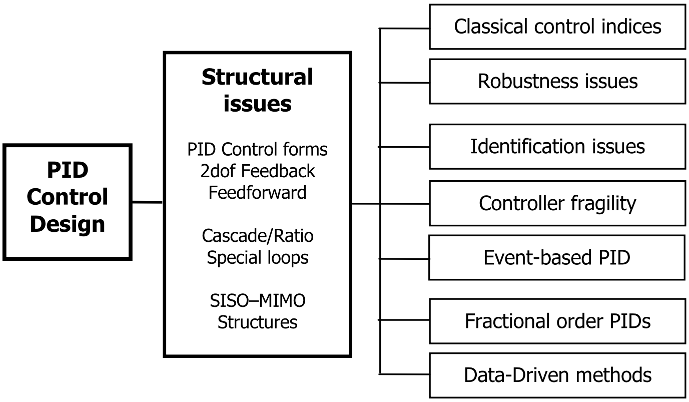


Fig. 1 Aspects of PID controller design

Design for PID Control

The influence of methods from the mainstream of new control design techniques is well captured across the eighteen chapter contributions. A summary of most of the design directions is best captured by the schematic in Fig. 1.

Reading the chapters, one interesting observation is just how thoroughly robustness issues have permeated the PID control design method. Even so, as Fig. 1 shows, there are other new design avenues opening up: PID controller fragility, fractional-order PID controllers, and data-driven PID methods, to name just three.

Technology, PID Implementation, and Applications

A recent article in the *IEEE Control Systems Magazine* [4], noted that "PID hardware is now dominated by five major vendors—ABB, Emerson, Foxboro (Invensys), Honeywell and Yokogawa", and this may affect the rapid take up of new PID control design ideas and concepts in PID hardware modules. However, the same article shows an extensive set of PID software modules and packages (over 25 products were listed), and this market is probably far more flexible. This is certainly one avenue for showcasing many of the new and fertile ideas discussed in this book. One particularly intriguing research direction is the potential of wireless technology to change the way PID controllers operate (Chap. 14 by G.K. McMillan). In a very closely related chapter, event-based PID control design is investigated using an instructive tutorial-style presentation (Chap. 16 by J. Sánchez, A. Visioli, and S. Dormido) that should interest many readers.

In a recent conversation with a former graduate student, now a professional control engineer in a large petroleum industry major, my enquiries about PID controller tuning elucidated the response that at his level PID controller tuning was a "rare bird" and that technicians used simple routines and looked after all the PID controller tuning in the company. This observation returns us to the issue that initiated the search for the ideal push-button PID tuning module; a search that was revitalised in the 1980s and saw the emergence of Åström's relay tuning concept and other devices based on the simple expert systems of the era. With all manner of new ideas posited in this volume, it will be interesting to see what impact they have on PID control technology in the years to come.

The editors, Ramon Vilanova and Antonio Visioli are to be congratulated on bringing together this invaluable collection of chapter-length contributions to provide a timely update of progress in the many aspects of PID control research, development, and industrial applications. We should also note that the editors have developed this book concept from a position of strength, for they are both chapter authors in this volume and regular contributors to this research field. It is also encouraging to see that a new generation of PID control researchers is now contributing to the field.

References

1. Rhinehart, R.R.: The Century's greatest contributions to control practice. ISA Trans. **39**, 3–13 (2000)
2. O'Dwyer, A.: Handbook of PI and PID Controller Tuning Rules, 3rd edn. Imperial College Press, London (2009). ISBN: 978-1-84816-242-6
3. Johnson, M.A., Moradi, M.H. (eds.): PID Control: New Identification and Design Methods (2005). ISBN: 978-1-85233-7092-5
4. Li, Y. Ang, K.H., Chong, G.C.Y.: Patents, software and hardware for PID control: An overview and analysis of the current art. IEEE Control Syst. **26**(1), 42–54 (2006)

Industrial Control Centre
Glasgow, Scotland, UK

M.A. Johnson

Preface

Proportional-Integral-Derivative (PID) controllers are by far the most adopted controllers in industry owing to the advantageous cost/benefit ratio that they are able to provide. In the last century, a large number of researchers have considered them, and industrial products have improved their functionality significantly.

Proposals for the design and tuning of PID based control systems have generated a really vast literature during the last decades. This sparked the special attention of the *IFAC Workshop PID'00 Present and Future of PID* control held in Terrassa, Spain, in April 2000. This event provided a state-of-the-art on many aspects of PID control, ranging from specialized theoretical research topics and tuning methods to interesting reviews of technological products, patents and software, confirming that the PID controller continues to generate deep interest and is a good augury for future research in the field of PID controllers.

Starting from this special event, in the last ten years there has been a renewed interest in the research on PID controllers, as witnessed by a large number of papers published on this subject. This book gives an overview of the advances made for PID controllers during this last decade.

The book is divided into four parts in which well-known experts and specialists address different topics (one for each chapter) in the following fields: (i) new approaches for tuning PID controllers; (ii) control structures and configurations for PID control; (iii) issues in PID control; (iv) non-standard approaches to PID control.

The first part concentrates on tuning methods. It starts with a review provided by A. O'Dwyer of the extensive literature of proposals for PI and PID controller tuning rules based on First-order plus Dead-time models. The next chapters follow by concentrating on specific topics related to PID tuning. A. Leva and M. Maggio discuss the availability of a process model and its use for assessing the behavior and characteristics of the control loop with respect to its robustness and performance, therefore addressing the problem of determining which tuning method is best suited to the particular problem at hand. An exposition of tuning rules for integral and unstable systems follows. In this chapter, A.S. Rao and M. Chidambaram introduce different design methods for PI/PID controllers for these systems. Different approaches are considered. In particular, analytical, IMC, pole placement and optimization methods. The advantage of using a two degree-of-freedom controller will also be addressed as well as the robustness of the controllers. Robustness is specifically addressed in Chap. 4 by R. Vilanova, V. Alfaro and O. Arrieta by revisiting the new trends in robust PI/PID tuning. This chapter reviews the main concepts and measures for robustness in feedback systems and reviews the classical and modern approaches to robust PID design. The chapter also presents a series of new tuning rules that appeared in the literature during recent years that are based on the use of the maximum of the sensitivity transfer function as a robustness measure. The attainment of such robustness specification turns out to be a challenge that suggests alternative frameworks for robust PID controller tuning. As a matter of fact, one of the interests that has appeared in recent years is that of formulating tuning approaches in the form of simple tuning rules. In that respect, S. Skogestad and C. Grimholt present the Simple Internal Model Control approach by addressing the choice of the tuning parameters for fast and smooth control. In addition, their novel set-point overshoot method will be presented as an alternative approach. This first part of the book concludes with a chapter devoted to the control of MIMO systems. Q.G. Wang and Z.Y. Nie discuss the specific problems that appear when a MIMO system is to be controlled by using PID controllers. With a special emphasis on robustness, PID controller design for MIMO processes to achieve the desired gain and phase margins are presented. Tuning of decentralized MIMO PID controllers based on such margins is presented. In this way, the robust stability of the multivariable system can be readily achieved and guaranteed.

In the second part of the book, attention is turned to special control structures frequently used in conjunction with PID controllers for the achievement of specific purposes. Three chapters are devoted to such approaches. In the first chapter of this section, J.L. Guzmán, T. Hagglund and A. Visioli expose the use of feedforward compensation in conjunction with a feedback PID controller. Both set-point following and load disturbance rejection tasks are considered. While for the set-point following the generation of causal and non-causal feedforward actions are considered, for the load disturbance it is shown that both controllers (feedback and feedforward should cooperate). The use of alternative feedback compensation schemes such as the cascade control system for improving disturbance rejection is well known. In Chap. 8, S. Majhi presents different approaches for such series-feedback compensation schemes as well as dead-time compensation systems for use when there is a large time delay in the system. This part ends by a chapter that presents considerations for multi-input multi-output processes by addressing the control problem as a multivariable control problem per se. In this respect, R. Katebi presents and compares existing multi-loop tuning methods for their stability and performance robustness and formulates new design guidelines to improve their closed-loop robustness.

The third part of the book presents a range of issues related to the application of a PID controller. In this respect, this part starts with the contribution of K. Tsakalis and S. Dash that addresses issues arising in system identification-based plant modeling for the purpose of tuning PID controllers. Minimal and maximal process model information methods are considered. Especially attractive are recent methods that provide several nominal models as well as a description of the uncertainty, and aim for a tuning that combines high performance, adequate robustness, and high reliability. Besides finding of a good model, the nominal stability of the feedback control system is always the central and first aspect to be considered. When dealing with a restricted structure controller such as a PI/PID, the problem becomes more difficult. In Chap. 11, L.H. Keel and S.P. Bhattacharyya provide a design approach based on the determination of the entire set of PID controllers that stabilizes a given plant. In addition, it is shown that the entire set of stabilizing PID controllers for a given plant can be found without an analytical model being available. This chapter also shows that the complete set of stabilizing PID controllers for a finite dimensional LTI plant, possibly cascaded with a delay, can be calculated directly from the frequency response (Nyquist/Bode) data. Once the controller is obtained, there are two different aspects that come into the scenario: those of the fragility of the resulting tuning and performance assessment. Then next two chapters deal, respectively, with these aspects. In Chap. 12, V.M. Alfaro and R. Vilanova provide an introduction to fragility measures for PID controllers. Subsequently, the fragility of PID controllers tuned with several of the available performance optimized and/or robust tuning rules will be evaluated using the delta 20 fragility index. The introduction of the fragility as part of the development process of a robust tuning rule for PID controllers is also considered. Considerations on performance assessment follow in Chap. 13, authored by A.W. Ordys and M.J. Grimble. Since many PID controllers are set up using intuition or very approximate tuning rules, it is even more important that PID designs can be benchmarked and the quality of control assessed. Furthermore, benchmarking methods can provide guidance for controller tuning. One of the advances made in the last decade has been the development of the so-called restricted structure benchmarking which provides a figure of merit which is much more representative of what might be achievable if the controller is tuned optimally. The use of these methods for controller tuning is also discussed. This third part ends with G.K. McMillan's contribution that introduces industrial considerations for PID based control loops. Different process control applications are addressed in this chapter, and it is shown how to deal with them when using a PID controller. In particular, challenging applications (namely, with the presence of high valve stiction, large wireless refresh times, high process nonlinearity and dead-time, multiple process constraints, abnormal operations and communication failures, to name a few) are discussed. Then, process control operations such as bioreaction, chemical reaction, crystallization, distillation, evaporation, neutralization and compression are considered.

The last part of the book comprises four chapters devoted to non-standard approaches to PID control. They constitute novel approaches of PID control that have recently appeared in the PID field. A (now) very popular theme is that of fractional-order PID control. B.M. Vinagre and C.A. Monje present the main characteristics of Fractional-order Control (FOC), an approach that has attracted a growing interest in the last decade. The application of the fractional order operators to the PID algorithm, thus giving the fractional-order PID (FOPID) controller, is first introduced. Then, the FOPID controller is studied in both the frequency and the time domains, and the structures, the tuning rules, and the ways for their implementation proposed in the literature will be reviewed and discussed, as well as their practical applications. Another different perspective for addressing the controller design problem is that of event based control. J. Sánchez, A. Visioli and S. Dormido present the basic concepts of event-based control where feedback control actions are computed when the process output is outside a certain detection band located around the set-point value; and once the process is inside the detection band, new control actions are not produced until the process leaves the region as a consequence of disturbances or a change of the set-point value. The chapter starts with a description of the first event-based PI controller published in the literature and continues by describing the evolution of this type of controller, to finish with the most recent implementations, as, for instance, a 2-DOF pure event-based PI controller. As another non-standard approach to obtaining a PID controller we have the data-driven, or also called model-free, methods. T. Yamamoto introduces the concept of data-driven (DD) controllers where a suitable set of PID parameters is automatically generated based on input/output data pairs of the controlled object stored in the database. This scheme can adjust the PID parameters in an on-line manner even if the system has nonlinear properties and/or time-variant system parameters. The fourth part of the book ends by introducing some considerations on predictive control approaches for PID control design. In this case, the design of PID control systems based on advanced control, e.g., generalized minimum variance control and generalized predictive control is described. In this chapter, generalized predictive control is attained by PID control by considering the GPC control law approximation and introducing the considerations needed in order to deal with future information. Finally, to obtain further high performance, predictive control based PID systems are extended to multirate systems.

The methodologies considered in this book are presented in order to highlight the theoretical and the implementation issues, so that they are clearly characterized both from an academic and industrial perspective. The book can therefore serve as a reference and source book for academic researchers who will consider it also as a stimulus for new ideas as well as for industrial practitioners and manufacturers of control systems who will find appropriate advanced solutions to their application problems.

Barcelona, Spain Ramon Vilanova
Brescia, Italy Antonio Visioli

Contents

Part I Tuning for PID Controllers

1 An Overview of Tuning Rules for the PI and PID Continuous-Time Control of Time-Delayed Single-Input, Single-Output (SISO) Processes 3
Aidan O'Dwyer

2 Model-Based PI(D) Autotuning 45
Alberto Leva and Martina Maggio

3 PI/PID Controllers Design for Integrating and Unstable Systems . . 75
A. Seshagiri Rao and M. Chidambaram

4 Robustness in PID Control 113
Ramon Vilanova, Víctor M. Alfaro, and Orlando Arrieta

5 The SIMC Method for Smooth PID Controller Tuning 147
Sigurd Skogestad and Chriss Grimholt

6 PID Control for MIMO Processes 177
Qing-Guo Wang and Zhuo-Yun Nie

Part II Control Structures and Configurations for PID Control

7 Feedforward Compensation for PID Control Loops 207
José Luis Guzmán, Tore Hägglund, and Antonio Visioli

8 Control Structures for Time Delay Systems 235
Somanath Majhi

9 Robust Multivariable Tuning Methods 255
Reza Katebi

Part III Issues in PID Control Systems

10 Identification for PID Control . . . 283
Kostas S. Tsakalis and Sachi Dash

11 Modern PID Control: Stabilizing Sets and Multiple Performance Specifications . . . 319
L.H. Keel and S.P. Bhattacharyya

12 Fragility Evaluation of PI and PID Controllers Tuning Rules . . . 349
Víctor M. Alfaro and Ramon Vilanova

13 Benchmarking and Tuning PID Controllers . . . 381
Andrew W. Ordys and Michael J. Grimble

14 Industrial Applications of PID Control . . . 415
Gregory K. McMillan

Part IV Non-standard Approaches to PID

15 Fractional-Order PID . . . 465
Blas M. Vinagre and Concepción A. Monje

16 Event-Based PID Control . . . 495
José Sánchez, Antonio Visioli, and Sebastián Dormido

17 Data-Driven PID Controller . . . 527
Toru Yamamoto

18 Predictive Control Approaches for PID Control Design and Its Extension to Multirate System . . . 553
Takao Sato

Index . . . 597

Part I
Tuning for PID Controllers

Chapter 1
An Overview of Tuning Rules for the PI and PID Continuous-Time Control of Time-Delayed Single-Input, Single-Output (SISO) Processes

Aidan O'Dwyer

1.1 Introduction

A time delay may be defined as the time interval between the start of an event at one point in a system and its resulting action at another point in the system. Delays are also known as transport lags or dead times; they arise in physical, chemical, biological and economic systems, as well as in the process of measurement and computation. Methods for the compensation of time-delayed processes may be broadly divided into parameter optimised controllers, such as *proportional-integral* (PI) or *proportional-integral-derivative* (PID) controllers, in which the controller parameters are adapted to the controller structure, and structurally optimised controllers, in which the controller structure and parameters are adapted optimally to the structure and parameters of the process model.

PI and PID controllers have been at the heart of control engineering practice for over seven decades and were suggested as the second most important control decision and communication instrument of the 20th century [483]. Historically, the first *tuning rule* (formula) for setting up controller parameters was defined in 1934 for the design of a proportional-derivative (PD) controller for a process exactly modelled by an *integrator plus delay* (IPD) model [84]. Subsequently, tuning rules were defined for PI and PID controllers, assuming that the process was exactly modelled by a *first-order lag plus delay* (FOLPD) model [86] or a pure delay model [86, 217].

The use of the PI or PID controller is ubiquitous in industry. It has been stated, for example, that in process control applications, more than 95% of the controllers are of PI or PID type [44, 65, 223, 296, 366, 576]. However, despite this development work, surveys indicating the state-of-the-art of control industrial practice report sobering results. For example, in testing of thousands of control loops in

A. O'Dwyer (✉)
School of Electrical Engineering Systems, Dublin Institute of Technology, Kevin St., Dublin 8, Ireland
e-mail: aidan.odwyer@dit.ie

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_1, © Springer-Verlag London Limited 2012

hundreds of plants, it has been found that more than 30% of installed PI or PID controllers are operating in manual mode and 65% of loops operating in automatic mode produce less variance in manual than in automatic (thus, the automatic controllers are poorly tuned) [156]. In another interesting study of 150,000 control loops at over 250 industrial sites around the globe, it was shown that 68% of all controllers had unacceptable performance; even the best site has only 70% of all controllers performing acceptably, while the worst site had 15% of all controllers performing acceptably [193]. Other literature [602] claims that "extensive industry testing" shows that 75% of all PID-based loops are out of tune. A survey of paper processing mills is quoted, in which 60% of the 36 mills surveyed stated that less than half of their control loops were well tuned (the majority of the mills reported that they had between 2000 and 4000 regulatory control loops). In a further such comment, it is claimed [157] that only 20% of all control loops surveyed in mill audits have been found to actually reduce process variability in automatic mode over the short term. Of the problem loops, increased process variability in automatic mode could be ascribed specifically to controller tuning problems in approximately 30% of cases. Many of the points made above are re-iterated by [680]. The situation has not improved more recently, with [609] reporting that 80% of PID controllers are badly tuned; 30% of PID controllers operate in manual with another 30% of the controlled loops increasing the short-term variability of the process to be controlled (typically due to too strong integral action). It is stated that 25% of all PID controller loops use default factory settings, implying that they have not been tuned at all.

Poor controller tuning is surprising, as very many tuning rules exist to allow the specification of the controller parameters. Tuning rules have the advantage of ease of calculation of the controller parameters (when compared to more analytical controller design methods), on the one hand; on the other hand, the use of tuning rules is a good alternative to trial and error tuning. It is clear that the many controller tuning rules proposed in the literature are not having an impact on industrial practice. One reason is that the tuning rules are not very accessible, being scattered throughout the control literature; in addition, the notation used is not unified. In a book published in 2003 [418], tuning rules for continuous-time PI and PID control of single-input, single-output (SISO) processes, with time delay, have been compiled and summarised, using a unified notation. A second edition of this book was published in 2006 [419], with a third edition published in 2009 [420]. In the third edition of the book, a total of 1731 tuning rules were compiled; of the tuning rules, 60% were specified for a self-regulating process model, 30% were specified for a non-self-regulating process model, and the remaining 10% were non-model specific tuning rules. Tuning rules continue to be specified since the third edition of the book was published.

The purpose of this chapter is to provide an outline of the tuning rules specified. First, a brief summary of the range of PI and PID controller structures proposed in the literature, together with the process models used to define the controller tuning rules, is provided. Then, controller architecture and process modelling issues are outlined. Finally, conclusions are drawn. Other reviews are recommended to the interested reader [21, 41, 45, 48–50, 80, 94, 132, 167, 190, 210, 296, 320, 329, 346, 347, 414, 417, 503, 581, 601].

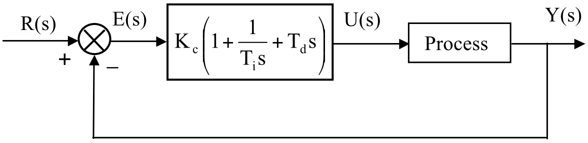


Fig. 1.1 Ideal PID controller in a unity feedback block diagram representation. This controller structure, and an equivalent structure, is also labelled the parallel, ideal parallel, non-interacting, parallel non-interacting, independent, gain independent or ISA controller [346]. 406 tuning rules have been identified for this controller structure [420]

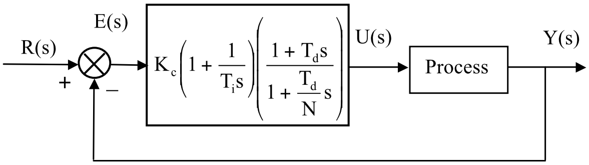

R(s)
E(s)
$K_c\left(1+\frac{1}{T_i s}\right)\left(\frac{1+T_d s}{1+\frac{T_d}{N}s}\right)$
U(s)
Process
Y(s)
+
–

Fig. 1.2 Classical PID controller in a unity feedback block diagram representation. Also labelled the cascade, interacting, series, interactive, rate-before-reset or analog controller [346]. 183 tuning rules have been identified for this controller structure [420]

1.2 Controller Architecture and Process Modelling

A practical difficulty with PID control technology is a lack of industrial standards, which has resulted in a wide variety of PID controller architectures. Nine different PI/PID controller structures have been identified [420]. Controller manufacturers vary in their choice of architecture; controller tuning that works well on one architecture may work poorly on another. Considering the PID controller, common architectures are:

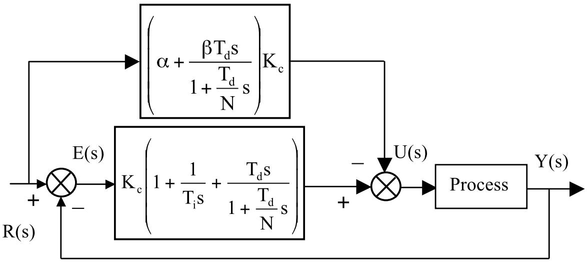
1. The 'ideal' PID controller (Fig. 1.1), given by

$$G_c(s) = K_c\left(1+\frac{1}{T_i s}+T_d s\right).$$

   This architecture is used, for example, on the Honeywell TDC3000 Process Manager Type A, non-interactive mode product [262].
2. The 'classical' PID controller (Fig. 1.2), given by

$$G_c(s) = K_c\left(1+\frac{1}{T_i s}\right)\frac{1+sT_d}{1+s\frac{T_d}{N}}.$$

   This architecture is used, for example, on the Honeywell TDC3000 Process Manager Type A, interactive mode product [262].



$\left(\alpha + \frac{\beta T_d s}{1+\frac{T_d}{N}s}\right)K_c$
E(s)
U(s)
Y(s)
$K_c\left(1+\frac{1}{T_i s}+\frac{T_d s}{1+\frac{T_d}{N}s}\right)$
Process
+
−
R(s)
−
+

Fig. 1.3 Two-degree-of-freedom controller 1, in a unity feedback block diagram representation. Also labelled the 'm-PID' controller [246], the 'ISA-PID' controller [338] and the 'P-I-PD (only P is DOF) incomplete 2DOF algorithm' [397]. 276 tuning rules have been identified for this controller structure [420]

3. The two-degree-of-freedom controller 1 (Fig. 1.3), given by

$$U(s) = K_c\left([1-\alpha] + \frac{1}{T_i s} + \frac{[1-\beta]T_d s}{1+\frac{T_d}{N}s}\right)R(s) - K_c\left(1 + \frac{1}{T_i s} + \frac{T_d s}{1+\frac{T_d}{N}s}\right)Y(s).$$

This architecture is used, for example, on the Omron E5CK digital controller with $\beta = 1$ and $N = 3$ [262].

The most dominant PI controller architecture is the 'ideal' PI controller; 563 tuning rules have been identified for this controller structure [420]. The controller transfer function is given by

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

The wide variety of controller architectures is mirrored by the wide variety of ways in which processes with time delay may be modelled. Common models are:

1. Self-regulating FOLPD model, given by

$$G_m(s) = \frac{K_m e^{-s\tau_m}}{1+sT_m}.$$

2. Self-regulating *second-order system plus time delay* (SOSPD) model, given by

$$G_m(s) = \frac{K_m e^{-s\tau_m}}{T_{m1}^2 s^2 + 2\xi_m T_{m1} s + 1} \quad \text{or} \quad G_m(s) = \frac{K_m e^{-s\tau_m}}{(1+T_{m1}s)(1+T_{m2}s)}.$$

Non-self-regulating IPD model, given by

$$G_m(s) = \frac{K_m e^{-s\tau_m}}{s}.$$

Non-self-regulating *first-order lag plus integral plus delay* (FOLIPD) model, given by

$$G_m(s) = \frac{K_m e^{-s\tau_m}}{s(1+sT_m)}.$$

Of course, the modelling strategy used influences the value of the model parameters, which, in turn, affect the controller values determined from the tuning rules. Fifty-nine modelling strategies have been detailed to determine the parameters of the FOLPD process model, for example. Space does not permit a full discussion of this issue; further details are provided in [420].

1.3 Tuning Rules for PI and PID Controllers

To review the action of the PID controller, consider the ideal PID controller, given by

$$G_c(s) = K_c\left(1 + \frac{1}{T_i s} + T_d s\right),$$

with $K_c =$ proportional gain, $T_i =$ integral time constant and $T_d =$ derivative time constant. If $T_i = \infty$ and $T_d = 0$ (that is, P control), then the closed-loop measured value is always less than the desired value for processes without an integrator term, as a positive error is necessary to keep the measured value constant, and less than the desired value. The introduction of integral action facilitates the achievement of equality between the measured value and the desired value, as a constant error produces an increasing controller output. The introduction of derivative action means that changes in the desired value may be anticipated, and thus an appropriate correction may be added prior to the actual change. Thus, in simplified terms, the PID controller allows contributions from present, past and future controller inputs.

PI and PID controller tuning rules may be broadly classified as follows:

- Process reaction curve tuning rules.
- Tuning rules based on minimising an appropriate performance criterion.
- Direct synthesis tuning rules.
- Tuning rules for robustness.
- Ultimate cycle tuning rules.

Tuning rules in the first four subdivisions are typically based on process model parameters; the development of a process model is typically not required for using tuning rules in the final subdivision. Some tuning rules could be considered to belong to more than one subdivision, so the subdivisions cannot be considered to be mutually exclusive; nevertheless, they provide a convenient way to classify the rules. An outline of tuning rules in these subdivisions is now provided.

1.3.1 Process Reaction Curve Tuning Rules

Process reaction curve tuning rules are based on process model data determined from a measured open-loop step response. The first (and most well-known) tuning rule of this type was suggested in 1942 [706]; in this method, the process is modelled by a FOLPD process model with the model parameters estimated using a tangent and point method. Simple formulae are used to define tuning parameters for PI and PID controllers. The PI controller settings are given by

$$K_c = \frac{0.9T_m}{K_m \tau_m}, \qquad T_i = 3.33\tau_m.$$

The (ideal) PID controller settings are given by

$$K_c \in \left[\frac{1.2T_m}{K_m \tau_m}, \frac{2T_m}{K_m \tau_m}\right], \qquad T_i = 2\tau_m, \qquad T_d = 0.5\tau_m.$$

Other process reaction curve tuning rules are also described, sometimes in graphical form, to control self-regulating processes modelled in:

- Pure delay form [217, 665].
- FOLPD form [2, 8, 43, 72, 116, 125, 131, 133, 134, 140, 163, 166, 216, 219–221, 226, 295, 301, 350, 391, 403, 405, 416, 425, 426, 482, 497, 499, 535, 536, 565, 581, 608, 662, 665].
- SOSPD form [8, 54].

Process reaction curve tuning rules are also described to control non-self-regulating processes modelled in:

- IPD form [8, 81, 84, 135, 136, 170, 171, 219, 408, 665, 706].
- FOLIPD form [135, 136].

The advantage of process reaction curve tuning strategies is that only a single experimental test is typically necessary. However, the disadvantages of the strategy are primarily based on the difficulty, in practice, of obtaining an accurate process model; for example, load changes may occur during the test which may distort the test results, and a large step input may be necessary to achieve a good signal-to-noise ratio [504]. Similar disadvantages will arise in any tuning method dependent on prior model development.

1.3.2 Tuning Rules Based on Minimising an Appropriate Performance Criterion

Tuning rules based on minimising an appropriate performance criterion are typically specified either for optimum regulator or optimum servo action (in closed loop). Performance criteria, such as the minimisation of the integral of absolute error (IAE) in a closed-loop environment, may be used to determine a unique set of controller parameter values.

1.3.2.1 Optimising Regulator Response

Tuning rules have been described, sometimes in graphical form, to optimise the regulator response of a compensated self-regulating process, modelled in:

- Pure delay form [180, 183, 197, 532–534].
- FOLPD form [9, 28, 30, 31, 104, 105, 144, 151, 153, 155, 165, 168, 172, 180, 182, 197, 208, 214, 220, 225, 231, 232, 238, 239, 255, 257, 273, 280, 298–300, 357, 368, 382, 400, 404, 405, 413, 427, 445, 450, 452, 457, 461, 532–534, 537, 551, 570, 585, 662, 679, 704, 705], with a zero [547].
- SOSPD form [68, 92, 197, 214, 218, 238, 239, 255, 357, 358, 368, 389, 394, 404, 443, 525, 526, 532–534, 567, 660], with a zero [368].
- Stable higher-order plus delay form (orders 3, 4 and 5) [404].

Tuning rules have also been described to optimise the regulator response of a compensated non-self-regulating process, modelled in:

- IPD form [38, 197, 220, 467, 532–534, 585, 614].
- FOLIPD form [39, 71, 197, 467, 468, 533, 610].
- Unstable FOLPD form [103, 237, 532, 614].
- Unstable SOSPD form with one unstable pole [237, 467, 468].

1.3.2.2 Optimising Servo Response

Similarly, tuning rules have been proposed to optimise the servo response of a compensated self-regulating process, modelled in:

- Pure delay form [242, 283].
- FOLPD form [9, 27, 28, 30, 31, 56, 57, 105, 144, 165, 175, 177, 214, 231, 232, 238, 239, 242, 255, 257, 280, 284, 294, 368, 371, 382, 404, 427, 494, 496, 549–551, 585–587, 589, 643, 649, 662, 666, 684, 699, 704, 705].
- SOSPD form [26, 92, 175, 238, 239, 255, 283, 368, 383, 394, 404, 443, 553, 567, 659, 660], with a zero [368, 643, 649].
- Stable higher-order plus delay form (orders 3, 4, and 5) [404].

Tuning rules have also been described to optimise the servo response of a compensated non-self-regulating process, modelled in:

- IPD form [18, 38, 446, 585, 614].
- FOLIPD form [18, 39, 610].
- Squared integral plus delay (I^2PD) form [18].
- Unstable FOLPD form [103, 237, 373, 614].
- Unstable SOSPD form with one unstable pole [237].

1.3.2.3 Other Approaches

Other tuning rules attempt to simultaneously optimise both the servo and regulator response of a closed-loop system, or adopt other performance minimisation approaches. Such tuning rules have been proposed when the self-regulating process is modelled in:

- Pure delay form [47, 541, 542].

- FOLPD form [2, 23, 29, 32–34, 47, 160, 165, 243, 270, 407, 506, 527, 532, 533, 572–574, 588, 590, 661].
- SOSPD form [243, 281, 400, 447–449, 553, 574, 661].
- Third-order system plus delay (TOSPD) form [464, 572].

Such tuning rules are also described to control non-self-regulating processes modelled in:

- IPD form [7, 29, 35, 47, 407, 447–449, 541, 542, 572, 574, 575, 588, 590].
- FOLIPD form [158, 161, 447–449, 572–575].
- SOSIPD form [447, 449, 572].
- Second-order system plus integral plus delay (SOSIPD) form [448, 575].
- Unstable FOLPD form [29, 35, 265, 441].

1.3.3 Direct Synthesis Tuning Rules

Direct synthesis tuning rules are those that allow a specified closed-loop response. Such tuning rules may be specified using a time domain metric, such as achieving the desired poles of the closed-loop response, or a frequency domain metric, such as achieving a specified gain margin and/or phase margin.

1.3.3.1 Achieving Time Domain Metrics

Tuning rules to achieve time domain metrics are defined to compensate self-regulating processes modelled in:

- Pure delay form [48, 76, 212, 272, 283, 313, 379, 409, 481], with a zero [116].
- FOLPD form [3, 10–12, 51, 56, 59, 63, 64, 76, 87, 95, 113, 116, 119, 123, 126, 130, 143, 145, 162, 169, 174, 187–189, 191, 198, 199, 206, 246, 248, 272, 275, 277, 283, 284, 286, 289, 292, 293, 299, 311–313, 348, 372, 377, 388, 392, 395, 398, 399, 409, 410, 412, 422–424, 432, 460, 474, 481, 498, 500, 502, 530, 538, 539, 542, 546, 549, 551, 552, 562, 563, 568, 569, 595–597, 604, 605, 617, 618, 620–622, 625, 626, 628, 641, 642, 656, 674, 685], with a zero [116, 547, 555].
- SOSPD form [10, 12, 13, 36, 51, 64, 73, 76, 95, 116, 126, 149, 192, 202, 211, 235, 236, 246, 248, 313, 348, 357, 372, 379, 399, 439, 450, 451, 465, 481, 492, 500, 538, 539, 542, 544–546, 552, 568, 572, 595, 618–620, 625, 626, 640, 659], with a zero [116, 121, 124, 246, 345, 465, 685].
- TOSPD form [376].
- General form, perhaps with a repeated pole [145, 192, 313, 316, 323, 477, 595].

Such tuning rules are also described to control non-self-regulating processes modelled in:

- IPD form [38, 61, 95, 114, 118, 123, 129, 174, 189, 212, 219, 250, 252–254, 313, 331, 410, 411, 469, 492, 542, 544, 560, 561, 564, 618, 623–625, 640], with a zero [116].

- FOLIPD form [39, 95, 96, 158, 161, 250, 254, 313, 463, 469, 542, 544, 561, 571, 596, 597, 599, 618, 625], with a zero [22, 116, 436].
- I^2PD form [212, 251, 542, 544, 683].
- General form with an integrator [313].
- Unstable FOLPD form [51, 52, 89, 103, 111, 115, 116, 240, 241, 263, 265, 372, 379, 441, 471, 559, 561–563, 606, 627], with a zero [556, 557, 561].
- Unstable SOSPD form with one unstable pole [51, 103, 240, 241, 372, 387, 561, 643], two unstable poles [387, 643] and with a zero [478, 479, 558, 561].

1.3.3.2 Achieving Frequency Domain Metrics

Tuning rules to achieve specific frequency domain metrics are also described for self-regulating processes modelled in:

- Pure delay form [48, 77, 202, 501], with a zero [271].
- FOLPD form [5, 14, 15, 48, 55, 69, 75, 91, 98, 100, 101, 109, 128, 129, 142, 144, 159, 173, 189, 191, 196, 201, 202, 207, 208, 222, 232, 235, 244, 246, 249, 287, 295, 302, 307, 308, 317, 318, 324, 333, 336, 340, 341, 343, 344, 384, 402, 415, 433, 434, 466, 500, 511, 548, 554, 577, 628, 641, 642, 645, 646, 652, 658, 672, 673, 675, 705], with a zero [150, 271, 380].
- SOSPD form [36, 48, 101, 150, 207, 209, 228–230, 235, 236, 244, 246, 249, 328, 341, 343, 374, 375, 415, 470, 500, 554, 577, 636, 647, 651, 675], with a zero [150, 246, 271, 285, 415, 470, 644, 653, 654].
- TOSPD form [235, 500, 629, 631, 633, 634].
- Fifth order model with delay form [629, 630, 632, 633].
- General form [500, 540].
- Non-model specific form [48, 58, 70, 173, 202, 307–309, 396, 566, 583, 629, 631, 650, 677].

Such tuning rules are also described to control non-self-regulating processes modelled in:

- IPD form [48, 78, 100, 106, 108, 112, 128, 189, 201, 202, 235, 236, 278, 279, 290, 297, 306, 352, 353, 375, 385, 415, 453, 480], with a zero [271].
- FOLIPD form [48, 100, 235, 290, 306, 308, 385, 415, 470, 480, 518, 638, 639, 643, 648, 655, 667, 669–671], with a zero [271].
- General model with integrator [359].
- I^2PD form [48, 480].
- First-order lag plus squared integral plus delay (FOLI^{2}PD) form [480].
- Unstable FOLPD form [17, 37, 110, 112, 148, 176, 201, 227, 363, 433, 442, 472, 561, 611, 639, 668, 701].
- Unstable FOLIPD form [480].
- Unstable SOSPD form with one unstable pole [117, 227, 314, 442, 470].

1.3.4 Tuning Rules for Robustness

Tuning rules in this category are specified to allow the resulting closed-loop control system to achieve a robust stability and/or a robust performance criterion. Tuning rules have been specified for the compensation of self-regulating processes modelled in:

- Pure delay form [48, 62, 186, 542, 603, 607, 681].
- FOLPD form [6, 16, 20, 25, 29, 48, 62, 71, 74, 99, 120, 122, 127, 137, 138, 146, 163, 181, 184–186, 215, 233, 234, 245, 247, 261, 266, 276, 321, 322, 325, 328, 333–335, 337, 339, 342, 343, 365, 369, 401, 421, 428, 437, 458, 476, 488, 489, 507, 513, 514, 517, 518, 520, 522, 528, 549, 579, 584, 591, 607, 612, 613, 615, 616, 637, 681, 686, 689, 695–697, 702, 708], with a zero [90, 328, 507].
- SOSPD form [16, 71, 74, 194, 207, 245, 264, 266, 302, 319, 321, 325, 328, 380, 406, 437, 439, 458, 488, 507, 513, 516, 517, 520, 522, 529, 550, 607, 682, 686, 697, 702], with a zero [102, 120, 122, 328, 380, 462, 507, 516, 593].
- TOSPD form [268, 380, 437].
- General form with a repeated pole [316].

Tuning rules have been specified for the compensation of non-self-regulating processes modelled in:

- IPD form [19, 20, 24, 38, 73, 95, 120, 122, 321, 328, 421, 429, 430, 440, 484–488, 510, 513, 515, 520, 522, 524, 543, 549, 579, 591, 657, 681, 686, 691, 692, 694, 700, 702, 707, 708].
- FOLIPD form [120, 122, 321, 328, 355, 429, 437, 488, 513, 515, 516, 520, 522, 578, 579, 599, 600, 607, 610, 692], with a zero [22, 328, 437, 438, 516, 520, 522].
- I^2PD form [355].
- Unstable FOLPD form [111, 321, 326, 328, 381, 444, 493, 508, 509, 513, 518, 519, 524, 561, 580, 686, 687, 693], with a zero [561].
- Unstable FOLIPD form [438, 523, 688], with a zero [516, 523].
- Unstable SOSPD form with one unstable pole [326, 328, 381, 493, 513, 518, 519, 523, 688], two unstable poles [326, 328, 512, 521, 523], with a zero [516, 561], with one unstable pole and a zero [523] and with two unstable poles and a zero [438, 523, 561].

1.3.5 Ultimate Cycle Tuning Rules

Ultimate cycle tuning rules are based on recording appropriate parameters at the ultimate frequency (i.e. the frequency at which marginal stability of the closed-loop control system occurs). The first such tuning rule was defined in 1942 [706] for the tuning of P, PI and PID controller parameters of a process that may or may not include a delay. Briefly, the experimental technique is as follows:

(a) Place the controller in proportional mode only.

(b) Increase K_c until the closed-loop system output goes marginally stable; record K_c (calling it K_u, the *ultimate gain*) and the *ultimate period*, T_u.

Simple formulae are used to define tuning parameters for PI and PID controllers. The PI controller settings are

$$K_c = 0.45K_u, \qquad T_i = 0.83T_u,$$

with the (ideal) PID controller settings given by

$$K_c \in [0.6K_u, K_u], \qquad T_i = 0.5T_u, \qquad T_d = 0.125T_u.$$

The tuning rules implicitly build an adequate frequency domain stability margin into the compensated system [147]. However, there are a number of disadvantages to the ultimate cycle tuning approach:

- The system must generally be destabilised under proportional control.
- The empirical nature of the method means that uniform performance is not achieved in general [258].
- Several trials must typically be made to determine the ultimate gain.
- The resulting process upsets may be detrimental to product quality.
- There is a danger of misinterpreting a limit cycle as representing the stability limit [457].
- The amplitude of the process variable signal may be so great that the experiment may not be carried out for cost or safety considerations.

Some of these disadvantages are addressed by defining modifications of the rules in which, for example, the proportional gain in the experiment is set up to give a closed-loop transient response decay ratio of 0.25, or a phase lag of 135°; ultimate cycle tuning rules, and their modifications, may compensate self-regulating processes modelled in:

- Pure delay form [393].
- FOLPD form [58, 67, 82, 179, 195, 202, 203, 205, 206, 224, 249, 313, 342, 351, 361, 386, 390, 454, 457, 473, 663, 664, 681, 698].
- SOSPD form [68, 249, 313, 315, 431, 592], with a zero [102].
- TOSPD form [313, 681].
- Other form [598].
- General, possibly delayed, stable form [1, 2, 4, 8, 48, 53, 66, 83, 88, 93, 97, 107, 139, 152, 154, 155, 164, 200, 208, 213, 216, 219, 256, 260, 267, 269, 274, 288, 291, 302, 310, 327, 349, 356, 360, 366, 367, 370, 378, 389, 391, 392, 408, 416, 425, 435, 445, 455–457, 459, 475, 490, 495, 550, 565, 581, 582, 594, 608, 635, 659, 664, 676, 678, 680, 681], sometimes to achieve a specified frequency domain metric [40, 42, 43, 79, 141, 147, 178, 202, 204, 259, 282, 302–306, 330, 332, 491, 505, 527, 531, 690].

Tuning rules have been specified for the compensation of non-self-regulating processes modelled in:

- IPD form [60, 354, 362, 681, 703].

- FOLIPD form [313, 390, 431, 454, 599, 600].
- SOSIPD form [313], with a zero [364].
- Third order lag plus integral plus delay (TOLIPD) form [313].
- General, possibly delayed, non-self-regulating form [356, 393].
- Unstable SOSPD form with one unstable pole [390].

1.4 Conclusions and Future Perspectives

Control academics and practitioners remain interested in the use of PI and PID controllers to compensate processes with time delay. This chapter outlines the work done in tuning rule development for such processes. The most startling statistic to emerge from the complete work is the quantity of tuning rules identified; to mid 2008, 563 PI tuning rules and 1168 PID tuning rules have been collated, a total of 1731 separate rules. Recent years have seen acceleration in the accumulation of tuning rules.

It is difficult to detect an overall trend in the research, in terms of the categories used by the author, as the decades of development work completed continues to be influential. However, the recent increased emphasis on the incorporation of robustness metrics in the specification of tuning rules is of interest. From a controller architecture perspective, there is a trend to specify tuning rules for two-degree-of-freedom controllers, which may reflect the greater interest in the use of such controllers by practitioners [420]. From a process perspective, there has been more interest in the development of tuning rules to control non-self-regulating processes in recent years, particularly those which are open-loop unstable; since one of the first tuning rules for these processes was proposed twenty-two years ago [148], over 180 further tuning rules have been proposed for these applications [420]. Finally, there is increasing interest in the specification of tuning rules for fractional-order PI or PID controllers, and for PI or PID controllers that may be embedded in applications such as cascade control systems and time-delay compensators, topics which are outside the scope of this review.

Reflecting on the PI and PID controller tuning rules reported in the chapter, there is a lack of comparative analysis regarding the performance and robustness of closed-loop systems compensated with the associated controllers; the absence of benchmark processes for testing, at least until relatively recently [46], is also notable. The main priority for future research in the area should be a critical analysis of available tuning rules, rather than the proposal of further tuning rules.

Historical Note The 75th anniversary of the receipt of the first technical paper describing tuning rules for setting up controller parameters [86] is presently being marked. The paper was received by the Philosophical Transactions of the Royal Society of London on July 15, 1935; the paper was received, in revised form, on November 26, 1935, and was read on February 2, 1936. The lead author of the paper subsequently took out a patent on the PID controller [85].

References

1. ABB: Operating Guide for Commander 300/310, Sect. 7 (1996)
2. ABB: Instruction manual for 53SL6000. Document: PN24991.pdf (2001). Available at www.abb.com. Cited 1 September 2004
3. Abbas, A.: A new set of controller tuning relations. ISA Trans. **36**, 183–187 (1997)
4. Aikman, A.R.: The frequency response approach to automatic control problems. Trans. Soc. Instrum. Technol. 2–16 (1950)
5. Alcántara, S., Pedret, C., Vilanova, R., Zhang, W.D.: Setpoint-oriented robust PID tuning from a simple min-max model matching specification. In: Proc. IEEE Conference on Emerging Technologies and Factory Automation, Mallorca, Spain, pp. 1–8 (2009)
6. Alcántara, S., Pedret, C., Vilanova, R.: On the model matching approach to PID design: analytical perspective for robust servo/regulator tradeoff tuning. J. Process Control **20**, 596–608 (2010)
7. Alenany, A., Abdelrahman, O., Ziedan, I.: Simple tuning rules of PID controllers for integrator/dead time processes. In: Proc. International Conference for Global Science and Technol., Cairo, Egypt (2005). Available at www.icgst.com/ACSE05/papers/P1110504001.pdf. Cited 4 January 2011
8. Alfaro, V.M.: Actualización del método de sintonización de controladores de Ziegler y Nichols. Ingénierie **15**(1–2), 39–52 (2005) (in Spanish)
9. Alfaro, V.M.: Estimación del desempeño IAE de los reguladores y servomecanismos PID. Ingénierie **15**(1), 79–90 (2005) (in Spanish)
10. Alfaro, V.M.: Analytical robust tuning of two-degree-of-freedom PI and PID controllers (ART_2) (2007). Available at http://www2.eie.ucr.ac.cr/~valfaro/docs/vma.art2.pdf. Cited 30 March 2009
11. Alfaro, V.M., Vilanova, R., Arrieta, O.: Analytical robust tuning of PI controllers for first-order-plus-dead-time processes. In: Proc. IEEE International Conference on Emerging Technologies and Factory Automation, Hamburg, Germany, pp. 273–280 (2008)
12. Alfaro, V.M., Vilanova, R., Arrieta, O.: Two-degree-of-freedom PI/PID controller tuning approach for smooth control on cascade control systems. In: Proc. 47th IEEE Conference on Decision and Control, Mexico, pp. 5680–5685 (2008)
13. Alfaro, V.M., Vilanova, R., Arrieta, O.: Robust tuning of two-degree-of-freedom (2-DoF) PI/PID based cascade control systems. J. Process Control **19**, 1658–1670 (2009)
14. Alfaro, V.M., Vilanova, R., Arrieta, O.: NORT: a non-oscillatory robust tuning approach for 2-DoF PI controllers. In: Proc. 18th IEEE International Conference on Control Applications, St. Petersburg, Russia, pp. 1003–1008 (2009)
15. Alfaro, V.M., Vilanova, R., Arrieta, O.: Maximum sensitivity based robust tuning for two-degree-of-freedom proportional-integral controllers. Ind. Eng. Chem. Res. **49**(11), 5415–5423 (2010)
16. Ali, A., Majhi, S.: PI/PID controller design based on IMC and percentage overshoot specification to controller setpoint change. ISA Trans. **48**, 10–15 (2009)
17. Ali, A., Majhi, S.: Controller design for unstable FOPTD plants based on sensitivity. In: Proc. IFAC World Congress, Seoul, Korea, pp. 5837–5841 (2009)
18. Ali, A., Majhi, S.: PID controller tuning for integrating processes. ISA Trans. **49**, 70–78 (2010)
19. Alvarez-Ramirez, J., Morales, A., Cervantes, I.: Robust proportional-integral control. Ind. Eng. Chem. Res. **37**, 4740–4747 (1998)
20. Andersson, M.: A MATLAB tool for rapid process identification and PID design. MSc thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden (2000)
21. Ang, K.H., Chong, G., Li, Y.: PID control system analysis, design and technology. IEEE Trans. Control Syst. Technol. **13**, 559–576 (2005)
22. Anil, C., Sree, R.P.: Design of PID controllers for FOPTD systems with an integrator and with/without a zero. Indian Chem. Eng., Sect. A **47**(4), 235–242 (2005)

23. Araki, M.: 2-degree of freedom control system. Syst. Control **29**, 649–656 (1985) (in Japanese)
24. Arbogast, J.E., Cooper, D.J.: Extension of IMC tuning correlations for non-self regulating (integrating) processes. ISA Trans. **46**, 303–311 (2007)
25. Arbogast, J.E., Cooper, D.J., Rice, R.C.: Model-based tuning methods for PID controllers (2006). Available at http://www.bin95.com/Model-Based%20Tuning%20Methods%20for%20PID%20Controllers-2.pdf. Cited 4 January 2011
26. Argelaguet, R., Pons, M., Martin Aguilar, J., Quevedo, J.: A new tuning of PID controllers based on LQR optimization. In: Proc. European Control Conference, Brussels, Belgium (1997). Available at www.cds.caltech.edu/conferences/related/ECC97/proceeds/251_500/ECC486.pdf. Cited 4 January 2011
27. Argelaguet, R., Pons, M., Quevedo, J., Aguilar, J.: A new tuning of PID controllers based on LQR optimization. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, pp. 303–308 (2000)
28. Arrieta, O.: Comparación del desempeño de los métodos de sintonización de controladores PI y PID basados en criterios integrales. Proyecto Eléctrico, Universidad de Costa Rica (2003). Available at http://www2.eie.ucr.ac.cr/~oarrieta/proyecto_%20Tesis_%20Licenciatura.pdf. Cited 6 September 2006 (in Spanish)
29. Arrieta, O.: PID control: servo/regulation performance and robustness issues. PhD thesis, Universitat Autònoma de Barcelona, September (2010)
30. Arrieta, O.: Sintonización de controladores PI y PID empleando un índice de desempeño de criterio múltiple. Dissertation, Licenciado en Ingeniería Eléctrica, Universidad de Costa Rica (2006). Available at http://www2.eie.ucr.ac.cr/~oarrieta/Tesis_Licenciatura.pdf. Cited 4 January 2011 (in Spanish)
31. Arrieta, O., Alfaro, V.M.: Sintonización de controladores PI y PID utilizando los criterios integrales IAE e ITAE. Ingénierie **13**(1–2), 31–39 (2003). Available at http://www2.eie.ucr.ac.cr/~oarrieta/oarrieta_valfaro03.pdf. Cited 4 January 2011 (in Spanish)
32. Arrieta, O., Vilanova, R.: PID autotuning settings for balanced servo/regulation operation. In: Proc. 15th Mediterranean Conference on Control and Automation, Athens, Greece (2007), paper T028-015
33. Arrieta, O., Visioli, A., Vilanova, R.: Improved PID autotuning for balanced control operation. In: Proc. IEEE Conference on Emerging Technologies and Factory Automation, Mallorca, Spain, pp. 1–8 (2009)
34. Arrieta, O., Visioli, A., Vilanova, R.: PID autotuning for weighted servo/regulator control operation. J. Process Control **20**, 472–480 (2010)
35. Arrieta, O., Vilanova, R., Visioli, A.: Proportional-Integral-Derivative tuning for servo/regulation control operation for unstable and integrating processes. Ind. Eng. Chem. Res. **50**(6), 3327–3334 (2011)
36. Arvanitis, K.G., Akritidis, C.B., Pasgianos, G.D., Sigrimis, N.A.: Controller tuning for second order dead-time fertigation mixing process. In: Proc. EurAgEng Conference on Agricultural Engineering (2000). Paper No 00-AE-011
37. Arvanitis, K.G., Sigrimis, N.A., Pasgianos, G.D., Kalogeropoulos, G.: On-line controller tuning for unstable processes with application to a biological reactor. In: Proc. IFAC Conference on Modelling and Control in Agriculture, Horticulture and Post-Harvest Processing, Waneningen, The Netherlands, pp. 191–196 (2000)
38. Arvanitis, K.G., Syrkos, G., Stellas, I.Z., Sigrimis, N.A.: Controller tuning for integrating processes with time delay. Part I: IPDT processes and the pseudo-derivative feedback control configuration. In: Proc. 11th Mediterranean Conference on Control and Automation (2003). Paper No. T7-040
39. Arvanitis, K.G., Syrkos, G., Stellas, I.Z., Sigrimis, N.A.: Controller tuning for integrating processes with time delay. Part III: The case of first order plus integral plus dead-time processes. In: Proc. 11th Mediterranean Conference on Control and Automation (2003). Paper No. T7-042
40. Åström, K.J.: Ziegler–Nichols auto-tuner. Report TFRT–3167, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden (1982)

41. Åström, K.J.: Tuning and adaptation. In: Proc. IFAC World Congress, San Francisco, USA, pp. 1–18 (1996). Plenary Volume
42. Åström, K.J., Hägglund, T.: Automatic tuning of simple regulators with specifications on phase and amplitude margins. Automatica **20**, 645–651 (1984)
43. Åström, K.J., Hägglund, T.: Automatic Tuning of PID Controllers. Instrument Society of America, North Carolina, USA (1988)
44. Åström, K.J., Hägglund, T.: PID Controllers: Theory, Design and Tuning. Instrument Society of America, North Carolina, USA (1995)
45. Åström, K.J., Hägglund, T.: The future of PID control. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 19–30 (2000)
46. Åström, K.J., Hägglund, T.: Benchmark systems for PID control. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 181–182 (2000)
47. Åström, K.J., Hägglund, T.: Revisiting the Ziegler–Nichols step response method for PID control. J. Process Control **14**, 635–650 (2004)
48. Åström, K.J., Hägglund, T.: Advanced PID control. Instrument Society of America, North Carolina, USA (2006)
49. Åström, K.J., Hägglund, T., Hang, C.C., Ho, W.K.: Automatic tuning and adaptation for PID controllers—a survey. Control Eng. Pract. **1**, 699–714 (1993)
50. Åström, K.J., Lee, T.H., Tan, K.K., Johansson, K.H.: Recent advances in relay feedback methods—a survey. In: Proc. IEEE International Conference on Syst., Man, Cybernetics, Vancouver, British Columbia, pp. 2616–2621 (1995)
51. Atherton, D.P., Boz, A.F.: Using standard forms for controller design. In: Proc. UKACC International Conference on Control, Swansea, UK, pp. 1066–1071 (1998)
52. Atherton, D.P., Majhi, S.: Tuning of optimum PIPD controllers. In: Proc. Third Portuguese Conference on Automatic Control, Coimbra, Portugal, pp. 549–554 (1998)
53. Atkinson, P.: Feedback Control Theory for Engineers. Heinemann, London (1968)
54. Auslander, D.M., Takahashi, Y., Tomizuka, M.: The next generation of single loop controllers: hardware and algorithms for the discrete/decimal process controller. Trans. ASME J. Dyn. Syst. Meas. Control **97**(3), 280–282 (1975)
55. Bai, J., Zhang, X.: A new adaptive PI controller and its application in HVAC systems. Energy Convers. Manag. **48**, 1043–1054 (2007)
56. Bain, D.M., Martin, G.D.: Simple PID tuning and PID closed-loop simulation. In: Proc. American Control Conference, pp. 338–342 (1983)
57. Barberà, E.: First order plus dead-time (FOPDT) processes: a new procedure for tuning PI and PID controllers (2006). Available at http://www.angel.qui.ub.es/abstracts/T10-004.pdf. Cited 9 May 2006
58. Bateson, N.: Introduction to Control System Technology. Prentice-Hall, New York (2002)
59. Bekker, J.E., Meckl, P.H., Hittle, D.C.: A tuning method for first-order processes with PI controllers. ASHRAE Trans. **97**(2), 19–23 (1991)
60. Belanger, P.W., Luyben, W.L.: Design of low-frequency compensators for improvement of plantwide regulatory performances. Ind. Eng. Chem. Res. **36**, 5339–5347 (1997)
61. Benjanarasuth, T., Ngamwiwit, J., Komine, N., Ochiai, Y.: CDM based two-degree of freedom PID controllers tuning formulas. Proc. Sch. Inf. Technol. Electron. Tokai Univ., Ser. E **30**, 53–58 (2005). Available at http://ci.nii.ac.jp/vol_issue/nels/AA11898836/ISS0000391486_jp.html. Cited 4 January 2011
62. Bequette, B.W.: Process Control: Modeling, Design and Simulation. Pearson Education, New Jersey (2003)
63. Bi, Q., Cai, W.-J., Lee, E.-L., Wang, Q.-G., Hang, C.-C., Zhang, Y.: Robust identification of first-order plus dead-time model from step response. Control Eng. Pract. **7**, 71–77 (1999)
64. Bi, Q., Cai, W.-J., Wang, Q.-G., Hang, C.-C., Lee, E.-L., Sun, Y., Liu, K.-D., Zhang, Y., Zou, B.: Advanced controller auto-tuning and its application in HVAC systems. Control Eng. Pract. **8**, 633–644 (2000)
65. Bialkowski, W.L.: Control of the pulp and paper making process. In: Levine, W.S. (ed.) The Control Handbook, pp. 1219–1242. CRC/IEEE Press, Boca Raton (1996)

66. Blickley, G.J.: Modern control started with Ziegler–Nichols tuning. Control Eng. **2**, 11–17 (1990)
67. Boe, E., Chang, H.-C.: Dynamics and tuning of systems with large delay. In: Proc. American Control Conference, pp. 1572–1578 (1988)
68. Bohl, A.H., McAvoy, T.J.: Linear feedback vs. time optimal control. II. The regulator problem. Ind. Eng. Chem. Process Des. Dev. **15**, 30–33 (1976)
69. Boiko, I.M.: Non-parametric tuning of PID controllers via modified second-order sliding mode algorithms. In: Proc. IFAC World Congress, Seoul, Korea, pp. 6214–6219 (2008)
70. Boiko, I.M.: Modified relay feedback test and its use for non-parametric loop tuning. In: Proc. American Control Conference, St. Louis, USA, pp. 4755–4760 (2008)
71. Boiko, I., Sun, X., Tamayo, E.: Performance analysis and tuning of variable-structure PID controllers for level process. In: Proc. 18th IEEE Conference on Control Applications, St. Petersburg, Russia, pp. 268–273 (2009)
72. Borresen, B.A., Grindal, A.: Controllability—back to basics. ASHRAE Trans. Res., 817–819 (1990)
73. Boudreau, M.A., McMillan, G.K.: New directions in bioprocess modelling and control: Appendix C—unification of controller tuning relationships (2006). Available at http://www.modelingandcontrol.com/NewDirectionsAppendixC.pdf. Cited 4 January 2011
74. Brambilla, A., Chen, S., Scali, C.: Robust tuning of conventional controllers. Hydrocarb. Process. 53–58 (1990)
75. Branica, I., Petrović, I., Perić, N.: Toolkit for PID dominant pole design. In: Proc. 9th IEEE Conference on Electronics, Circuits and Syst., vol. 3, pp. 1247–1250 (2002)
76. Bryant, G.F., Iskenderoglu, E.F., McClure, C.H.: Design of controllers for time delay systems. In: Bryant, G.F. (ed.) Automation of Tandem Mills, pp. 81–106. The Iron and Steel Institute, London (1973)
77. Buckley, P.S.: Techniques of Process Control. Wiley, New York (1964)
78. Buckley, P., Shunta, J., Luyben, W.: Design of distillation column control systems. Butterworth-Heinemann, London (1985)
79. Bucz, Š., Marič, L., Harsányi, L., Veselý, V.: A simple robust PID controller design method based on sine wave identification of the uncertain plant. J. Electr. Eng. **61**(3), 164–170 (2010)
80. Bueno, S.S., De Keyser, R.M.C., Favier, G.: Auto-tuning and adaptive tuning of PID controllers. J. A, Benelux Q. J. Autom. Control **32**(1), 28–34 (1991)
81. Bunzemeier, A.: Ein vorschlag zur regelung integral wirkender prozesse mit eingangsstorung. Autom.tech. Prax. **40**, 26–35 (1998) (in German)
82. Byeon, J., Kim, J.-S., Chun, D., Sung, S.W., Lee, J.: Third quadrant Nyquist point for the autotuning of PI controllers. In: Proc. ICROS-SICE International Joint Conference, Fukuoka, Japan, pp. 3283–3286 (2009)
83. Calcev, G., Gorez, R.: Iterative techniques for PID controller tuning. In: Proc. 34th Conference on Decision and Control, New Orleans, USA, pp. 3209–3210 (1995)
84. Callender, A.: Preliminary notes on automatic control. I.C.I. Alkali Ltd., Northwich, UK, Central File No. R.525/15/3 (1934)
85. Callender, A., Stevenson, A.B.: Automatic control of variable physical characteristics. US patent 2,175,985 (1939)
86. Callender, A., Hartree, D.R., Porter, A.: Time-lag in a control system. Philos. Trans. R. Soc. Lond. Ser. A **235**, 415–444 (1935/1936)
87. Camacho, O.E., Smith, C., Chacón, E.: Toward an implementation of sliding mode control to chemical processes. In: Proc. IEEE International Symposium on Industrial Electronics, Guimarães, Portugal, vol. 3, pp. 1101–1105 (1997)
88. Carr, D.: AN-CNTL-13: PID control and controller tuning techniques (1986). Available at http://www.eurotherm.com/training/tutorial/instrumentation/an13_2.doc. Cited 3 September 2004
89. Chandrashekar, R., Sree, R.P., Chidambaram, M.: Design of PI/PID controllers for unstable systems with time delay by synthesis method. Indian Chem. Eng. Sect. A **44**(2), 82–88 (2002)

90. Chang, D.-M., Yu, C.-C., Chien, I.-L.: Identification and control of an overshoot lead-lag plant. J. Chin. Inst. Chem. Eng. **28**, 79–89 (1997)
91. Chao, H., Luo, Y., Di, L., Chen, Y.Q.: Roll-channel fractional order controller design for a small fixed-wing unmanned aerial vehicle. Control Eng. Pract. **18**, 761–772 (2010)
92. Chao, Y.-C., Lin, H.-S., Guu, Y.-W., Chang, Y.-H.: Optimal tuning of a practical PID controller for second order processes with delay. J. Chin. Inst. Chem. Eng. **20**, 7–15 (1989)
93. Chau, P.C.: Process Control—A First Course with MATLAB. Cambridge University Press, New York (2002)
94. Chen, G.: Conventional and fuzzy PID controllers: an overview. Int. J. Intell. Control Syst. **1**, 235–246 (1996)
95. Chen, D., Seborg, D.E.: PI/PID controller design based on direct synthesis and disturbance rejection. Ind. Eng. Chem. Res. **41**, 4807–4822 (2002)
96. Chen, Y., Won, S.: Simple fuzzy PID controller tuning of integrating process with dead time. In: Proc. International Conference on Control, Automation and Syst., Seoul, Korea, pp. 618–622 (2008)
97. Chen, F., Yang, Z.: Self-tuning PM method and its formulas deduction in PID regulators. Acta Autom. Sin. **19**(6), 736–740 (1993) (in Chinese)
98. Chen, C.-L., Yang, S.-F.: PI tuning based on peak amplitude ratio. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 195–198 (2000)
99. Chen, C.-L., Huang, H.-P., Lo, H.-C.: Tuning of PID controllers for self-regulating processes. J. Chin. Inst. Chem. Eng. **28**, 313–327 (1997)
100. Chen, C.-L., Hsu, S.-H., Huang, H.-P.: Tuning PI/PD controllers based on gain/phase margins and maximum closed loop magnitude. J. Chin. Inst. Chem. Eng. **30**, 23–29 (1999)
101. Chen, C.-L., Huang, H.-P., Hsieh, C.-T.: Tuning of PI/PID controllers based on specification of maximum closed-loop amplitude ratio. J. Chem. Eng. Jpn. **32**, 783–788 (1999)
102. Chen, P., Zhang, W., Zhu, L.: Design and tuning method of PID controller for a class of inverse response processes. In: Proc. American Control Conference, Minneapolis, USA, pp. 274–279 (2006)
103. Chen, C.-C., Huang, H.-P., Liaw, H.-J.: Set-point weighted PID controller tuning for time-delayed unstable processes. Ind. Eng. Chem. Res. **47**, 6983–6990 (2008)
104. Chen, Y., Bhaskaran, T., Xue, D.: Practical tuning rule development for fractional order proportional and integral controllers. J. Comput. Nonlinear Dyn. **3**, 021403 (2008)
105. Cheng, G.S., Hung, J.C.: A least-squares based self-tuning of PID controller. In: Proc. IEEE South East Conference, Raleigh, USA, pp. 325–332 (1985)
106. Cheng, Y.-C., Yu, C.-C.: Nonlinear process control using multiple models: relay feedback approach. Ind. Eng. Chem. Res. **39**, 420–431 (2000)
107. Chesmond, C.J.: Control System Technology. Edward Arnold, London (1982)
108. Chidambaram, M.: Design of PI controllers for integrator/dead-time processes. Hung. J. Ind. Chem. **22**, 37–39 (1994)
109. Chidambaram, M.: Design formulae for PID controllers. Indian Chem. Eng. Sect. A **37**(3), 90–94 (1995)
110. Chidambaram, M.: Design of PI and PID controllers for an unstable first-order plus time delay system. Hung. J. Ind. Chem. **23**, 123–127 (1995)
111. Chidambaram, M.: Control of unstable systems: a review. J. Energy Heat Mass Transf. **19**, 49–56 (1997)
112. Chidambaram, M.: Applied Process Control. Allied Publishers PVT, Delhi (1998)
113. Chidambaram, M.: Set point weighted PI/PID controllers for stable systems. Chem. Eng. Commun. **179**, 1–13 (2000)
114. Chidambaram, M.: Set-point weighted PI/PID controllers for integrating plus dead-time processes. In: Proc. National Symposium on Intelligent Measurement and Control, Chennai, India, pp. 324–331 (2000)
115. Chidambaram, M.: Set-point weighted PI/PID controllers for unstable first-order plus time delay systems. In: Proc. International Conference on Communications, Control and Signal Processing, Bangalore, India, pp. 173–177 (2000)

116. Chidambaram, M.: Computer Control of Processes. Alpha Science International, Oxford (2002)
117. Chidambaram, M., Kalyan, V.S.: Robust control of unstable second order plus time delay systems. In: Proc. International Conference on Advances in Chemical Engineering, Chennai, India, pp. 277–280 (1996)
118. Chidambaram, M., Sree, R.P.: A simple method of tuning PID controllers for integrator/dead-time processes. Comput. Chem. Eng. **27**, 211–215 (2003)
119. Chidambaram, M., Sree, R.P., Srinivas, M.N.: Reply to the comments by Dr. A. Abbas on "A simple method of tuning PID controllers for stable and unstable FOPTD systems" [Comp. Chem. Engineering V28 (2004) 2201–2218]. Comput. Chem. Eng. **29**, 1155 (2005)
120. Chien, I.-L.: IMC-PID controller design—an extension. In: Proc. IFAC Adaptive Control of Chemical Processes Conference, Copenhagen, Denmark, pp. 147–152 (1988)
121. Chien, I.-L.: Simple PID controller tuning method for processes with inverse response plus dead time or large overshoot response plus dead time. Ind. Eng. Chem. Res. **42**, 4461–4477 (2003)
122. Chien, I.-L., Fruehauf, P.S.: Consider IMC tuning to improve controller performance. Chem. Eng. Prog. 33–41 (1990)
123. Chien, I.-L., Huang, H.-P., Yang, J.-C.: A simple multiloop tuning method for PID controllers with no proportional kick. Ind. Eng. Chem. Res. **38**, 1456–1468 (1999)
124. Chien, I.-L., Chung, Y.-C., Chen, B.-S., Chuang, C.-Y.: Simple PID controller tuning method for processes with inverse response plus dead time or large overshoot response plus dead time. Ind. Eng. Chem. Res. **42**, 4461–4477 (2003)
125. Chien, K.L., Hrones, J.A., Reswick, J.B.: On the automatic control of generalised passive systems. Trans. ASME **74**, 175–185 (1952)
126. Chiu, K.C., Corripio, A.B., Smith, C.L.: Digital controller algorithms. Part III. Tuning PI and PID controllers. Instrum. Control Syst. **December**, 41–43 (1973)
127. Chun, D., Choi, J.Y., Lee, J.: Parallel compensation with a secondary measurement. Ind. Eng. Chem. Res. **38**, 1575–1579 (1999)
128. Clarke, D.W.: PI auto-tuning during a single transient. IEE Proc., Control Theory Appl. **153**(6), 671–683 (2006)
129. Cluett, W.R., Wang, L.: New tuning rules for PID control. Pulp Pap. Can. **3**(6), 52–55 (1997)
130. Cogan, B., de Paor, A.M., Quinn, A.: PI control of first-order lag plus time-delay plants: root locus design for optimal stability. Trans. Inst. Meas. Control **31**(5), 365–379 (2009)
131. Cohen, G.H., Coon, G.A.: Theoretical considerations of retarded control. Trans. ASME **75**, 827–834 (1953)
132. Cominos, P., Munro, N.: PID controllers: recent tuning methods and design to specification. IEE Proc., Control Theory Appl. **149**(1), 46–53 (2002)
133. Connell, B.: Process Instrumentation Applications Manual. McGraw-Hill, New York (1996)
134. ControlSoft Inc.: PID loop tuning pocket guide (Version 2.2, DS405–02/05) (2005). Available at http://www.controlsoftinc.com. Cited 30 June 2005
135. Coon, G.A.: How to find controller settings from process characteristics. Control Eng. **3**, 66–76 (1956)
136. Coon, G.A.: Control charts for proportional action. ISA J. **11**, 81–82 (1964)
137. Cooper, D.J.: PID control of the heat exchanger (2006). Available at http://www.controlguru.com/wp/p78.html. Cited 4 January 2011
138. Cooper, D.J.: PID with CO filter control of the heat exchanger (2006). Available at http://www.controlguru.com/wp/p86.html. Cited 4 January 2011
139. Corripio, A.B.: Tuning of Industrial Control Systems. Instrument Society of America, North Carolina, USA (1990)
140. Corripio, A.B.: Tuning of Industrial Control Systems. Instrument Society of America, North Carolina, USA (2005)
141. Cox, C.S., Arden, W.J.B., Doonan, A.F.: CAD software facilities tuning of traditional and predictive control strategies. In: Proc. ISA Advances in Instrumentation and Control Conference, Anaheim, USA, vol. 49, Part 2, pp. 241–250 (1994)

142. Cox, C.S., Daniel, P.R., Lowdon, A.: Quicktune: a reliable automatic strategy for determining PI and PPI controller parameters using a FOLPD model. Control Eng. Pract. **5**, 1463–1472 (1997)
143. Cuesta, A., Grau, L., López, I.: CACSD tools for tuning multi-rate PID controllers in time and frequency domains. In: Proc. IEEE International Symposium on Computer Aided Control Syst. Design, Munich, Germany, pp. 3036–3041 (2006)
144. Cvejn, J.: Sub-optimal PID controller settings for FOPDT systems with long dead time. J. Process Control **19**, 1486–1495 (2009)
145. Davydov, N.I., Idzon, O.M., Simonova, O.V.: Determining the parameters of PID-controller settings using the transient response of the controlled plant. Therm. Eng. **42**, 801–807 (1995)
146. De Oliveira, R., Corrêa, R.G., Kwong, W.H.: An IMC-PID tuning procedure based on the integral squared error (ISE) criterion: a guide tour to understand its features. In: Proc. IFAC Workshop on Control Education and Technol. Transfer Issues, Curitiba, Brazil, pp. 87–91 (1995)
147. De Paor, A.M.: A fiftieth anniversary celebration of the Ziegler–Nichols PID controller. Int. J. Electr. Eng. Educ. **30**, 303–316 (1993)
148. De Paor, A.M., O'Malley, M.: Controllers of Ziegler–Nichols type for unstable processes with time delay. Int. J. Control **49**, 1273–1284 (1989)
149. Derbel, H.B.J.: Design of PID controllers for time-delay systems by the pole compensation technique. In: Proc. 6th International Multi-Conference on Syst., Signals and Devices, Djerba, Tunisia, pp. 1–6 (2009)
150. Desbiens, A.: La commande automatique des systèmes dynamiques. Masters thesis, Université Laval, Canada (2008). Available at http://w3.gel.ulaval.ca/~desbiens/GEL-21946/NotesDeCours/master.pdf (in French). Cited 14 December 2010.
151. Devanathan, R.: An analysis of minimum integrated error solution with application to self-tuning controller. J. Electr. Electron. Eng. Aust. **11**, 172–177 (1991)
152. Dutton, K., Thompson, S., Barraclough, B.: The Art of Control Engineering. Addison-Wesley Longman, Boston (1997)
153. ECOSSE Team: The ECOSSE Control Hypercourse (1996). Available at http://eweb.chemeng.ed.ac.uk/courses/control/course/map/controllers/correlations.html. Cited 4 January 2011
154. ECOSSE Team: The ECOSSE Control Hypercourse (1996). Available at http://eweb.chemeng.ed.ac.uk/courses/control/course/map/controllers/damped.html. Cited 4 January 2011
155. Edgar, T.F., Smith, C.L., Shinskey, F.G., Gassman, G.W., Schafbuch, P.J., McAvoy, T.J., Seborg, D.E.: Process Control in Perry's Chemical Engineers' Handbook, vol. 8, pp. 1–84. McGraw-Hill International, New York (1997). Editors R.H. Perry and D.W. Green
156. Ender, D.B.: Process control performance: not as good as you think. Control Eng. **40**, 180–190 (1993)
157. Entech Control Engineering Ltd.: Competency in Process Control—Industry Guidelines (1994). Version 1.0, 3/94
158. Eriksson, L.: PID controller design and tuning in networked control systems. Ph.D. thesis, Helsinki University of Technology, Finland (2008)
159. Eriksson, L.M., Johansson, M.: PID controller tuning rules for varying time-delay systems. In: Proc. American Control Conference, New York City, USA, pp. 619–625 (2007)
160. Eriksson, L.M., Johansson, M.: Simple PID tuning rules for varying time-delay systems. In: Proc. 46th IEEE Conference on Decision and Control, New Orleans, USA, pp. 1801–1807 (2007)
161. Eriksson, L., Oksanen, T., Mikkola, K.: PID controller tuning rules for integrating processes with varying time delays. J. Franklin Inst. **346**(5), 470–487 (2009)
162. Ettaleb, L., Roche, A.: On-line tuning of malfunctioning control loops. In: Proc. Control Syst. 2000, Victoria, Canada, pp. 139–144 (2000)
163. Faanes, A., Skogestad, S.: pH-neutralization: integrated process and control design. Comput. Chem. Eng. **28**, 1475–1487 (2004)

164. Farrington, G.H.: Communications on “The practical application of frequency response analysis to automatic process control”. Proc. Inst. Mech. Eng. **162**, 346–347 (1950)
165. Fertik, H.A.: Tuning controllers for noisy processes. ISA Trans. **14**, 292–304 (1975)
166. Fertik, H.A., Sharpe, R.: Optimizing the computer control of breakpoint chlorination. In: Advances in Instrumentation: Proc. ISA Conference and Exhibit, Chicago, USA, vol. 34, Part 1, pp. 373–386 (1979)
167. Fisher, D.G.: Process control: an overview and personal perspective. Can. J. Chem. Eng. **69**, 5–26 (1991)
168. Fliess, M., Join, C.: Intelligent PID controllers. In: Proc. 16th Mediterranean Conference on Control and Automation, Ajaccio, France, pp. 326–331 (2008)
169. Foley, M.W., Ramharack, N.R., Copeland, B.R.: Comparison of PI controller tuning methods. Ind. Eng. Chem. Res. **44**(17), 6741–6750 (2005)
170. Ford, R.L.: The determination of the optimum process-controller settings and their confirmation by means of an electronic simulator. IEE Proc., Part 2 **101**, 141–155 (1953)
171. Ford, R.L.: The determination of the optimum process-controller settings and their confirmation by means of an electronic simulator. IEE Proc., Part 2 **101**, 173–177 (1953)
172. Frank, P.M., Lenz, R.: Entwurf erweiterter PI-regler für totzeitstrecken mit verzögerung erster ordnung. ETZ, Elektrotech. Z., Ausg. A **90**(3), 57–63 (1969) (in German)
173. Friman, M., Waller, K.V.: A two channel relay for autotuning. Ind. Eng. Chem. Res. **36**, 2662–2671 (1997)
174. Fruehauf, P.S., Chien, I.-L., Lauritsen, M.D.: Simplified IMC-PID tuning rules. In: Proc. ISA Advances in Instrumentation and Control Conference, Chicago, USA, vol. 48, pp. 1745–1766 (1993)
175. Fukura, S., Tanura, H.: PI controller tuning of second order lag plus dead time processes subject to a reference input. Trans. Soc. Instrum. Control Eng. **19**(6), 514–515 (1983)
176. Gaikward, R., Chidambaram, M.: Design of PID controllers for unstable systems. In: Proc. National Symposium on Intelligent Measurement and Control, Chennai, India, pp. 332–339 (2000)
177. Gallier, P.W., Otto, R.E.: Self-tuning computer adapts DDC algorithms. Instrum. Technol. 65–70 (1968)
178. García, R.F., Castelo, F.J.P.: A complement to autotuning methods on PID controllers. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 101–104 (2000)
179. Geng, G., Geary, G.M.: On performance and tuning of PID controllers in HVAC systems. In: Proc. 2nd IEEE Conference on Control Applications, Vancouver, Canada pp. 819–824 (1993)
180. Gerry, J.P.: How to control processes with large dead times (1998). Available at http://www.expertune.com/artdt.html. Cited 4 January 2011
181. Gerry, J.P.: Tuning process controllers starts in manual (1999). Available at http://www.expertune.com/ArtInTechMay99.html. Cited 4 January 2011
182. Gerry, J.P.: How to control a process with long dead time (2003). Available at http://www.expertune.com/learncast.html. Cited 27 June 2005
183. Gerry, J.P., Hansen, P.D.: Choosing the right controller. Chem. Eng. **25**, 65–68 (1987)
184. Gong, X.F.: Normalised tuning method of PID controller parameters. J. Zhejiang Univ. Sci. **43**(1), 43–48 (2000) (in Chinese)
185. Gong, X., Gao, J., Zhou, C.: Extension of IMC tuning to improve controller performance. In: Proc. IEEE International Conference on Syst., Man and Cybernetics, pp. 1770–1775 (1996)
186. Gong, X.F., Gao, J., Zhou, C.: Extension of IMC tuning of PID control parameter. Control Decis. **13**(4), 337–341 (1998) (in Chinese)
187. Gonzalez, A.M.: Un planteamiento continuo de la autosintonia de controladores PI y PID. Ph.D. dissertation, Dept. de Informatica y Automatica, UNED, Madrid, Spain (1994) (in Spanish)
188. Goodwin, G.C., Graebe, S.F., Salgado, M.E.: Control System Design. Prentice Hall, New Jersey (2001)

189. Gorecki, H., Fuska, S., Grabowski, P., Korytowski, A.: Analysis and Synthesis of Time Delay Systems. Wiley, New York (1989)
190. Gorez, R.: A survey of PID auto-tuning methods. J. A, Benelux Q. J. Autom. Control **38**, 3–10 (1997)
191. Gorez, R.: New design relations for 2-DOF PID-like control systems. Automatica **39**, 901–908 (2003)
192. Gorez, R., Klàn, P.: Nonmodel-based explicit design relations for PID controllers. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 141–148 (2000)
193. GPG346 Good Practice Guide: The Carbon Trust. Improving the effectiveness of basic closed loop control systems. Available at http://www.carbontrust.co.uk/Publications/pages/publicationdetail.aspx?id=GPG346. Cited 4 January 2011
194. Gu, D., Zhang, W.: Design of an H_∞ based PI controller for AQM routers supporting TCP flows. In: Proc. Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference, Shanghai, China, pp. 603–608 (2009)
195. Gu, D., Liu, T., Zhang, W.: Study on the relationship between two typical modelling methods in process control. In: Proc. 42nd IEEE Conference on Decision and Control, Maui, USA, pp. 4082–4083 (2003)
196. Gude, J.J., Kahoraho, E.: New tuning rules for PI and fractional PI controllers. In: Proc. AdChem, Istanbul, Turkey (2009)
197. Haalman, A.: Adjusting controllers for a deadtime process. Control Eng. **July**, 71–73 (1965)
198. Haeri, M.: Tuning rules for the PID controller using a DMC strategy. Asian J. Control **4**, 410–417 (2002)
199. Haeri, M.: PI design based on DMC strategy. Trans. Inst. Meas. Control **27**, 21–36 (2005)
200. Hägglund, T., Åström, K.J.: Industrial adaptive controllers based on frequency response techniques. Automatica **27**, 599–609 (1991)
201. Hägglund, T., Åström, K.J.: Revisiting the Ziegler–Nichols tuning rules for PI control. Asian J. Control **4**, 364–380 (2002)
202. Hägglund, T., Åström, K.J.: Revisiting the Ziegler–Nichols tuning rules for PI control—Part II. The frequency response method. Asian J. Control **6**(4), 469–482 (2004)
203. Hang, C.C., Åström, K.J.: Refinements of the Ziegler–Nichols tuning formulae for PID auto-tuners. In: Proc. ISA International Conference and Exhibition. Advances in Instrumentation, vol. 43, pp. 1021–1030 (1988)
204. Hang, C.C., Åström, K.J.: Practical aspects of PID auto-tuners based on relay feedback. In: Proc. IFAC Adaptive control of Chemical Processes Conference, Copenhagen, Denmark, pp. 153–158 (1988)
205. Hang, C.C., Cao, L.: Improvement of transient response by means of variable set-point weighting. IEEE Trans. Ind. Electron. **43**, 477–484 (1996)
206. Hang, C.C., Åström, K.J., Ho, W.K.: Refinements of the Ziegler–Nichols tuning formula. IEE Proc. Part D. Control Theory Appl. **138**, 111–118 (1991)
207. Hang, C.C., Ho, W.K., Cao, L.S.: A comparison of two design methods for PID controllers. In: Proc. ISA Advances in Instrumentation and Control Conference, Chicago, USA, vol. 48, pp. 959–967 (1993)
208. Hang, C.C., Lee, T.H., Ho, W.K.: Adaptive Control. Instrument Society of America, North Carolina, USA (1993)
209. Hang, C.C., Ho, W.H., Cao, L.S.: A comparison of two design methods for PID controllers. ISA Trans. **33**, 147–151 (1994)
210. Hang, C.C., Åström, K.J., Wang, Q.G.: Relay feedback auto-tuning of process controllers—a tutorial review. J. Process Control **12**, 143–162 (2002)
211. Hansen, P.D.: Controller structure and tuning for unmeasured load disturbance. In: Proc. American Control Conference, Philadelphia, USA, vol. 1, pp. 131–136 (1998)
212. Hansen, P.D.: Robust adaptive PID controller tuning for unmeasured load rejection. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 487–494 (2000)

213. Harriott, P.: Process Control. McGraw-Hill, New York (1964)
214. Harriott, P.: Optimum controller settings for processes with dead time: effects of type and location of disturbance. Ind. Eng. Chem. Res. **27**(11), 2060–2063 (1988)
215. Harris, T.J., Tyreus, B.D.: Comments on internal model control. 4. PID controller design. Ind. Eng. Chem. Res. **26**, 2161–2162 (1987)
216. Harrold, D.: Process controller tuning guidelines. Control Eng. (1999). Available at http://www.smartsys.bg/index.php?page=Engineering&ask=PCTG. Cited 10 January 2012
217. Hartree, D.R., Porter, A., Callender, A., Stevenson, A.B.: Time-lag in a control system—II. Proc. R. Soc. Lond. A **161**, 460–476 (1937)
218. Hassan, G.A.: Computer-aided tuning of analog and digital controllers. Control Comput. **21**(1), 1–6 (1993)
219. Hay, J.: Regeltechniek 1. Die Keure n.v., Brugge (1998) (in Flemish)
220. Hazebroek, P., Van der Waerden, B.L.: The optimum tuning of regulators. Trans. ASME **72**, 317–322 (1950)
221. Heck, E.: Stetige Regler an Regelstrecken—bestehend aus Verzögerungsgliedern erster Ordnung. Heiz. Lüft. Haustech. **20**(9), 333–337 (1969) (in German)
222. Henry, J., Schaedel, H.M.: International co-operation in control engineering education using online experiments. Eur. J. Eng. Educ. **30**(2), 265–274 (2005)
223. Hersh, M.A., Johnson, M.A.: A study of advanced control systems in the workplace. Control Eng. Pract. **5**(6), 771–778 (1997)
224. Hill, A.G., Adams, C.B.: Effect of disturbance dynamics on optimum control of 3rd and 4th order processes. In: Proc. ISA International Conference and Exhibition, Houston, USA, vol. 43, Part 3, pp. 967–983 (1988)
225. Hill, A.G., Venable, S.W.: The effect of model error on optimum PID controller tuning. In: Proc. ISA International Conference and Exhibition, Philadelphia, USA, vol. 44, Part 1, pp. 51–64 (1989)
226. Hiroi, K., Terauchi, Y.: Two degrees of freedom algorithm. In: Proc. ISA International Conference and Exhibition. Advances in Instrumentation, pp. 789–796 (1986)
227. Ho, W.K., Xu, W.: PID tuning for unstable processes based on gain and phase-margin specifications. IEE Proc., Control Theory Appl. **145**, 392–396 (1998)
228. Ho, W.K., Hang, C.C., Zhou, J.H., Yip, C.K.: Adaptive PID control of a process with underdamped response. In: Proc. Asian Control Conference, Tokyo, Japan, pp. 335–338 (1994)
229. Ho, W.K., Hang, C.C., Cao, L.S.: Tuning of PID controllers based on gain and phase margin specifications. Automatica **31**, 497–502 (1995)
230. Ho, W.K., Hang, C.C., Zhou, J.: Self-tuning PID control of a plant with under-damped response with specifications on gain and phase margins. IEEE Trans. Control Syst. Technol. **5**, 446–452 (1997)
231. Ho, W.K., Lim, K.W., Xu, W.: Optimal gain and phase margin tuning for PID controllers. Automatica **34**, 1009–1014 (1998)
232. Ho, W.K., Lim, K.W., Hang, C.C., Ni, L.Y.: Getting more phase margin and performance out of PID controllers. Automatica **35**, 1579–1585 (1999)
233. Ho, W.K., Lee, T.H., Han, H.P., Hong, Y.: Self-tuning IMC-PID control with interval gain and phase margins assignment. IEEE Trans. Control Syst. Technol. **9**, 535–541 (2001)
234. Horn, I.G., Arulandu, J.R., Gombas, C.J., Van Antwerp, J.G., Braatz, R.D.: Improved filter design in internal model control. Ind. Eng. Chem. Res. **35**, 3437–3441 (1996)
235. Hougen, J.O.: Measurement and Control Applications. Instrument Society of America, North Carolina, USA (1979)
236. Hougen, J.O.: A software program for process controller parameter selection. In: Proc. ISA International Conference and Exhibition. Advances in Instrumentation. vol. 43, Part 1, pp. 441–456 (1988)
237. Huang, C.-T., Lin, Y.-S.: Tuning PID controller for open-loop unstable processes with time delay. Chem. Eng. Commun. **133**, 11–30 (1995)
238. Huang, C.-T., Chou, C.-J., Wang, J.-L.: Tuning of PID controllers based on the second order model by calculation. J. Chin. Inst. Chem. Eng. **27**, 107–120 (1996)

239. Huang, H.-P., Chao, Y.-C.: Optimal tuning of a practical digital PID controller. Chem. Eng. Commun. **18**, 51–61 (1982)
240. Huang, H.-P., Chen, C.-C.: Control-system synthesis for open-loop unstable process with time delay. IEE Proc., Control Theory Appl. **144**, 334–346 (1997)
241. Huang, H.-P., Chen, C.-C.: Auto-tuning of PID controllers for second-order unstable process having dead time. J. Chem. Eng. Jpn. **32**, 486–497 (1999)
242. Huang, H.-P., Jeng, J.-C.: Monitoring and assessment of control performance for single loop systems. Ind. Eng. Chem. Res. **41**, 1297–1309 (2002)
243. Huang, H.-P., Jeng, J.-C.: Identification for monitoring and autotuning of PID controllers. J. Chem. Eng. Jpn. **36**, 284–296 (2003)
244. Huang, H.-P., Jeng, J.-C.: Process reaction curve and relay methods—identification and PID tuning. In: Johnson, M.A., Moradi, M.H. (eds.) PID Control: New Identification and Design Methods, pp. 297–338. Springer, London (2005)
245. Huang, H.-P., Chien, I.-L., Lee, Y.-C., Wang, G.-B.: A simple method for tuning cascade control systems. Chem. Eng. Commun. **165**, 89–121 (1998)
246. Huang, H.-P., Lee, M.-W., Chen, C.-L.: Inverse-based design for a modified PID controller. J. Chin. Inst. Chem. Eng. **31**, 225–236 (2000)
247. Huang, H.-P., Roan, M.-L., Jeng, J.-C.: On-line adaptive tuning for PID controllers. IEE Proc., Control Theory Appl. **149**, 60–67 (2002)
248. Huang, H.P., Luo, K.-Y., Jeng, J.-C.: Model based auto-tuning system using relay feedback. In: Proc. IFAC Advanced Control of Chemical Processes Conference, Hong Kong, China, pp. 625–630 (2003)
249. Huang, H.-P., Jeng, J.-C., Luo, K.-Y.: Auto-tune system using single-run relay feedback test and model-based controller design. J. Process Control **15**, 713–727 (2005)
250. Huang, H.-P., Lin, F.-Y., Jeng, J.-C.: Multi-loop PID controllers design for MIMO processes containing integrator(s). J. Chem. Eng. Jpn. **38**(9), 742–756 (2005)
251. Huba, M.: P- und PD-Polvorgaberegler für Regelstrecken mit begrenzter Stellgröße (P and PD pole assignment controllers for constrained systems). Automatisierungstechnik **53**(6), 273–284 (2005) (in German)
252. Huba, M.: Robust design of integrating controllers for IPDT plant. In: Proc. 17th International Conference on Process Control, Strbske Pleso, Slovak Republic, pp. 353–357 (2009)
253. Huba, M., Ťapák, P.: Relay identification by analyzing nonsymmetrical oscillations for single integrator with time delay. J. Cybern. Inf. **10**, 68–77 (2010)
254. Huba, M., Žáková, K.: Contribution to the theoretical analysis of the Ziegler–Nichols method. J. Electr. Eng. **54**(7–8), 188–194 (2003)
255. Hwang, S.-H.: Closed-loop automatic tuning of single-input-single-output systems. Ind. Eng. Chem. Res. **34**, 2406–2417 (1995)
256. Hwang, S.-H., Chang, H.-C.: A theoretical examination of closed-loop properties and tuning methods of single-loop PI controllers. Chem. Eng. Sci. **42**, 2395–2415 (1987)
257. Hwang, S.-H., Fang, S.-M.: Closed-loop tuning method based on dominant pole placement. Chem. Eng. Commun. **136**, 45–66 (1995)
258. Hwang, S.-H., Tseng, T.-S.: Process identification and control based on dominant pole expansions. Chem. Eng. Sci. **49**(12), 1973–1983 (1994)
259. Hypiusová, M., Veselý, V.: Tuning of PID controller for guaranteed performance. In: Proc. International Carpathian Control Conference, Sinaia, Romania, pp. 243–246 (2008)
260. Idzerda, H.H., Ensing, L., Janssen, J.M.L., Offereins, R.P.: Design and applications of an electronic simulator for control systems. Trans. Soc. Instrum. Technol. **September**, 105–122 (1955)
261. Isaksson, A.J., Graebe, S.F.: Analytical PID parameter expressions for higher order systems. Automatica **35**, 1121–1130 (1999)
262. ISMC: RAPID: Robust Advanced PID Control Manual. Intelligent System Modeling and Control nv, Belgium (1999)
263. Jacob, E.F., Chidambaram, M.: Design of controllers for unstable first-order plus time delay systems. Comput. Chem. Eng. **20**, 579–584 (1996)

264. Jahanmiri, A., Fallahi, H.R.: New methods for process identification and design of feedback controller. Trans. Inst. Chem. Eng. **75**(A), 519–522 (1997)
265. Jhunjhunwala, M., Chidambaram, M.: PID controller tuning for unstable systems by optimization method. Chem. Eng. Commun. **185**, 91–113 (2001)
266. Jin, Q., Quan, L., Wang, X., Qi, F.: Base on all-pole approximation a new internal model PID control method for the system with time delays. In: Proc. IEEE International Conference on Mechatronics and Automation, Changchun, China, pp. 268–273 (2009)
267. Johnson, E.F.: Use of frequency response analysis in chemical engineering process control. Chem. Eng. Prog. **52**(2), 64-F–68-F (1956)
268. Jones, R.W., Tham, M.T.: Control strategies for process intensified systems. In: Proc. SICE-ICASE International Joint Conference, Busan, Korea, pp. 4618–4623 (2006)
269. Jones, K.O., Williams, D., Montgomery, P.A.: On-line application of an auto-tuning PID controller for dissolved oxygen concentration in a fermentation process. Trans. Inst. Meas. Control **19**, 253–262 (1997)
270. Juang, W.-S., Wang, F.-S.: Design of PID controller by concept of Dahlin's Law. J. Chin. Inst. Chem. Eng. **26**, 133–136 (1995)
271. Jyothi, S.N., Arvind, S., Chidambaram, M.: Design on PI/PID controller for systems with a zero. Indian Chem. Eng. Sect. A **43**(4), 288–293 (2001)
272. Kamimura, K., Yamada, A., Matsuba, T., Kimbara, A., Kurosu, S., Kasahara, M.: CAT (Computer-aided tuning) software for PID controllers. ASHRAE Trans. **100**(1), 180–190 (1994)
273. Kang, T.-W.: Effect of disturbance dynamics optimum control of second-order plus dead time process. Ph.D. thesis, Oklahoma State University, USA (1989)
274. Karaboga, D., Kalinli, A.: Tuning PID controller parameters using Tabu search algorithm. In: Proc. IEEE International Conference on Syst., Man and Cybernetics, pp. 134–136 (1996)
275. Kasahara, M., Matsuba, T., Murasawa, I., Hashimoto, Y., Kamimura, K., Kimbara, A., Kurosu, S.: A tuning method of two degrees of freedom PID controller. ASHRAE Trans. **103**, 278–289 (1997)
276. Kasahara, M., Matsuba, T., Kuzuu, Y., Yamazaki, T., Hashimoto, Y., Kamimura, K., Kurosu, S.: Design and tuning of robust PID controller for HVAC systems. ASHRAE Trans. **105**(2), 154–166 (1999)
277. Kasahara, M., Yamazaki, T., Kuzuu, Y., Hashimoto, Y., Kamimura, K., Matsuba, T., Kurosu, S.: Stability analysis and tuning of PID controller in VAV systems. ASHRAE Trans. **107**, 285–296 (2001)
278. Kaya, I.: A PI-PD controller design for control of unstable and integrating processes. ISA Trans. **42**, 111–121 (2003)
279. Kaya, I., Atherton, D.P.: A PI-PD controller design for integrating processes. In: Proc. American Control Conference, San Diego, USA, pp. 258–262 (1999)
280. Kaya, A., Scheib, T.J.: Tuning of PID controls of different structures. Control Eng. **July**, 62–65 (1988)
281. Keane, M.A., Yu, J., Koza, J.R.: Automatic synthesis of both the topology and tuning of a common parameterized controller for two families of plants using genetic programming. In: Proc. Genetic and Evolutionary Computation Conference, Las Vegas, USA, pp. 496–504 (2000)
282. Keane, M.A., Koza, J.R., Streeter, M.J.: Apparatus for improved general-purpose PID and non-PID controllers. US Patent No. 6,847,851 B1 (2005)
283. Keviczky, L., Csáki, F.: Design of control systems with dead time in the time domain. Acta Tech. Acad. Sci. Hung. **74**(1–2), 63–84 (1973)
284. Khan, B.Z., Lehman, B.: Setpoint PI controllers for systems with large normalised dead time. IEEE Trans. Control Syst. Technol. **4**, 459–466 (1996)
285. Khodabakhshian, A., Golbon, N.: Unified PID design for load frequency control. In: Proc. IEEE International Conference on Control Applications, Taipei, Taiwan, pp. 1627–1632 (2004)

286. Kim, I.-H., Fok, S., Fregene, K., Lee, D.-H., Oh, T.-S., Wang, D.W.L.: Neural-network bases system identification and controller synthesis for an industrial sewing machine. Int. J. Control. Autom. Syst. **2**(1), 83–91 (2004)
287. King, R.: Private communication, 6 March 2006
288. Kinney, T.B.: Tuning process controllers. Chem. Eng. **19**, 67–72 (1983)
289. Klán, P.: Moderní metody nastavení PID regulátorů [Modern methods of setting PID controllers]. Část I: Procesy s přechodovou charakteristikou typu "S". Automa **9**, 54–57 (2000) (in Czech)
290. Klán, P.: Moderní metody nastavení PID regulátorů. Část II: Integrační procesy. Automa **1**, 52–54 (2001) (in Czech)
291. Klán, P., Gorez, R.: Vyvážené nastavení PI regulátorů. Automa **4**, 49–53 (2000) (in Czech)
292. Klán, P., Gorez, R.: Nastavení PI regulátorů chránící akční členy [I control with actuator preservation]. Automa **2**, 50–52 (2005) (in Czech)
293. Klán, P., Gorez, R.: On aggressiveness of PI control. In: Proc. IFAC World Congress, Prague, Czech Republic (2005)
294. Klán, P., Gorez, R.: PI controller design for actuator preservation. In: Proc. IFAC World Congress, Seoul, Korea, pp. 5820–5824 (2008)
295. Klein, M., Walter, H., Pandit, M.: Digitaler PI-regler: Neue Einstellregeln mit Hilfe der Streckensprungantwort [Digital PI-control: new tuning rules based on step response identification]. at-Automatisierungstechnik **40**(8), 291–299 (1992) (in German)
296. Koivo, H.N., Tanttu, J.T.: Tuning of PID controllers: survey of SISO and MIMO techniques. In: Proc. IFAC Intelligent Tuning Adaptive Control Symposium, Singapore, pp. 75–80 (1991)
297. Kookos, I.K., Lygeros, A.I., Arvanitis, K.G.: On-line PI controller tuning for integrator/dead time processes. Eur. J. Control **5**, 19–31 (1999)
298. Kosinsani, S.: Effect of disturbance dynamic characteristics on optimum PID controller tuning constants. Ph.D. thesis, Oklahoma State University, USA (1985)
299. Kotaki, M., Yamakawa, Y., Yamazaki, T., Kamimura, K., Kurosu, S.: A tuning method for PID controller that considers changes in system characteristics. ASHRAE Trans. **111**(2), 13–22 (2005)
300. Kotaki, M., Yamkazi, T., Matuba, T., Kamimura, K., Kurosu, S.: A tuning method for PID controller under consideration of changes in plant characteristics. Trans. Soc. Instrum. Control Eng. **41**(2), 177–179 (2005)
301. Kraus, T.W.: Pattern-recognizing self-tuning controller. US Patent Number 4,602,326 (1986)
302. Kristiansson, B.: PID controllers design and evaluation. PhD thesis, Chalmers University of Technology, Sweden (2003)
303. Kristiansson, B., Lennartson, B.: Robust design of PID controllers including auto-tuning rules. In: Proc. American Control Conference, vol. 5, pp. 3131–3132 (1998)
304. Kristiansson, B., Lennartson, B.: Optimal PID controllers for unstable and resonant plants. In: Proc. Conference on Decision and Control, Tampa, USA, pp. 4380–4381 (1998)
305. Kristiansson, B., Lennartson, B.: Near optimal tuning rules for PI and PID controllers. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 369–374 (2000)
306. Kristiansson, B., Lennartson, B.: Robust and optimal tuning of PI and PID controllers. IEE Proc., Control Theory Appl. **149**, 17–25 (2002)
307. Kristiansson, B., Lennartson, B.: Convenient almost optimal and robust tuning of PI and PID controllers. In: Proc. IFAC World Congress, Barcelona, Spain (2002)
308. Kristiansson, B., Lennartson, B.: Robust tuning of PI and PID controllers. IEEE Control Syst. Mag. **26**(1), 55–69 (2006)
309. Kristiansson, B., Lennartson, B.: Evaluation and tuning of robust PID controllers. In: Proc. Nordic Process Control Workshop, Trondheim, Norway (2003). Available at http://www.itk.ntnu.no/groups/npcw11/PapersDownload/Kristiansson&Lennartson.pdf. Cited 4 January 2011

310. Kristiansson, B., Lennartson, B., Fransson, C.-M.: From PI to H_∞ control in a unified framework. In: Proc. 39th IEEE Conference on Decision and Control, Sydney, Australia, pp. 2740–2745 (2000)

311. Kuhn, U.: Ein praxisnahe Einstellregel für PID-regler: die T-summen-regel. Autom.tech. Prax. **37**(5), 10–16 (1995) (in German)

312. Kukal, J.: O volbč parametrů PI a PID regulátorů. Automatizace **49**(1), 16–20 (2006) (in Czech)

313. Kuwata, R.: An improved ultimate sensitivity method and PID; I-PD control characteristics. Trans. Soc. Instrum. Control Eng. **23**(3), 232–239 (1987)

314. Kwak, H.J., Sung, S.W., Lee, I.-B.: Stabilizability conditions and controller design for unstable processes. Trans. Inst. Chem. Eng. **78**(A), 549–556 (2000)

315. Landau, I.D., Voda, A.: An analytical method for the auto-calibration of PID controllers. In: Proc. 31st Conference on Decision and Control, Tucson, USA, pp. 3237–3242 (1992)

316. Larionescu, S.: Reglarea automată a instalaţiilor pe baza unui model intern (2002). Available at http://www.geocities.com/larionescu/Infos1.htm. Cited 27 May 2008 (in Romanian)

317. Latzel, W.: Einstellregeln für kontinuierliche und Abtast-Regler nach der Methode der Betragsanpassung (On the setting of continuous and sampled data controllers by the method of gain adjustment). Automatisierungstechnik **36**, 170–178 (1988) (in German)

318. Lavanya, K., Umamaheswari, B., Panda, R.C.: System identification and controller tuning rule for DC-DC converter using ripple voltage waveform. In: Proc. International Conference on Power Electronics, Drives and Energy Syst., New Delhi, India, pp. 1–4 (2006)

319. Lavanya, K., Umamaheswari, B., Panda, R.C.: Identification of second order plus dead time systems using relay feedback test. Indian Chem. Eng. Sect. A **48**(3), 94–102 (2006)

320. Lee, C.-H.: A survey of PID controller design based on gain and phase margins. Int. J. Comput. Cogn. **2**(3), 63–100 (2004)

321. Lee, J., Edgar, T.F.: Improved PI controller with delayed or filtered integral mode. AIChE J. **48**, 2844–2850 (2002)

322. Lee, W.S., Shi, J.: Modified IMC-PID controllers and generalised PID controllers for first-order plus dead-time processes. In: Proc. Seventh International Conference on Control, Automation, Robotics and Vision, Singapore, pp. 898–903 (2002)

323. Lee, C.-H., Teng, C.-C.: Calculation of PID controller parameters by using a fuzzy neural network. ISA Trans. **42**, 391–400 (2003)

324. Lee, J., Choi, J.Y., Lee, S.D., Kwon, Y.S.: Improved method for a PID controller tuning by the dominant pole placement. J. Korean Inst. Chem. Eng. **30**(5), 631–634 (1992) (in Korean)

325. Lee, Y., Park, S., Lee, M., Brosilow, C.: PID controller tuning to obtain desired closed-loop responses for SI/SO systems. AIChE J. **44**, 106–115 (1998)

326. Lee, Y., Lee, J., Park, S.: PID tuning for integrating and unstable processes with time delay. Chem. Eng. Sci. **55**, 3481–3493 (2000)

327. Lee, J., Park, H.C., Sung, S.W.: Analytical expression of ultimate gains and ultimate periods with phase-optimal approximations of time delays. Can. J. Chem. Eng. **83**, 990–995 (2005)

328. Lee, Y., Park, S., Lee, M.: Consider the generalized IMC-PID method for PID controller tuning of time-delay processes. Hydrocarb. Process. **January**, 87–91 (2006)

329. Lelic, M., Gajic, Z.: A reference guide to PID controllers in the nineties. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 73–82 (2000)

330. Lennartson, B., Kristiansson, B.: Pass band and high frequency robustness for PID control. In: Proc. 36th IEEE Conference on Decision and Control, San Diego, USA, pp. 2666–2671 (1997)

331. Leonard, F.: Optimum PIDS controllers, an alternative for unstable delayed systems. In: Proc. IEEE Conference on Control Applications, Glasgow, UK, pp. 1207–1210 (1994)

332. Leva, A.: PID autotuning algorithm based on relay feedback. IEE Proc. Part D. Control Theory Appl. **140**, 328–338 (1993)

333. Leva, A.: Model-based tuning: the very basics and some useful techniques. J. A: Benelux Q. J. Autom. Control **42**(3), 14–22 (2001)

334. Leva, A.: Model-based proportional-integral-derivative autotuning improved with relay feedback identification. IEE Proc., Control Theory Appl. **152**, 247–256 (2005)

335. Leva, A.: Model-based PI(D) tuning with ad-hoc regulator expressions. In: Proc. American Control Conference, New York, USA, pp. 5802–5807 (2007)
336. Leva, A.: Autotuning of PI+p controllers. In: Proc. IEEE International Symposium on Computer-Aided Control Syst. Design, San Antonio, USA, pp. 1049–1054 (2008)
337. Leva, A., Colombo, A.M.: Estimating model mismatch overbounds for the robust autotuning of industrial regulators. Automatica **26**, 1855–1861 (2000)
338. Leva, A., Colombo, A.M.: Implementation of a robust PID autotuner in a control design environment. Trans. Inst. Meas. Control **23**, 1–20 (2001)
339. Leva, A., Colombo, A.M.: On the IMC-based synthesis of the feedback block of ISA-PID regulators. Trans. Inst. Meas. Control **26**, 417–440 (2004)
340. Leva, A., Maggio, M.: The PI+p controller structure and its tuning. J. Process Control **19**, 1451–1457 (2009)
341. Leva, A., Maffezzoni, C., Scattolini, R.: Self-tuning PI-PID regulators for stable systems with varying delay. Automatica **30**, 1171–1183 (1994)
342. Leva, A., Bascetta, L., Schiavo, F.: Model-based proportional-integral/proportional-integral-derivative (PI/PID) autotuning with fast relay identification. Ind. Eng. Chem. Res. **45**(12), 4052–4062 (2006)
343. Leva, A., Cox, C., Ruano, A.: Hands-on PID autotuning: a guide to better utilisation (2006). Available at http://www.ifac-control.org/publications/list-of-professional-briefs/pb_final_levacoxruano.pdf/view. Cited 4 January 2011
344. Li, Z., Su, X., Lin, P.: A practical algorithm for PID auto-tuning. Adv. Model. Anal. C, Syst. Anal. Control Des. **40**(2), 17–27 (1994)
345. Li, Y.-N., Zhang, W.-D., Liu, T.: New design method of PID controller for inverse response processes with time delay. J. Shanghai Chiao-Tung Univ. **38**(4), 506–509 (2004) (in Chinese)
346. Li, Y., Ang, K.H., Chong, G.C.Y.: PID control system analysis and design. IEEE Control Syst. Mag. **26**(1), 32–41 (2006)
347. Li, Y., Ang, K.H., Chong, G.C.Y.: Patents, software and hardware for PID control: an overview and analysis of the current art. IEEE Control Syst. Mag. **26**(1), 42–54 (2006)
348. Lim, C.M., Tan, W.C., Lee, T.S.: Tuning of PID controllers for first and second-order lag processes with dead time. Int. J. Electr. Eng. Educ. **22**, 345–353 (1985)
349. Lipták, B. (ed.): Instrument Engineers Handbook Volume II: Process Control. Chilton, Philadelphia (1970)
350. Lipták, B.: Controller tuning II: Problems and methods (2001). Available at http://www.controlmagazine.com. Cited 21 November 2003
351. Lipták, B.: Post-il Energy Technology: The World's First Solar-Hydrogen Demonstration Power Plant, p. 193. CRC Press, Boca Raton (2009)
352. Litrico, X., Fromion, V.: Tuning of robust distant downstream PI controllers for an irrigation canal pool. I: Theory. J. Irrig. Drain. Eng. **132**(4), 359–368 (2006)
353. Litrico, X., Fromion, V., Baume, J.-P.: Tuning of robust distant downstream PI controllers for an irrigation canal pool. II: Implementation issues. J. Irrig. Drain. Eng. **132**(4), 369–379 (2006)
354. Litrico, X., Malaterre, P.-O., Baume, J.-P., Vion, P.-Y., Ribot-Bruno, J.: Automatic tuning of PI controllers for an irrigation canal pool. J. Irrig. Drain. Eng. **133**(1), 27–37 (2007)
355. Liu, T., Gu, D., Zhang, W.: A H infinity design method of PID controller for second-order processes with integrator and time delay. In: Proc. 42nd IEEE Conference on Decision and Control, Maui, USA, pp. 6044–6049 (2003)
356. Lloyd, S.G.: Tuning arrangements for turning the control parameters of a controller. U.S. Patent Number 5,283,729 (1994)
357. Lopez, A.M.: Optimisation of system response. Ph.D. dissertation, Louisiana State University, USA (1968)
358. Lopez, A.M., Smith, C.L., Murrill, P.W.: An advanced tuning method. Br. Chem. Eng. **14**, 1553–1555 (1969)
359. Loron, L.: Tuning of PID controllers by the non-symmetrical optimum method. Automatica **33**, 103–107 (1997)

360. Luo, K.-N., Kuo, C.-Y., Sheu, L.-T.: A novel method for fuzzy self-tuning PID controllers. In: Proc. Asian Fuzzy Syst. Symposium, pp. 194–199 (1996)
361. Luyben, W.L.: Dynamics and control of recycle systems. 1. Simple open loop and closed loop systems. Ind. Eng. Chem. Res. **32**, 466–475 (1993)
362. Luyben, W.L.: Tuning proportional-integral-derivative controllers for integrator/deadtime processes. Ind. Eng. Chem. Res. **35**, 3480–3483 (1996)
363. Luyben, W.L.: Tuning temperature controllers on openloop unstable reactors. Ind. Eng. Chem. Res. **37**, 4322–4331 (1998)
364. Luyben, W.L.: Tuning proportional-integral controllers for processes with both inverse response and deadtime. Ind. Eng. Chem. Res. **39**, 973–976 (2000)
365. Luyben, W.L.: Effect of derivative algorithm and tuning selection on the PID control of deadtime processes. Ind. Eng. Chem. Res. **40**, 3605–3611 (2001)
366. Luyben, W.L., Luyben, M.L.: Essentials of Process Control. McGraw-Hill, Singapore (1997)
367. MacLellan, G.D.S.: Communications on "The practical application of frequency response analysis to automatic process control". In: Proc. Institution of Mechanical Engineers (London), vol. 162, pp. 347–348 (1950)
368. Madhuranthakam, C.R., Elkamel, A., Budman, H.: Optimal tuning of PID controllers for FOPDT, SOPDT and SOPDT with lead processes. Chem. Eng. Process. **47**, 251–264 (2008)
369. Maffezzoni, C., Rocco, P.: Robust tuning of PID regulators based on step-response identification. Eur. J. Control **3**, 125–136 (1997)
370. Majhi, S.: Relay feedback process identification and controller design. Ph.D. thesis, University of Sussex, UK (1999)
371. Majhi, S.: On-line PI control of stable processes. J. Process Control **15**, 859–867 (2005)
372. Majhi, S., Atherton, D.P.: Autotuning and controller design for processes with small time delays. IEE Proc., Control Theory Appl. **146**(5), 415–425 (1999)
373. Majhi, S., Atherton, D.P.: Online tuning of controllers for an unstable FOPDT process. IEE Proc., Control Theory Appl. **147**, 421–427 (2000)
374. Majhi, S., Litz, L.: On-line tuning of PID controllers. In: Proc. American Control Conference, Denver, USA, pp. 5003–5004 (2003)
375. Majhi, S., Mahanta, C.: Tuning of controllers for integrating time delay processes. In: Proc. IEEE Region 10 Int. Conference on Electrical and Electronic Technol., vol. 1, pp. 317–320 (2001)
376. Malwatkar, G.M., Sonawane, S.H., Waghmare, L.M.: Tuning PID controllers for higher-order oscillatory systems with improved performance. ISA Trans. **48**, 347–353 (2009)
377. Mann, G.K.I., Hu, B.-G., Gosine, R.G.: Time-domain based design and analysis of new PID tuning rules. IEE Proc., Control Theory Appl. **148**, 251–261 (2001)
378. Mantz, R.J., Tacconi, E.J.: Complementary rules to Ziegler and Nichols' rules for a regulating and tracking controller. Int. J. Control **49**, 1465–1471 (1989)
379. Manum, H.: Extensions of Skogestad's SIMC tuning rules to oscillatory and unstable processes (2005). Available at http://www.nt.ntnu.no/users/skoge/diplom/prosjekt05/manum/rapport.pdf. Cited 4 January 2011
380. Marchetti, G., Scali, C.: Use of modified relay techniques for the design of model-based controllers for chemical processes. Ind. Eng. Chem. Res. **39**, 3325–3334 (2000)
381. Marchetti, G., Scali, C., Lewin, D.R.: Identification and control of open-loop unstable processes by relay methods. Automatica **37**, 2049–2055 (2001)
382. Marlin, T.E.: Process Control. McGraw-Hill, New York (1995)
383. Maroto, R.: Ecuaciones para la sintonización de controladores PID con acción derivative aplicada a la señal realimentada. Proyecto Eléctrico, Universidad de Costa Rica (2007). Available at http://eie.ucr.ac.cr/uploads/file/proybach/pb0714t.pdf. Cited 4 January 2011 (in Spanish)
384. Mataušek, M.R., Kvaščev, G.S.: A unified step response procedure for autotuning of PI controller and Smith predictor for stable processes. J. Process Control **13**, 787–800 (2003)
385. Mataušek, M.R., Ribić, A.I.: Design and robust tuning of control scheme based on the PD controller plus Disturbance Observer and low-order integrating first-order plus dead-time model. ISA Trans. **48**(4), 410–416 (2009)

386. Matsuba, T., Kasahara, M., Murasawa, I., Hashimoto, Y., Kamimura, K., Kimbara, A., Kurosu, S.: Stability limit of room air temperature of a VAV system. ASHRAE Trans. **104**(2), 257–265 (1998)
387. Mazzini, H.M., Taroco, C.G., Ribeiro, L.: Design of two-degree-of-freedom control scheme for unstable SOPTD systems. In: Proc. Conferência Internacional de Aplicações Industriais, Poços de Caldas, Brazil, pp. 1–6 (2008)
388. McAnany, D.E.: A pole placement technique for optimum PID control parameters. In: Proc. ISA Advances in Instrumentation and Control Conference, Chicago, USA, vol. 48, pp. 1775–1782 (1993)
389. McAvoy, T.J., Johnson, E.F.: Quality of control problem for dead-time plants. Ind. Eng. Chem. Process Des. Dev. **6**, 440–446 (1967)
390. McMillan, G.K.: Control loop performance. In: Proc. ISA International Conference and Exhibition, Houston, USA, vol. 39, pp. 589–603 (1984)
391. McMillan, G.K.: Tuning and Control Loop Performance—A Practitioner's Guide, 3rd edn. Instrument Society of America, North Carolina, USA (1994)
392. McMillan, G.K.: Good Tuning: A Pocket Guide, 2nd edn. Instrument Society of America, North Carolina, USA (2005)
393. McMillan, G.: Tips-N-Techniques (TNT)—Tuning furnace and incinerator pressure loops (2008). Available at http://www.modelingandcontrol.com/2008/08/tipsntechniques_tnt_tuning_fur_1.html. Cited 4 January 2011
394. Méndez, V.: Ecuaciones para la sintonización de controladores PID utilizando funciones de costo del tipo IT^mE^n. Proyecto Eléctrico, Universidad de Costa Rica (2006). Available at http://www.eie.ucr.ac.cr/uploads/file/proybach/pb0631t.pdf. Cited 4 January 2011 (in Spanish)
395. Mesa, F., Lozano, J.L., Marin, L.: On the consideration of FOPDT and SOPDT responses as bounds of PI tuning. In: Proc. IEEE Mediterranean Electrotechnical Conference, Málaga, Spain, pp. 421–424 (2006)
396. Mitchell, R.J.: 'Flat phase' PID controllers. In: Proc. UK Automatic Control Conference, Manchester, UK (2008)
397. Mizutani, M., Hiroi, K.: New two degrees of freedom PID algorithm with 1st lag mean (super 2dof PID algorithm). In: Proc. ISA Advances in Instrumentation and Control, Anaheim, USA, vol. 46, Part 2, pp. 1125–1132 (1991)
398. Mnif, F.: New tuning rules of PI-like controllers with transient performances for monotonic time-delay systems. ISA Trans. **47**, 401–406 (2008)
399. Mollenkamp, R.A., Smith, C.L., Corripio, A.B.: Using models to tune industrial controllers. Instrum. Control Syst. **September**, 46–47 (1973)
400. Montenegro, A.: Sintonización de controladores PID de dos grados de libertad para lograr un desempeño optimo balanceado. Proyecto Eléctrico, Universidad de Costa Rica, Facultad de Ingeniería (2007). Available at http://www.eie.ucr.ac.cr/uploads/file/proybach/pb07_II/pb0725t.pdf. Cited 4 January 2011 (in Spanish)
401. Morari, M., Zafiriou, E.: Robust Process Control. Prentice-Hall, Englewood Cliffs (1989)
402. Morilla, F., González, A., Duro, N.: Auto-tuning PID controllers in terms of relative damping. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 161–166 (2000)
403. Moros, R.: Strecke mit Ausgleich höherer Ordnung (1999). Available at http://techni.tachemie.uni-leipzig.de/reg/regeintn.html. Cited 4 January 2011 (in German)
404. Murata, H., Sagara, S.: The noninteger order lag element plus dead time approximation of the process dynamics and its application to the optimal setting of the PI controller. Syst. Control **21**(9), 517–524 (1977) (in Japanese)
405. Murrill, P.W.: Automatic Control of Processes. International Textbook Co., Pennsylvania, USA (1967)
406. Naşcu, I., De Keyser, R., Folea, S., Buzdugan, T.: Development and evaluation of a PID autotuning controller. In: Proc. IEEE-TTTC International Conference on Automation, Quality, Testing and Robotics, Cluj-Napora, Romania, pp. 122–127 (2006)

407. Nemati, H., Bagheri, P.: A new approach to tune the two-degree-of-freedom (2DOF). In: Proc. IEEE International Symposium on Computer-Aided Control System Design, Yokohama, Japan, pp. 1819–1824 (2010)
408. NI Labview: PID control toolkit user manual (2001). Available at http://www.ni.com/pdf/manuals/322192a.pdf. Cited 4 January 2011
409. Nomura, M., Saito, T., Kitamori, T.: A simple tuning method based on partial model matching for PID controller. Trans. Inst. Electr. Eng. Jpn. **113-C**(1), 59–68 (1993) (in Japanese)
410. Normey-Rico, J.E., Alcalá, I., Gómez-Ortega, J., Camacho, E.F.: Robust PID tuning application to a mobile robot path tracking problem. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 648–653 (2000)
411. Normey-Rico, J.E., Alcalá, I., Gómez-Ortega, J., Camacho, E.F.: Mobile robot path tracking using a robust PID controller. Control Eng. Pract. **9**, 1209–1214 (2001)
412. Normey-Rico, J.E., Camacho, E.F.: Control of Dead-Time Processes, pp. 93–95. Springer, London (2007)
413. O'Connor, G.E., Denn, M.M.: Three mode control as an optimal control. Chem. Eng. Sci. **27**, 121–127 (1972)
414. O'Dwyer, A.: PID compensation of time delayed processes: a survey. In: Proc. Irish Signals and Syst. Conference, Dublin, Ireland, pp. 5–12 (2000)
415. O'Dwyer, A.: PI and PID controller tuning rule design for processes with delay, to achieve constant gain and phase margins for all values of delay. In: Proc. Irish Signals and Syst. Conference, N.U.I., Maynooth, Ireland, pp. 96–100 (2001)
416. O'Dwyer, A.: Series PID controller tuning rules. Technical Report AOD-01-13, Dublin Institute of Technology, Ireland (2001)
417. O'Dwyer, A.: PID compensation of time delayed processes 1998–2002: a survey. In: Proc. American Control Conference, Denver, USA, pp. 1494–1499 (2003)
418. O'Dwyer, A.: Handbook of PI and PID Controller Tuning Rules, 1st edn. Imperial College Press, London (2003)
419. O'Dwyer, A.: Handbook of PI and PID Controller Tuning Rules, 2nd edn. Imperial College Press, London (2006)
420. O'Dwyer, A.: Handbook of PI and PID Controller Tuning Rules, 3rd edn. Imperial College Press, London (2009)
421. Ogawa, S.: PI controller tuning for robust performance. In: Proc. 4th IEEE Conference on Control Applications, pp. 101–106 (1995)
422. Ogawa, M., Katayama, T.: A robust tuning method for I-PD controller incorporating a constraint on manipulated variable. Trans. Soc. Instrum. Control Eng. **E-1**(1), 265–273 (2001)
423. Okada, Y., Yamakawa, Y., Yamazaki, T., Kurosu, S.: Tuning method of PI controller for given damping coefficient. In: Proc. SICE-ICASE International Joint Conference, Busan, Korea, pp. 5204–5207 (2006)
424. Okada, Y., Yamakawa, Y., Yamazaki, T., Kurosu, S.: Tuning method of PID controller for desired damping coefficient. In: Proc. SICE Annual Conference, Takamatsu, Japan, pp. 795–799 (2007)
425. OMEGA Books: Introduction to temperature controllers (2005). Available at http://www.gii.upv.es/personal/gbenet/IIN/tema_transductores/OmegaBooks/tech_ref_temp.pdf, Z111–Z117. Cited 4 January 2011
426. Oppelt, W.: Einige Faustformeln zur Einstellung von Regelvorgängen. Chem. Ing. Tech. **23**(8), 190–193 (1951) (in German)
427. Ou, W.-H., Chen, Y.-W.: Adaptive actual PID control with an adjustable identification interval. Chem. Eng. Commun. **134**, 93–105 (1995)
428. Ou, L., Gu, D., Zhang, W., Cai, Y.: H_∞ PID controller stabilization for stable processes with time delay. In: Proc. IEEE International Conference on Industrial Technol, Hong Kong, China, pp. 655–659 (2005)
429. Ou, L., Tang, Y., Gu, D., Zhang, W.: Stability analysis of PID controllers for integral processes with time delay. In: Proc. American Control Conference, Portland, USA, pp. 4247–4252 (2005)

430. Ou, L., Zhang, W., Gu, D.: Stabilization of LTI time-delayed processes using analytical PID controllers. In: Proc. IFAC World Congress, Prague, Czech Republic (2005)
431. Oubrahim, R., Leonard, F.: PID tuning by a composed structure. In: Proc. UKACC International Conference on Control, Swansea, UK, vol. 2, pp. 1333–1338 (1998)
432. Ozawa, K., Noda, Y., Yamazaki, T., Kamimura, K., Kurosu, S.: A tuning method for PID controller using optimisation subject to constraints on control input. ASHRAE Trans. **109**, 79–88 (2003)
433. Padhy, P.K., Majhi, S.: Relay based PI-PD design for stable and unstable FOPDT processes. Comput. Chem. Eng. **30**(5), 790–796 (2006)
434. Padhy, P.K., Majhi, S.: An exact method for on-line identification of FOPDT processes. In: Proc. IEEE International Conference on Industrial Technol., Mumbai, India, pp. 1528–1532 (2006)
435. Pagola, F.L., Pecharromán, R.R.: PID auto-tuning based on a second point of frequency response. In: Proc. IFAC World Congress, Barcelona, Spain (2002). Available at http://www.iit.upcomillas.es/docs/02FLPH01.pdf. Cited 4 January 2011
436. Pai, N.-S., Chang, S.-C., Huang, C.-T.: Tuning PI/PID controllers for integrating processes with deadtime and inverse response by simple calculations. J. Process Control **20**, 726–733 (2010)
437. Panda, R.C.: Synthesis of PID tuning rule using the desired closed-loop response. Ind. Eng. Chem. Res. **47**(22), 8684–8692 (2008)
438. Panda, R.C.: Synthesis of PID controller for unstable and integrating processes. Chem. Eng. Sci. **64**, 2807–2816 (2009)
439. Panda, R.C., Yu, C.-C., Huang, H.-P.: PID tuning rules for SOPDT systems: review and some new results. ISA Trans. **43**, 283–295 (2004)
440. Panda, R.C., Hung, S.-B., Yu, C.-C.: An integrated modified smith predictor with PID controller for integrator plus deadtime processes. Ind. Eng. Chem. Res. **45**, 1397–1407 (2006)
441. Paraskevopoulos, P.N., Pasgianos, G.D., Arvanitis, K.G.: New tuning and identification methods for unstable first order plus dead-time processes based on pseudoderivative feedback control. IEEE Trans. Control Syst. Technol. **12**, 455–464 (2004)
442. Paraskevopoulos, P.N., Pasgianos, G.D., Arvanitis, K.G.: PID-type controller tuning for unstable first order plus dead time processes based on gain and phase margin specifications. IEEE Trans. Control Syst. Technol. **14**(5), 926–936 (2006)
443. Park, H.I., Sung, S.W., Lee, I.-B., Lee, J.: A simple autotuning method using proportional controller. Chem. Eng. Commun. **161**, 163–184 (1997)
444. Park, J.H., Sung, S.W., Lee, I.-B.: An enhanced PID control strategy for unstable processes. Automatica **34**, 751–756 (1998)
445. Parr, E.A.: Industrial Control Handbook, vol. 3. BSP Professional Books, Oxford (1989)
446. Paz Ramos, M.A., Morales, L.E.M., Juan, L.B.M., Bazán, G.R.: Genetic rules to tune proportional + derivative controllers for integrative processes with time delays. In: Proc. 15th International Conference on Electronics, Communications and Computers, Puebla, Mexico (2005)
447. Pecharromán, R.: Private communication, 9 May 2000
448. Pecharromán, R.: Ajuste automático de reguladores industriales. Algoritmo robustos de identificación y síntesis. Ph.D. thesis, Universidad Pontificia Comillas, Madrid, Spain (2000) (in Spanish)
449. Pecharromán, R.R., Pagola, F.L.: Control design for PID controllers auto-tuning based on improved identification. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 89–94 (2000)
450. Pemberton, T.J.: PID: the logical control algorithm. Control Eng. **19**(5), 66–67 (1972)
451. Pemberton, T.J.: PID: the logical control algorithm II. Control Eng. **19**(7), 61–63 (1972)
452. Peng, H., Wu, S.-C.: Identification and control of typical industrial process. J. Cent. South Univ. Technol. **7**, 165–169 (2000)
453. Penner, A.: Tuning rules for a PI controller. In: Proc. ISA International Conference and Exhibition, Houston, USA, vol. 43, Part 3, pp. 1037–1051 (1988)

454. Perić, N., Petrović, I., Branica, I.: A method of PID controller autotuning. In: Proc. IFAC Conference on Control of Industrial Syst, Belfort, France, pp. 597–602 (1997)
455. Pessen, D.W.: Optimum three-mode controller settings for automatic start-up. Trans. ASME **75**, 843–849 (1953)
456. Pessen, D.W.: How to "tune in" a three-mode controller. Instrumentation **7**(3), 29–32 (1954)
457. Pessen, D.W.: A new look at PID-controller tuning. J. Dyn. Syst. Meas. Control **116**(3), 553–557 (1994)
458. Petterson, T.S.: Tuning and control strategy for an offshore process subject to minimum environmental impact (2007). Available at http://www.nt.ntnu.no/users/skoge/diplom/prosjekt07/pettersen/Rapport.pdf. Cited 4 January 2011
459. Pettit, J.W., Carr, D.M.: Self-tuning controller. US Patent No. 4,669,040 (1987)
460. Pinnella, M.J., Wechselberger, E., Hittle, D.C., Pedersen, C.O.: Self-tuning digital integral control. ASHRAE Trans. **92**(2B), 202–209 (1986)
461. PMA (Prozeß-und Maschinen-Automation) GmbH: Industrieregler KS40-1, KS41-1, KS42-1 Manuelle Optimierung, p. 18 (2006). Available at http://www.pma-online.de/de/pdf/ba_ks40-1_41-1_42-1_d_9499-040-62718.pdf. Cited 4 January 2011 (in German)
462. Pohjola, M.: PID controller design in networked control systems. M.Sc Thesis, Department of Automation and Systems Technology, Helsinki University of Technology, Finland (2006)
463. Pohjola, M.: Adaptive jitter margin PID controller. In: Proc. 4th IEEE Conference on Automation Science and Engineering, Washington DC, USA, pp. 534–539 (2008)
464. Polonyi, M.J.G.: PID controller tuning using standard form optimization. Control Eng. **March**, 102–106 (1989)
465. Pomerleau, A., Poulin, É.: Manipulated variable based PI tuning and detection of poor settings: an industrial experience. ISA Trans. **43**, 445–457 (2004)
466. Potočnik, B., Škrjanc, I., Matko, D., Zupančič, B.: Samonastavljivi PI regulator na podlagi metode z relejskim preskusom. Elektroteh. Vestn. **68**, 115–122 (2001) (in Slovenian)
467. Poulin, E., Pomerleau, A.: PID tuning for integrating and unstable processes. IEE Proc., Control Theory Appl. **143**, 429–435 (1996)
468. Poulin, E., Pomerleau, A.: Unified PID design method based on a maximum peak resonance specification. IEE Proc., Control Theory Appl. **144**, 566–574 (1997)
469. Poulin, E., Pomerleau, A.: PI settings for integrating processes based on ultimate cycle information. IEEE Trans. Control Syst. Technol. **7**, 509–511 (1999)
470. Poulin, E., Pomerleau, A., Desbiens, A., Hodouin, D.: Development and evaluation of an auto-tuning and adaptive PID controller. Automatica **32**, 71–82 (1996)
471. Pramod, S., Chidambaram, M.: Closed loop identification of transfer function model for unstable bioreactors for tuning PID controllers. Bioprocess Biosyst. Eng. **22**, 185–188 (2000)
472. Prashanti, G., Chidambaram, M.: Set-point weighted PID controllers for unstable systems. J. Franklin Inst. **337**, 201–215 (2000)
473. Prokop, R., Korbel, J.: Relé ve zpětné vazbě aneb převrat v návrhu regulátorů. Automatizace **48**(3), 190–195 (2006) (in Czech)
474. Prokop, R., Husták, P., Prokopová, Z.: Robust PID-like controllers—design and tuning. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 320–325 (2000)
475. Prokop, R., Korbel, J., Matušů, R.: PI autotuners based on biased relay identification. In: Proc. IFAC World Congress, Prague, Czech Republic (2005)
476. Pulkkinen, J., Koivo, H.N., Mäkelä, K.: Tuning of a robust PID controller—application to heating process in extruder. In: Proc. 2nd IEEE Conference on Control Applications, Vancouver, Canada, pp. 811–816 (1993)
477. Ramasamy, M., Sundaramoorthy, S.: PID controller tuning for desired closed loop responses for SISO systems using impulse response. Comput. Chem. Eng. **32**, 1773–1788 (2008)
478. Rao, A.S., Chidambaram, M.: Control of unstable processes with two RHP poles, a zero and time delay. Asia-Pac. J. Chem. Eng. **1**, 63–69 (2006)
479. Rao, A.S., Chidambaram, M.: Enhanced two-degrees-of-freedom control strategy for second-order unstable processes with time delay. Ind. Eng. Chem. Res. **45**, 3604–3614 (2006)

480. Rao, A.S., Rao, V.S.R., Chidambaram, M.: Direct synthesis-based controller design for integrating processes with time delay. J. Franklin Inst. **346**, 38–56 (2009)
481. Ream, N.: The calculation of process control settings from frequency characteristics. Trans. Soc. Instrum. Technol. **6**(1), 19–28 (1954)
482. Reswick, J.B.: Disturbance-response feedback – a new control concept. Trans. ASME **78**, 153–162 (1956)
483. Rhinehart, R.R.: The century's greatest contributions to control practice. ISA Trans. **39**, 3–13 (2000)
484. Rice, R.C.: A rule based design methodology for the control of non-self regulating processes. Ph.D. thesis, University of Connecticut, USA (2004)
485. Rice, R., Cooper, D.J.: Design and tuning of PID controllers for integrating (non-self-regulating) processes. In: Proc. ISA Annual Meeting, Chicago, USA, vol. 424 (2002). Paper P057
486. Rice, B., Cooper, D.: Recognizing integrating (non-self regulating) process behavior (2006). Available at http://www.controlguru.com/wp/p79.html. Cited 4 January 2011
487. Rice, R., Cooper, D.J.: Improve control of liquid level loops. Chemical Eng. Prog. **June**, 54–61 (2008)
488. Rivera, D.E., Jun, K.S.: An integrated identification and control design methodology for multivariable process system applications. IEEE Control Syst. Mag. **20**(3), 25–37 (2000)
489. Rivera, D.E., Morari, M., Skogestad, S.: Internal model control. 4. PID controller design. Ind. Eng. Chem. Process Des. Dev. **25**, 252–265 (1986)
490. Robbins, L.A.: Tune control loops for minimum variability. Chemical Eng. Prog. **January**, 68–70 (2002)
491. Rotach, Y.Va.: Calculation of the robust settings of automatic controllers. Therm. Eng. **41**, 764–769 (1994)
492. Rotach, Y.Va.: Automatic tuning of PID-controllers—expert and formal methods. Therm. Eng. **42**, 794–800 (1995)
493. Rotstein, G.E., Lewin, D.E.: Simple PI and PID tuning for open-loop unstable systems. Ind. Eng. Chem. Res. **30**, 1864–1869 (1991)
494. Rovira, A.A., Murrill, P.W., Smith, C.L.: Tuning controllers for setpoint changes. Instrum. Control Syst. **42**, 67–69 (1969)
495. Rutherford, C.I.: The practical application of frequency response analysis to automatic process control. Proc. Inst. Mech. Eng. **162**, 334–354 (1950)
496. Sadeghi, J., Tych, W.: Deriving new robust adjustment parameters for PID controllers using scale-down and scale-up techniques with a new optimization method. In: Proc. ICSE: 16th Conference on Syst. Engineering, Coventry, UK, pp. 608–613 (2003)
497. Sain, S.G., Özgen, C.: Identification and tuning of processes with large deadtime. Control Comput. **20**(3), 73–78 (1992)
498. Saito, T., Kawakami, J., Takahashi, S., Suehiro, T., Matsumoto, H., Tachibana, K.: PID controller system. US Patent No. 4,903,192 (1990)
499. Sakai, Y., Nakai, Y., Miyabe, A., Kawano, T.: Self-tuning controller, US Patent 4,881,160 (1989)
500. Schaedel, H.M.: A new method of direct PID controller design based on the principle of cascaded damping ratios. In: Proc. European Control Conference, Brussels, Belgium (1997). Paper WE-A H4
501. Schlegel, M.: Nová metoda pro návrh PI(D) regulátoru—teorie pro praxi (new method for the design of a PI(D) controller—theory for practice). Automatizace **41**(2), 70–78 (1998) (in Czech)
502. Schneider, D.M.: Control of processes with time delays. IEEE Trans. Ind. Appl. **24**, 186–191 (1988)
503. Seborg, D.E., Edgar, T.F., Shah, S.L.: Adaptive control strategies for process control: a survey. AIChE J. **32**, 881–913 (1986)
504. Seborg, D.E., Edgar, T.F., Mellichamp, D.A.: Process Dynamics and Control. Wiley, New York (1989)

505. Seki, H., Ogawa, M., Ohshima, M.: Retuning PID temperature controller for an unstable gas-phase polyolefin reactor. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 473–478 (2000)
506. Setiawan, A., Albright, L.D., Phelan, R.M.: Application of pseudo-derivative-feedback algorithm in greenhouse air temperature control. Comput. Electron. Agric. **26**, 283–302 (2000)
507. Shamsuzzoha, M., Lee, M.: IMC based control system design of PID cascaded filter. In: Proc. SICE-ICASE International Joint Conference, Busan, Korea, pp. 2485–2490 (2006)
508. Shamsuzzoha, M., Lee, M.: Design of robust PID controllers for unstable processes. In: Proc. SICE-ICASE International Joint Conference, Busan, Korea, pp. 3324–3329 (2006)
509. Shamsuzzoha, M., Lee, M.: Design of robust PID controller for the unstable dead time dominant processes. Theories Appl. Chem. Eng. **12**(2), 1478–1481 (2006)
510. Shamsuzzoha, M., Lee, M.: Tuning of integrating and integrating processes with dead time and inverse response. Theories Appl. Chem. Eng. **12**(2), 1482–1485 (2006)
511. Shamsuzzoha, M., Lee, M.: PID controller design strategy for first order time delay processes. Theories Appl. Chem. Eng. **13**(1), 121–124 (2007)
512. Shamsuzzoha, M., Lee, M.: Enhanced performance for two-degree-of-freedom control scheme for second order unstable processes with time delay. In: Proc. International Conference on Control, Automation and Syst., Seoul, Korea, pp. 240–245 (2007)
513. Shamsuzzoha, M., Lee, M.: IMC-PID controller design for improved disturbance rejection of time delayed processes. Ind. Eng. Chem. Res. **46**, 2077–2081 (2007)
514. Shamsuzzoha, M., Lee, M.: An enhanced performance PID filter controller for first order time delay processes. J. Chem. Eng. Jpn. **40**(6), 501–510 (2007)
515. Shamsuzzoha, M., Lee, M.: PID controller design for integrating processes with time delay. Korean J. Chem. Eng. **25**(4), 637–645 (2008)
516. Shamsuzzoha, M., Lee, M.: Design of advanced PID controller for enhanced disturbance rejection of second-order processes with time delay. AIChE J. **54**(6), 1526–1536 (2008)
517. Shamsuzzoha, M., Skogestad, S.: The setpoint overshoot method: a simple and fast closed-loop approach for PID tuning. J. Process Control **20**, 1220–1234 (2010)
518. Shamsuzzoha, M., Lee, K., Lee, M., Lee, J.: Design of IMC filter for improved disturbance rejection of PID controller. Theories Appl. Chem. Eng. **10**(2), 1288–1291 (2004)
519. Shamsuzzoha, M., Lee, M., Lee, J.: IMC-PID controller tuning for improved disturbance rejection of unstable time delay processes. Theories Appl. Chem. Eng. **11**(2), 1782–1785 (2005)
520. Shamsuzzoha, M., Park, J., Lee, M.: IMC based method for control system design of PID cascaded filter. Theories Appl. Chem. Eng. **12**(1), 111–114 (2006)
521. Shamsuzzoha, M., Park, J., Lee, M.: PID controller design for unstable process with negative/positive zero. Theories Appl. Chem. Eng. **12**(2), 1474–1477 (2006)
522. Shamsuzzoha, M., Lee, M., Park, J.: Robust PID controller design of time delay processes with/without zero. In: Proc. IEEE International Conference on Industrial Technol, pp. 2256–2261 (2006)
523. Shamsuzzoha, M., Jeon, J., Lee, M.: Improved analytical PID controller design for the second order unstable process with time delay. In: Proc. European Symposium on Computer Aided Process Engineering, Bucharest, Romania, pp. 1–6 (2007)
524. Shamsuzzoha, M., Yoon, M., Lee, M.: Analytical controller design of integrating and first order unstable time delay process. In: Preprints Proc. 8th International IFAC Symposium on Dynamics and Control of Process Syst, Cancún, Mexico, vol. 2, pp. 397–402 (2007)
525. Shen, J.-C.: Tuning PID controller for a plant with under-damped response. In: Proc. IEEE International Conference on Control Applications, Hawaii, USA, pp. 115–120 (1999)
526. Shen, J.-C.: New tuning method for PID control of a plant with underdamped response. Asian J. Control **2**(1), 31–41 (2000)
527. Shen, J.-C.: New tuning method for PID controller. ISA Trans. **41**, 473–484 (2002)
528. Shi, J., Lee, W.S.: IMC-PID controllers for first-order plus dead-time processes: a simple design with guaranteed phase margin. In: Proc. IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, Beijing, China, pp. 1397–1400 (2002)

529. Shi, J., Lee, W.S.: Set point response and disturbance rejection tradeoff for second-order plus dead time processes. In: Proc. 5th Asian Control Conference, Melbourne, Australia, pp. 881–887 (2004)
530. Shigemasa, T., Iino, Y., Kanda, M.: Two degrees of freedom PID auto-tuning controller. In: Proc. ISA International Conference and Exhibition. Advances in Instrumentation and Control, pp. 703–711 (1987)
531. Shin, C.-H., Yoon, M.-H., Park, I.-S.: Automatic tuning algorithm of the PID controller using two Nyquist points identification. In: Proc. Society of Instrument and Control Engineers Annual Conference, Tokyo, Japan, pp. 1225–1228 (1997)
532. Shinskey, F.G.: Process Control Systems—Application, Design and Tuning, 3rd edn. McGraw-Hill, New York (1988)
533. Shinskey, F.G.: Feedback Controllers for the Process Industries. McGraw-Hill, New York (1994)
534. Shinskey, F.G.: Process Control Systems—Application, Design and Tuning, 4th edn. McGraw-Hill, New York (1996)
535. Shinskey, F.G.: PID-deadtime control of distributed processes. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 14–18 (2000)
536. Shinskey, F.G.: PID-deadtime control of distributed processes. Control Eng. Pract. **9**, 1177–1183 (2001)
537. Shinskey, F.G.: Process control diagnostics (2003). Available at http://www.isa.org/CustomSource/ISA/Div_PDFs/PDF_News/CPI_1.pdf. Cited 4 January 2011
538. Sklaroff, M.: Adaptive controller in a process control system and a method therefor. US Patent 5,170,341 (1992)
539. Skoczowski, S.: Model following PID control with a fast model. In: Proc. 6th Portuguese Conference on Automatic Control, pp. 494–499 (2004)
540. Skoczowski, S., Tarasiejski, L.: Tuning of PID controllers based on gain and phase margin specifications using Strejc's process model with time delay. In: Proc. 3rd International Symposium on Methods and Models in Automation and Robotics, Miedzyzdroje, Poland, pp. 765–770 (1996)
541. Skogestad, S.: Probably the best simple PID tuning rules in the world. AIChE Annual Meeting, Reno, USA (2001). Available at http://www.nt.ntnu.no/users/skoge/publications/2001/skogestad_reno/tuningpaper.pdf. Cited 4 January 2011
542. Skogestad, S.: Simple analytic rules for model reduction and PID controller tuning. J. Process Control **13**, 291–309 (2003)
543. Skogestad, S.: Lower limit on controller gain for acceptable disturbance rejection. In: Proc. International Symposium on Advanced Control of Chemical Processes, Hong Kong, China (2004)
544. Skogestad, S.: Simple analytic rules for model reduction and PID controller tuning. Model. Identif. Control **25**, 85–120 (2004)
545. Skogestad, S.: (2004). Available at http://www.nt.ntnu.no/users/skoge/publications/2003/tuningPID/more/extensions/oscillating.txt. Cited 4 January 2011
546. Skogestad, S.: Tuning for smooth PID control with acceptable disturbance rejection. Ind. Eng. Chem. Res. **45**, 7817–7822 (2006)
547. Slätteke, O.: Modeling and control of the paper machine drying section. Ph.D. thesis, Lund University, Sweden (2006)
548. Smith, L.: A modified Smith predictor for extruded diameter control. In: InstMC Mini Symposium in UKACC International Conference on Control, Swansea, UK (1998). Lecture 5
549. Smith, C.L.: Intelligently tune PI controllers. Chemical Eng. **August**, 169–177 (2002)
550. Smith, C.L.: Intelligently tune PID controllers. Chemical Eng. **January**, 56–62 (2003)
551. Smith, C.A., Corripio, A.B.: Principles and Practice of Automatic Process Control, 2nd edn. Wiley, New York (1997)
552. Smith, C.L., Corripio, A.B., Martin, J.: Controller tuning from simple process models. Instrum. Technol. **December**, 39–44 (1975)

553. Solera, E.: Sintonización de controladores PI/PID con los criterios IAE e ITAE, para plantas de polo doble. Proyecto Eléctrico, Universidad de Costa Rica (2005). Available at http://www.eie.ucr.ac.cr/uploads/file/proybach/pb0508t.pdf. Cited 4 January 2011 (in Spanish)
554. Somani, M.K., Kothare, M.V., Chidambaram, M.: Design formulae for PI controllers. Hung. J. Ind. Chem. **20**, 205–211 (1992)
555. Sree, R.P., Chidambaram, M.: Simple method of tuning PI controllers for stable inverse response systems. J. Indian Inst. Sci. **83**, 73–85 (2003)
556. Sree, R.P., Chidambaram, M.: Control of unstable bioreactor with dominant unstable zero. Chem. Biochem. Eng. Q. **17**(2), 139–145 (2003)
557. Sree, R.P., Chidambaram, M.: A simple method of tuning PI controllers for unstable systems with a zero. Chem. Biochem. Eng. Q. **17**(3), 207–212 (2003)
558. Sree, R.P., Chidambaram, M.: Control of unstable reactor with an unstable zero. Indian Chem. Eng. Sect. A **46**(1), 21–26 (2004)
559. Sree, R.P., Chidambaram, M.: Set point weighted PID controllers for unstable systems. Chem. Eng. Commun. **192**, 1–13 (2005)
560. Sree, R.P., Chidambaram, M.: A simple and robust method of tuning PID controllers for integrator/dead time processes. J. Chem. Eng. Jpn. **38**(2), 113–119 (2005)
561. Sree, R.P., Chidambaram, M.: Control of Unstable Systems. Alpha Science International, Oxford (2006)
562. Sree, R.P., Srinivas, M.N., Chidambaram, M.: A simple method of tuning PID controllers for stable and unstable FOPTD systems. Comput. Chem. Eng. **28**, 2201–2218 (2004)
563. Srinivas, M.N., Chidambaram, M.: Set-point weighted PID controllers. In: Proc. International Conference on Quality, Reliability and Control, Mumbai, India, pp. C22-1–C22-5 (2001)
564. Srividya, R., Chidambaram, M.: On-line controllers tuning for integrator plus delay systems. Process Control Qual. **9**, 59–66 (1997)
565. St. Clair, D.W.: Controller Tuning and Control Loop Performance, 2nd edn. Straight Line Control CO, Newark (1997)
566. Streeter, M.J., Keane, M.A., Koza, J.R.: Automatic synthesis using genetic programming of improved PID tuning rules. In: Ruano, A.D. (ed.) Preprint Proc. Intelligent Control Syst. and Signal Processing Conference, pp. 494–499 (2003)
567. Sung, S.W., O, J., Lee, I.-B., Lee, J., Yi, S.-H.: Automatic tuning of PID controller using second-order plus time delay model. J. Chem. Eng. Jpn. **29**, 991–999 (1996)
568. Suyama, K.: A simple design method for sampled-data PID control systems with adequate step responses. In: Proc. International Conference on Industrial Electronics, Control, Instrumentation and Automation, pp. 1117–1122 (1992)
569. Suyama, K.: A simple design method for sampled-data I-PD control systems. In: Proc. Annual Conference of the IEEE, Industrial Electronics Society, Hawaii, USA, pp. 2293–2298 (1993)
570. Syrcos, G., Kookos, I.K.: PID controller tuning using mathematical programming. Chem. Eng. Process. **44**, 41–49 (2005)
571. Tachibana, Y.: Identification of a system with dead time and its application to auto tuner. Electr. Eng. Jpn. **104**, 128–137 (1984)
572. Taguchi, H., Araki, M.: Two-degree-of-freedom PID controllers—their functions and optimal tuning. In: Preprints Proc. PID ’00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 95–100 (2000)
573. Taguchi, H., Araki, M.: On tuning of two-degree-of-freedom PID controllers with consideration on location of disturbance input. Trans. Soc. Instrum. Control Eng. **38**(5), 441–446 (2002) (in Japanese)
574. Taguchi, H., Doi, M., Araki, M.: Optimal parameters of two-degrees-of-freedom PID control systems. Trans. Soc. Instrum. Control Eng. **23**(9), 889–895 (1987) (in Japanese)
575. Taguchi, H., Kokawa, M., Araki, M.: Optimal tuning of two-degree-of-freedom PD controllers. In: Proc. Asian Control Conference, Singapore, pp. 268–273 (2002)

576. Takatsu, H., Itoh, T.: Future needs for control theory in industry—report of the control technology survey of Japanese industry. IEEE Trans. Control Syst. Technol. **7**(3), 298–305 (1999)
577. Tan, K.K., Lee, T.H., Wang, Q.G.: Enhanced automatic tuning procedure for process control of PI/PID controller. AIChE J. **42**, 2555–2562 (1996)
578. Tan, W., Liu, J., Sun, W.: PID tuning for integrating processes. In: Proc. IEEE International Conference on Control Applications, Trieste, Italy, vol. 2, pp. 873–876 (1998)
579. Tan, W., Liu, K., Tam, P.K.S.: PID tuning based on loop-shaping H_∞ control. IEE Proc., Control Theory Appl. **145**(6), 485–490 (1998)
580. Tan, W., Yuan, Y., Niu, Y.: Tuning of PID controller for unstable process. In: Proc. IEEE International Conference on Control Applications, Hawaii, USA, pp. 121–124 (1999)
581. Tan, K.K., Wang, Q.G., Hang, C.C., Hägglund, T.J.: Advances in PID Control, Advances in Industrial Control Series. Springer, London (1999)
582. Tan, K.K., Lee, T.H., Jiang, X.: On-line relay identification, assessment and tuning of PID controller. J. Process Control **11**, 483–496 (2001)
583. Tang, W., Shi, S., Wang, M.: Autotuning PID control for large time-delay processes and its application to paper basis weight control. Ind. Eng. Chem. Res. **41**, 4318–4327 (2002)
584. Tang, W., Wang, M., Chao, Y., He, L., Itoh, H.: A study on the internal relationship among Smith predictor, Dahlin controller & PID. In: Proc. IEEE International Conference on Automation and Logistics, Jinan, China, pp. 3101–3106 (2007)
585. Tavakoli, S., Banookh, A.: Robust PI control design using particle swarm optimization. J. Comput. Sci. Eng. **1**(1), 36–41 (2010)
586. Tavakoli, S., Fleming, P.: Optimal tuning of PI controllers for first order plus dead time/long dead time models using dimensional analysis. In: Proc. European Control Conference, Cambridge, UK (2003)
587. Tavakoli, S., Tavakoli, M.: Optimal tuning of PID controllers for first order plus time delay models using dimensional analysis. In: Proc. 4th IEEE International Conference on Control and Automation, Canada, pp. 942–946 (2003)
588. Tavakoli, S., Griffin, I., Fleming, P.J.: Robust PI controller for load disturbance rejection and setpoint regulation. In: Proc. IEEE Conference on Control Applications, Toronto, Canada, pp. 1015–1020 (2005)
589. Tavakoli, S., Griffin, I., Fleming, P.J.: Tuning of decentralized PI (PID) controllers for TITO processes. Control Eng. Pract. **14**, 1069–1080 (2006)
590. Tavakoli, S., Griffin, I., Fleming, P.J.: Multi-objective optimization approach to the PI tuning problem. In: Proc. IEEE Congress on Evolutionary Computation, Singapore, pp. 3165–3171 (2007)
591. Thomasson, F.Y.: Tuning guide for basic control loops. In: Proc. Process Control, Electrical and Information Conference, pp. 137–148 (1997)
592. Thyagarajan, T., Yu, C.-C.: Improved autotuning using the shape factor from relay feedback. Ind. Eng. Chem. Res. **42**, 4425–4440 (2003)
593. Thyagarajan, T., Esakkiappan, C., Sujatha, V.: Modelling and control of inverse response process with time delay using relay feedback test. In: Proc. International Conference on Modelling, Identification and Control, Okayama, Japan, pp. 494–499 (2010)
594. Tinham, B.: Tuning PID controllers. Control Instrum. **September**, 79–83 (1989)
595. Trybus, L.: A set of PID tuning rules. Arch. Control Sci. **15(LI)**(1), 5–17 (2005)
596. Tsang, K.M., Rad, A.B.: A new approach to auto-tuning of PID controllers. Int. J. Syst. Sci. **26**(3), 639–658 (1995)
597. Tsang, K.M., Rad, A.B., To, F.W.: Online tuning of PID controllers using delayed state variable filters. In: Proc. IEEE Region 10 Conference on Computer, Communication, Control and Power Engineering, Beijing, China, vol. 4, pp. 415–419. (1993)
598. Tyreus, B.D., Luyben, W.L.: Unusual dynamics of a reactor/preheater process with deadtime, inverse response and openloop instability. J. Process Control **3**(4), 241–251 (1992)
599. Umamaheswari, S., Palanisamy, V., Chidambaram, M.: A simple method of tuning PI controllers for interval plant of cold rolling mill. Int. J. Recent Trends Eng. **1**(4), 41–45 (2009)

600. Umamaheswari, S., Palanisamy, V., Chidambaram, M.: A simple method of tuning PID controllers for interval plant of cold rolling mill. In: Proc. International Conference on Control, Automation, Communication and Energy Conservation, Erode, India, pp. 1–6 (2009)
601. Unar, M.A., Murray-Smith, D.J., Shah, S.F.A.: Technical Report CSC-96016 (1996). Available at http://www.mech.gla.ac.uk/Research/Control/Publications/Rabstracts/abs96016.html. Cited 14 December 2010
602. Universal Dynamic Technologies: Brainwave: The New Concept in Process Control. Sales Literature (1998)
603. Urrea, R., Castellanos-Sahagun, E., Alvarez, J., Alvarez-Ramirez, J.: Distillate cascade composition control using a two-temperature measurement secondary component. Ind. Eng. Chem. Res. **45**, 6828–6841 (2006)
604. Valentine, C.C., Chidambaram, M.: Robust PI and PID control of stable first order plus time delay systems. Indian Chem. Eng. Sect. A **39**(1), 9–14 (1997)
605. Valentine, C.C., Chidambaram, M.: PID control of unstable time delay systems. Chem. Eng. Commun. **162**, 63–74 (1997)
606. Valentine, C.C., Chidambaram, M.: Robust control of unstable first order plus time delay systems. Indian Chem. Eng. Sect. A **40**(1), 19–23 (1998)
607. Van der Grinten, P.M.E.M.: Finding optimum controller settings. Control Eng. **December**, 51–56 (1963)
608. VanDoren, V.J.: Ziegler–Nichols methods facilitate loop tuning. Control Eng. December 1998
609. Van Overschee, P., De Moor, B.: RaPID: the end of heuristic PID tuning. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 687–692 (2000)
610. Velázquez-Figueroa, C.: Automated rule-based dynamic modelling and controller design. Ph.D. thesis, University of Connecticut, USA (1997)
611. Venkatashankar, V., Chidambaram, M.: Design of P and PI controllers for unstable first order plus time delay systems. Int. J. Control **60**, 137–144 (1994)
612. Vilanova, R.: PID controller tuning rules for robust step-response of first-order-plus-dead-time models. In: Proc. American Control Conference, Minneapolis, USA, pp. 256–261 (2006)
613. Vilanova, R., Balaguer, P.: ISA-PID controller tuning: a combined min-max/ise approach. In: Proc. IEEE International Conference on Control Applications, Munich, Germany, pp. 2956–2961 (2006)
614. Visioli, A.: Optimal tuning of PID controllers for integral and unstable processes. IEE Proc., Control Theory Appl. **148**(1), 180–184 (2001)
615. Visioli, A.: Improving the load disturbance rejection performances of IMC-tuned PID controllers. In: Proc. IFAC World Congress, Barcelona, Spain, pp. 295–300 (2002)
616. Visioli, A.: Experimental evaluation of a Plug&Control strategy for level control. In: Proc. IFAC World Congress, Prague, Czech Republic (2005)
617. Visioli, A.: Practical PID Control. Springer, London (2006)
618. Vítečková, M.: Seřízení číslicových I analogových regulátorů pro regulované soustavy s dopravním zpožděním [Digital and analog controller tuning for processes with time delay]. Automatizace **42**(2), 106–111 (1999) (in Czech)
619. Vítečková, M.: Ukazatelé kvality pro regulační obvody seřízené metodou inverze dynamiki. In: Proc. ASR 2001 Seminar, Instruments and Control, Ostrava, Czech Republic, pp. 1–6 (2001) (in Czech)
620. Vítečková, M.: Simple PI and PID controller tuning. Sb. Věd. Pr. Vysoké šk. Báň.-Tech. Univ. Ostrava, Řada Strojní **2**(LII), 225–230 (2006). Available at http://www.fs.vsb.cz/transactions/2006-2/1562_VITECKOVA_Miluse.pdf. Cited 4 January 2011
621. Vítečková, M., Víteček, A.: Analytical controller tuning method for proportional non-oscillatory plants with time delay. In: Proc. International Carpathian Control Conference, Malenovice, Czech Republic, pp. 297–302 (2002)
622. Vítečková, M., Víteček, A.: Analytical digital and analog controller tuning method for proportional non-oscillatory plants with time delay. In: Proc. 2nd IFAC Conference on Control Syst. Design, Bratislava, Slovak Republic, pp. 59–64 (2003)

623. Vítečková, M., Víteček, A.: Controller tuning for integral plus time delay plants. In: Sborník vědeckých prací Vysoké školy báňské—Technické univerzity Ostrava, Ostrava, Czech Republic, pp. 159–166 (2007). Available at http://www.fs.vsb.cz/transactions/2007-2/1571_VITECKOVA_Miluse_VITECEK_Antonin.pdf. Cited 4 January 2011
624. Vítečková, M., Víteček, A.: Two-degree of freedom controller tuning for integral plus time delay plants. ICIC Express Lett. **2**(3), 225–229 (2008)
625. Vítečková, M., Víteček, A., Smutný, L.: Controller tuning for controlled plants with time delay. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 283–288 (2000)
626. Vítečková, M., Víteček, A., Smutný, L.: Simple PI and PID controllers tuning for monotone self-regulating plants. In: Preprints Proc. PID '00: IFAC Workshop on Digital Control, Terrassa, Spain, pp. 289–294 (2000)
627. Vivek, S., Chidambaram, M.: An improved relay autotuning of PID controllers for unstable FOPDT systems. Comput. Chem. Eng. **29**, 2060–2068 (2005)
628. Voda, A., Landau, I.D.: The autocalibration of PI controllers based on two frequency measurements. Int. J. Adapt. Control Signal Process. **9**, 395–421 (1995)
629. Vrančić, D.: Design of anti-windup and bumpless transfer protection. Part II: PID controller tuning by multiple integration method. PhD thesis, University of Ljubljana, J. Stefan Institute, Ljubljana, Slovenia (1996)
630. Vrančić, D., Lumbar, S.: Improving PID controller disturbance rejection by means of magnitude optimum. Report DP-8955. J. Stefan Institute, Ljubljana, Slovenia (2004)
631. Vrančić, D., Peng, Y., Strmčnik, S., Hanus, R.: A new tuning method for PI controllers based on a process step response. In: Proc. CESA '96 IMACS MultiConference Symposium on Control, Optimisation and Supervision, Lille, France, vol. 2, pp. 790–794 (1996)
632. Vrančić, D., Peng, Y., Strmčnik, S., Juričić, D.: A multiple integration tuning method for filtered PID controller. In: Preprints Proc. IFAC 14th World Congress, Beijing, China (1999). Paper 3b-02-3
633. Vrančić, D., Kocijan, J., Strmčnik, S.: Simplified disturbance rejection tuning method for PID controllers. In: Proc. 5th Asian Control Conference, Melbourne, Australia, vol. 1, pp. 492–497 (2004)
634. Vrančić, D., Kristiansson, B., Strmčnik, S.: Reduced MO tuning method for PID controllers. In: Proc. 5th Asian Control Conference, Melbourne, Australia, vol. 1, pp. 460–465 (2004)
635. Wade, H.L.: Regulatory and Advanced Regulatory Control: System Development. ISA, North Carolina, USA (1994)
636. Wang, X.-S.: PID controller tuning method for improving performance. Opt. Precis. Eng. **8**(4), 381–384 (2000) (in Chinese)
637. Wang, J.-Y.: Control system design using the NCD technique. Masters thesis, Feng Chia University, Taiwan. Available at http://ethesys.lib.fcu.edu.tw. Cited 25 October 2006
638. Wang, Y.-G., Cai, W.-J.: PID tuning for integrating processes with sensitivity specification. In: Proc. IEEE Conference on Decision and Control, Orlando, USA, pp. 4087–4091 (2001)
639. Wang, Y.-G., Cai, W.-J.: Advanced proportional-integral-derivative tuning for integrating and unstable processes with gain and phase margin specifications. Ind. Eng. Chem. Res. **41**, 2910–2914 (2002)
640. Wang, T.-S., Clements, W.C.: Adaptive multivariable PID control of a distillation column with unknown and varying dead time. Chem. Eng. Commun. **132**, 1–13 (1995)
641. Wang, L., Cluett, W.R.: Tuning PID controllers for integrating processes. IEE Proc., Control Theory Appl. **144**(5), 385–392 (1997)
642. Wang, L., Cluett, W.R.: From Plant Data to Process Control. Taylor and Francis, New York (2000)
643. Wang, H., Jin, X.: Direct synthesis approach of PID controller for second-order delayed unstable processes. In: Proc. 5th World Congress on Intelligent Control and Automation, Hangzhou, China, pp. 19–23 (2004)
644. Wang, Y.-G., Shao, H.-H.: PID autotuner based on gain- and phase-margin specification. Ind. Eng. Chem. Res. **38**, 3007–3012 (1999)

645. Wang, Y.-G., Shao, H.-H.: Automatic tuning of optimal PI controllers. In: Proc. IEEE Conference on Decision and Control, Phoenix, Arizona, pp. 3802–3803 (1999)
646. Wang, Y.-G., Shao, H.-H.: Optimal tuning for PI controller. Automatica **36**, 147–152 (2000)
647. Wang, Y.-G., Shao, H.-H.: PID auto-tuner based on sensitivity specification. Trans. Inst. Chem. Eng. **78**(A), 312–316 (2000)
648. Wang, Y.-G., Xu, X.-M.: PID tuning for unstable processes with sensitivity specification. In: Proc. Chinese Control and Decision Conference, Guilin, China, pp. 3460–3464 (2009)
649. Wang, F.-S., Juang, W.-S., Chan, C.-T.: Optimal tuning of PID controllers for single and cascade control loops. Chem. Eng. Commun. **132**, 15–34 (1995)
650. Wang, L., Barnes, T.J.D., Cluett, W.R.: New frequency-domain design method for PID controllers. IEE Proc., Control Theory Appl. **142**, 265–271 (1995)
651. Wang, Q.-G., Lee, T.-H., Fung, H.-W., Bi, Q., Zhang, Y.: PID tuning for improved performance. IEEE Trans. Control Syst. Technol. **7**(4), 457–465 (1999)
652. Wang, Q.-G., Zhang, Y., Guo, X.: Robust closed-loop controller auto-tuning. In: Proc. 15th IEEE International Symposium on Intelligent, Control, Rio, Patras, Greece, pp. 133–138 (2000)
653. Wang, Y.-G., Shao, H.-H., Wang, J.: PI tuning for processes with large dead time. In: Proc. American Control Conference, Chicago, USA, pp. 4274–4278 (2000)
654. Wang, Q.-G., Zhang, Y., Guo, X.: Robust closed-loop identification with application to auto-tuning. J. Process Control **11**, 519–530 (2001)
655. Wang, Y.-G., Cai, W.-J., Shi, Z.-G.: PID autotuning for integrating processes with specifications on gain and phase margins. In: Proc. American Control Conference, Arlington, USA, pp. 2181–2185 (2001)
656. Wang, Y., Schinkel, M., Schmitt-Hartmann, T., Hunt, K.J.: PID and PID-like controller design by pole assignment within D-stable regions. Asian J. Control **4**(4), 423–432 (2002)
657. Wang, J.-G., Zhang, J.-G., Zhao, Z.-C.: Extended IMC-PID control for integrator and dead time process. Electr. Mach. Control **9**(2), 133–135 (2005) (in Chinese)
658. Wang, C., Luo, Y., Chen, Y.: Fractional order proportional integral (FOPI) and [proportional integral] (FO[PI]) controller designs for first order plus time delay (FOPTD) systems. In: Proc. Chinese Control and Decision Conference, Guilin, China, pp. 329–334 (2009)
659. Wills, D.M.: Tuning maps for three-mode controllers. Control Eng. **April**, 104–108 (1962)
660. Wills, D.M.: A guide to controller tuning. Control Eng. **August**, 93–95 (1962)
661. Wilton, S.R.: Controller tuning. ISA Trans. **38**, 157–170 (1999)
662. Witt, S.D., Waggoner, R.C.: Tuning parameters for non-PID three-mode controllers. Hydrocarb. Process. **June**, 74–78 (1990)
663. Wojsznis, W.K., Blevins, T.L.: System and method for automatically tuning a process controller. US Patent Number 5,453,925 (1995)
664. Wojsznis, W.K., Blevins, T.L., Thiele, D.: Neural network assisted control loop tuner. In: Proc. IEEE International Conference on Control Applications, Hawaii, USA, vol. 1, pp. 427–431 (1999)
665. Wolfe, W.A.: Controller settings for optimum control. Trans. ASME **73**, 413–418 (1951)
666. Xing, J., Wang, P., Wang, L.: A self-tuning PID controller based on expert system. In: Proc. IFAC New Technologies for Computer Control Conference, Hong Kong, China, pp. 479–484 (2001)
667. Xu, J., Shao, H.: Advanced PID tuning for integrating processes with a new robustness specification. In: Proc. American Control Conference, Denver, USA, pp. 3961–3966 (2003)
668. Xu, J., Shao, H.: Advanced PID tuning for unstable processes based on a new robustness specification. In: Proc. American Control Conference, Denver, USA, pp. 368–372 (2003)
669. Xu, J., Shao, H.: A novel method of PID tuning for integrating processes. In: Proc. 42nd IEEE Conference on Decision and Control, Maui, USA, pp. 139–142 (2003)
670. Xu, J., Shao, H.: A new tuning method of PID controller for integrating processes. Chin. J. Sci. Instrum. **25**(6), 714–716 (2004) (in Chinese)
671. Xu, J., Shao, H.: A new tuning method of PID controller for integrating processes. Chin. J. Sci. Instrum. **25**(6), 720 (2004) (in Chinese)

672. Xu, J., Shao, H.: PI tuning for large dead-time processes with a new robustness specification. J. Syst. Eng. Electron. **15**(3), 333–336 (2004)
673. Xu, J.-H., Sun, R., Shao, H.-H.: PI controller tuning for large dead-time processes. Control Decis. **19**(1), 99–101 (2004) (in Chinese)
674. Xu, Y., Deng, H., Zhang, P., Yang, J.: Tuning PI/PID active queue management controllers supporting TCP/IP flows. In: Proc. International Conference on Communications, Circuits and Syst., Hong Kong, China, vol. 1, pp. 630–634 (2005)
675. Yang, J.C.-Y., Clarke, D.W.: Control using self-validating sensors. Trans. Inst. Meas. Control **18**, 15–23 (1996)
676. Yang, P., Pan, S.: New PID parameters tuning formulae for relay feedback self-tuning. In: Proc. IEEE International Symposium on Knowledge Acquisition and Modeling, Wuhan, China, pp. 567–569 (2008)
677. Yi, C., De Moor, B.L.R.: Robustness analysis and control system design for a hydraulic servo system. IEEE Trans. Control Syst. Technol. **2**(3), 183–197 (1994)
678. Young, A.J.: An Introduction to Process Control System Design. Longman, Green, London (1955)
679. Yu, S.W.: Optimal PI tuning for load disturbances. J. Chin. Inst. Chem. Eng. **19**(6), 349–357 (1988)
680. Yu, C.-C.: Autotuning of PID Controllers. Advances in Industrial Control Series. Springer, London (1999)
681. Yu, C.-C.: Autotuning of PID Controllers, 2nd edn. Advances in Industrial Control Series. Springer, London (2006)
682. Yu, L., Ma, M., Hu, W., Shi, Z., Shu, Y.: Design of parameter tunable robust controller for active queue management based on H_∞ control theory. J. Netw. Comput. Appl. (2010). doi:10.1016/j.jnca.2010.10.006
683. Žáková, K.: One type of controller design for delayed double integrator system. WSEAS Trans. Syst. Control **1**(3), 62–69 (2008)
684. Zamani, Z.T., Moshiri, B., Fatehi, A., Sedigh, A.K.: Relay feedback based monitoring and autotuning of processes with gain nonlinearity. In: Proc. UK Automatic Control Conference, Manchester, UK (2008)
685. Zhang, D.: Process dynamics and controller selection of DCS. In: Proc. ISA Advances in Instrumentation and Control Conference, Anaheim, USA, vol. 49, Part 2, pp. 231–240 (1994)
686. Zhang, W.: Analytical design for process control. Post-doctoral Research Report (1998). Available at http://automation.sjtu.edu.cn/wdzhang/pdfrpt.pdf. Cited 29 September 2009 (in Chinese)
687. Zhang, W.: Optimal design of the refined Ziegler–Nichols proportional-integral-derivative controller for stable and unstable processes with time delays. Ind. Eng. Chem. Res. **45**, 1408–1419 (2006)
688. Zhang, W., Xu, X.: H_∞ PID controller design for runaway processes with time delay. ISA Trans. **41**, 317–322 (2002)
689. Zhang, W.D., Sun, Y.X., Xu, X.M.: Modified PID controller based on H_∞ theory. In: Proc. IEEE International Conference on Industrial Technol, pp. 9–12 (1996)
690. Zhang, G., Shao, C., Chai, T.: A new method for independently tuning PID parameters. In: Proc. Conference on Decision and Control, Kobe, Japan, pp. 2527–2532 (1996)
691. Zhang, W., Sun, Y., Xu, X.: PID control for integrator and dead time process. Acta Autom. Sin. **25**(4), 518–523 (1999) (in Chinese)
692. Zhang, W., Xu, X., Sun, Y.: Quantitative performance design for integrating processes with time delay. Automatica **35**, 719–723 (1999)
693. Zhang, W., Xu, X., Zhang, W.: PID design of unstable processes by using closed loop constraints. J. Shanghai Jiaotong Univ. **34**(5), 589–592 (2000)
694. Zhang, J.-G., Liu, Z.-Y., Pei, R.: Two-degree-of-freedom PID control for integrator and dead time process. Control Decis. **17**(6), 886–889 (2002) (in Chinese)
695. Zhang, J., Li, L., Chen, Z., Zhao, Z.: IMC tuning of two-degree-of-freedom regulator. Chin. J. Sci. Instrum. **23**(1), 28–30 (2002) (in Chinese)

696. Zhang, J., Li, L., Chen, Z., Zhao, Z.: IMC tuning of two-degree-of-freedom regulator. Chin. J. Sci. Instrum. **23**(1), 48 (2002) (in Chinese)
697. Zhang, W., Xi, Y., Yang, G., Xu, X.: Design PID controllers for desired time-domain or frequency-domain response. ISA Trans. **41**, 511–520 (2002)
698. Zhang, W., Gu, D., Xu, X.: A unified approach to design the RZN PID controller for stable and unstable processes with time delay. In: Proc. IEEE Conference on Decision and Control, Maui, USA, pp. 4080–4081 (2003)
699. Zhang, F.-B., Wang, G.-D., Zhang, D.-H., Liu, X.-H.: Optimal ITAE tuning formulae for parameters of PID controller. J. Northeast. Univ. Nat. Sci. **26**(8), 755–758 (2005)
700. Zhang, J., Wang, J., Zhao, Z.: A novel two-degree-of-freedom PID controller for integrator and dead time process. In: Proc. World Congress on Intelligent Control and Automation, Dalian, China, pp. 6388–6391 (2006)
701. Zhao, Y., Jia, L., Cai, W.: The system identification and PID auto-tuning for unstable processes. In: Proc. IEEE Conference on Industrial Electronics and Applications, Singapore, pp. 176–180 (2008)
702. Zhong, Q.-C., Li, H.-X.: 2-degree-of-freedom proportional-integral-derivative-type controller incorporating the Smith principle for processes with dead time. Ind. Eng. Chem. Res. **41**, 2448–2454 (2002)
703. Zhou, X., Dong, X., Zhang, Y., Fang, Y.: Automatic tuning of PI controller for atomic force microscope based on relay with hysteresis. In: Proc. IEEE International Conference on Control Applications, St. Petersburg, Russia, pp. 1271–1275 (2009)
704. Zhuang, M.: Computer-aided PID controller design. Ph.D. thesis. University of Sussex, UK (1992)
705. Zhuang, M., Atherton, D.P.: Automatic tuning of optimum PID controllers. IEE Proc. Part D. Control Theory Appl. **140**, 216–224 (1993)
706. Ziegler, J.G., Nichols, N.B.: Optimum settings for automatic controllers. Trans. Am. Soc. Mech. Eng. **64**, 759–768 (1942)
707. Zou, H., Brigham, S.E.: Process control system with asymptotic auto-tuning. US Patent Number 5,818,714 (1998)
708. Zou, H., Hedstrom, K.P., Warrior, J., Hays, C.L.: Field based process control system with auto-tuning. US Patent Number 5,691,896 (1997)

Chapter 2
Model-Based PI(D) Autotuning

Alberto Leva and Martina Maggio

2.1 Introduction

Any autotuning procedure starts by taking input/output measurements from the process. This can be done in open- or closed-loop, by deliberately injecting some *stimulus* or simply relying on the excitation provided by standard manoeuvres like set point changes, and in a number of different manners, see e.g. the discussions in works like [5, 6, 33] and the references therein.

No matter how the said measurements were obtained, in some cases they are directly employed to determine the regulator parameters. In other cases, the same measurements are conversely used to first obtain a process *model*, i.e. something that permits to simulate the closed-loop system and, sticking to the linear time-invariant single-variable context, virtually always takes the form of a transfer function. That model, together with conveniently expressed specifications, is subsequently used to tune the regulator. Autotuning procedures involving a model in the sense just shown are said to be *model-based*, and model-based autotuning (hereinafter MBAT for short) is the subject of this chapter.

In the literature, many classifications of PI/PID (auto)tuning techniques were proposed, see e.g. again [5] or works like [15, 16, 48, 65, 66], and throughout such a vast research *corpus*, the word "model" is used with various meanings. For example, referring to the synthetic and clear scheme of [52, p. 904], the subject of this chapter would fall into the "parametric model methods" set. It is therefore worth stressing that the distinctive character of MBAT as intended herein is the presence of a process model in a form suitable for analysing and simulating the closed-loop system. The typical workflow of MBAT is summarised in Fig. 2.1.

A. Leva (✉) · M. Maggio
Dipartimento di Elettronica e Informazione, Politecnico di Milano, via Ponzio 34/5,
20133 Milano, Italy
e-mail: leva@elet.polimi.it

M. Maggio
e-mail: maggio@elet.polimi.it

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_2, © Springer-Verlag London Limited 2012

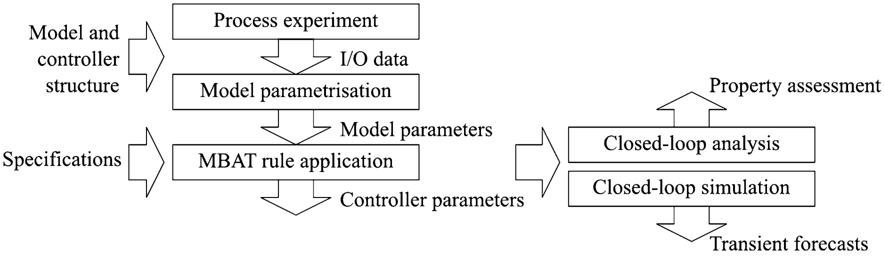


Fig. 2.1 Summary of the typical MBAT workflow

The distinctive character of MBAT just pointed out results in some relevant strengths, but also in substantially two weaknesses. The main strong point is that, as anticipated, the closed-loop system can be simulated before applying the tuned regulator to the real process. Hence, not only structural properties such as stability, performance, and robustness can be checked, but also the most relevant characteristics of the obtained transients, like peak values and durations, can be forecast. Another strength of MBAT is that the control specifications can be stipulated with reference to the model, for example requiring that the closed-loop set point step response be "n times faster" than that of the model in open loop. In general, the MBAT approach allows one to make the specification more readable and easy to interpret, thus comprehensible also by non-specialist personnel, to the advantage of industrial acceptability.

The mentioned weaknesses, on the other hand, both come from the interplay between the model identification (or better *parameterisation*, since the model structure is typically fixed a priori) and the subsequent regulator tuning. In MBAT, the model structure is almost invariantly fixed a priori—the word "almost" being removable if industrial applications are considered—based on that of the regulator to be tuned (e.g. PI or PID). This unavoidably gives rise to process/model mismatches that can adversely affect the closed-loop transients' forecasts and sometimes even the assessment of structural properties such as stability. In addition, the process stimulation used to produce the input/output measurements has to frequently obey potentially strict technological limits, and tuning time is frequently a relevant issue for industrial acceptance. In one word, the model needs obtaining from finite (often quite short) sets of noisy data, in the virtually ubiquitous presence of poor excitation. As a result, the model is typically obtained with ad hoc techniques involving some heuristics, such as the method of areas, of moments, and numerous others.

A chapter of this book is entirely devoted to the "identification for PID" subject, so we do not further delve into the matter here and just point out its two most relevant consequences as seen from the MBAT standpoint. First, even if the used tuning rule is well suited for the particular problem at hand, different model identification methods may cause that rule to produce quite significantly different results, and for the same reason, apparently, the same couple of model identification method and tuning rule may behave very differently in different control problems. In the opinion of the authors, this is maybe the toughest difficulty that MBAT has to face in order to achieve the wide acceptance it potentially deserves. Second, not only the identified model will be nothing more than a (hopefully adequate) approximation of the process dynamics, but the available data will hardly ever—to put it mildly—be sufficient to set up a robustness problem in a formally sound manner. In fact, as shown later on, identification/tuning integration and robustness (or "nonfragility") are nowadays two relevant research lines in the MBAT domain.

In order to discuss the issues sketched above, this chapter is organised as follows. First, Sect. 2.2 proposes a trivial introductory example to establish the chapter's perspective. Section 2.3 then provides a panorama of the major MBAT methods proposed in the literature, referring the reader to more extensive works for the details that cannot fit herein. At the same time, suggestions are given on how to organise the presented methods into a simple yet reasoned taxonomy that can easily incorporate and classify the numerous methods not quoted for space reasons. Section 2.4, in part elaborating from the mentioned taxonomy suggestions, addresses the problem of assessing the tuning results and monitoring the control loop, and also of determining when a new autotuning operation is advisable. Up to this point, reference is made to well-assessed MBAT methods and, wherever possible, to methods that do have an industrial realisation and therefore a backlog of use experience. Section 2.5 conversely deals with modern research issues, not yet or only sparingly reflected in the applications, and illustrates the topics that appear more promising based on the experience and the opinion of the authors. Here it is also impossible to exhaust the matter, and thus references are provided for the subjects not touched here. Finally, in Sect. 2.6 some conclusions are drawn, and possible future research perspectives are briefly sketched.

2.2 A Simple Introductory Example

To get a rapid idea of what MBAT is from *both* the methodological and the engineering point of view, it is useful to start with a simple example and some comments. To this end, the following is a deliberately trivial MBAT procedure to tune a PI in the form $R(s) = K(1 + 1/sT_i)$ for an asymptotically stable process.

1. Lead the process to steady state.
2. Set the PI to manual mode and apply a step control variation.
3. Wait for the controlled variable to settle.
4. Attempt to describe the process with a first-order model with delay. Set the gain μ_M to the ratio between the difference of the values of the final and initial controlled variables and the control step amplitude, the delay D_M to the time needed to complete 5% of the overall transient, and the time constant T_M to the time needed to go from 5% to 70% of the same transient.
5. Attempt to enforce a prescribed cutoff frequency ω_c and phase margin φ_m, privileging the latter if its achievement prevents to attain the former and using a cancellation strategy. Omitting simple computation, this means using the MBAT

rule that

$$T_i = T_M, \qquad K = \min\left(\frac{\omega_c T_M}{\mu_M}, \frac{T_M}{\mu_M D_M}\left(\frac{\pi}{2} - \varphi_m\right)\right). \tag{2.1}$$

The reader is now encouraged to re-consider the example and notice the following facts. There is a MBAT *rule*, namely (2.1), that realises a *tuning policy* (pole/zero cancellation) with an *objective* or *tuning desire* (a given response speed if possible under a stability degree constraint) expressed by *specifications* (ω_c and φ_m). This is done *assuming* a *model structure* (first order plus delay) that will attempt to explain the measured data and depends inherently of the chosen *controller structure*. Finally an *experiment* is designed to *parameterise* the model, and various (reasonable but arbitrary) decisions are taken on how to process the data in order to parameterise the model (e.g. the 5% and 70% thresholds, but in real-life cases, also filtering, detrending, outlier removal, and much more).

It should be clear that one thing is a MBAT *rule*, and another is a MBAT *procedure*, as in fact the process of turning the first into the second requires some decision (arbitrary, as *not stemming from the rule*) practically at each step. Correspondingly, discussing the first is merely a methodological fact, while turning it into the second requires a lot of engineering effort, where any of the mentioned decisions has a potentially significant impact on the obtained results and thus the product's applicability and success. This is probably the main reason why MBAT research is difficult, despite prescribing some property for a system with overall less than ten parameters may seem a sinecure. In fact, a major challenge is how to give the mentioned "engineering" facts a methodological dignity, so as to be capable of treating them in a formal manner, and not just like implementation-related incidentals.

This chapter will try to take such an attitude, assuming some reader's familiarity with PID tuning (although no fundamental facts will be totally omitted), with the aim of serving as a guide in a huge corpus of literature that would be impossible even to summarise here.

2.3 Tuning Methods in the Literature

This section reports an extremely synthetic overview of the MBAT history, limited to its major milestones, by describing a few methods that in the opinion of the authors provided some theoretical advance or were particularly successful in the applications. The choice of organising the section in this way was dictated by the enormous extent of the matter: entire books are devoted at listing and sometimes evaluating PID tuning methods [5, 55], and if this section were structured in the same way, it would be nothing more than a shallow rough copy of such works. On the contrary, after detailing the considered model and controller structures, the following two Sects. 2.3.3 and 2.3.4 deal with the PI and the PID controller, telling the MBAT story in extreme synthesis, from the dawn up to recent times but excluding modern research issues. Then, Sects. 2.3.5 and 2.3.6 summarise the *scenario* and propose the announced minimal taxonomy guidelines for MBAT methods, while Sect. 2.3.7 provides a few samples of what was not possible to treat herein.

2.3.1 The Typical Model Structures

As anticipated, models for MBAT need to be simple. One reason is the necessity of parameterising them based on a limited amount of data in the presence of poor excitation. A second one is the need for simple and preferably explicit tuning rules, to the advantage of a safe autotuner operation. A third reason is the opportunity of summarising the main (and hopefully control-relevant) process dynamics with few parameters, to allow for an easy interpretation of their values, also on the part of the typical plant personnel.

In this chapter, the decision was taken to omit treating unstable models. The motivation is that asymptotically stable or integrating ones cover the great majority of the cases of interest, although relevant ones (especially for example in the domain of chemical reactions) are left out. Also, discussing MBAT for unstable processes, especially for the involved robustness issues, would require a large amount of space, and it was considered preferable to disregard the matter completely instead of reporting a necessarily partial treatise.

Given the considerations above, the structures employed in virtually all the literature are the First Order Plus Dead Time (FOPDT), the Integrator Plus Dead Time (IPDT), the First Order and Integrator Plus Dead Time (FOIPDT), and the Second Order Plus Dead Time (SOPDT) in the OverDamped (OD) and UnderDamped (UD) subtypes, the latter appearing typically in mechanical systems. In fact different acronyms are frequently used instead of those reported above, but the meaning of such numerous variants in the literature should be obvious to the reader. For the purpose of this work, the mentioned model structures are thus expressed in the form of SISO transfer functions as follows.

$$\text{FOPDT} \qquad M(s) = \frac{\mu_M \, e^{-sD_M}}{1 + sT_M}, \tag{2.2a}$$

$$\text{IPDT} \qquad M(s) = \frac{\mu_M \, e^{-sD_M}}{s}, \tag{2.2b}$$

$$\text{FOIPDT} \qquad M(s) = \frac{\mu_M \, e^{-sD_M}}{s(1 + sT_M)}, \tag{2.2c}$$

$$\text{SOPDT-OD} \quad M(s) = \frac{\mu_M \, e^{-sD_M}}{(1 + sT_{M1})(1 + sT_{M2})}, \tag{2.2d}$$

$$\text{SOPDT-UD} \quad M(s) = \frac{\mu_M \, e^{-sD_M}}{(1 + 2\frac{\xi_M}{\omega_M}s + \frac{s^2}{\omega_M^2})}. \tag{2.2e}$$

2.3.2 The Considered Controller Structures

For apparent space reasons and for the purpose of the chapter, we limit here the scope to the one-degree-of-freedom PI and the PID controllers, and for uniformity, we refer to their so-called ISA forms, i.e. the error-to-control transfer functions

$$\text{PI} \qquad R(s) = K\left(1 + \frac{1}{sT_i}\right), \tag{2.3a}$$

$$\text{PID} \quad R(s) = K\left(1 + \frac{1}{sT_i} + \frac{sT_d}{1 + sT_d/N}\right), \tag{2.3b}$$

where K is the gain, T_i and T_d the integral and derivative times, respectively, and N the ratio between T_d and the time constant of the "derivative filter" in the real PID case, the ideal PID corresponding to $N = \infty$.

2.3.3 Some MBAT Methods for the PI

The first and historical tuning method for PI (and PID) controllers is by common opinion that proposed by Ziegler and Nichols in 1942 [75]. In fact, however, previous works by Callander and coworkers [7, 25] in 1935–36 proposed a similar method, thereby constituting the first known "tuning rule". The work by Callender et al. actually dates back to an internal report of ICI (Alkali) Ltd., Northwich, written in 1934—that is, eight years before the paper by Ziegler and Nichols—and was discovered by Aidan O'Dwyer in 2004. The reader interested in the fascinating story of that discovery can find it in [56]. The Callender rule for the PID is presented later on in Sect. 2.3.4.

Coming back to the Ziegler–Nichols PI rule, it refers to the FOPDT model (2.2a) and the ISA PI (2.3a) and takes the form

$$K = \frac{0.9T_M}{\mu_M D_M}, \qquad T_i = 3.33T_M. \tag{2.4}$$

The goal is to obtain a quarter decay ratio for the nominal closed loop, which makes (2.4) applicable for $D_M/T_M \leq 1$. The model is parameterised based on the application of the tangent method to an open-loop process unit step response, whence other published rules equivalent to (2.4) that refer to the tangent parameters directly. It is worth noticing that the ancestor of most MBAT methods *did* specify the model parameterisation procedure—a habit not maintained in several subsequent proposals. Finally, of course the same concept just sketched can be applied to different model structures. For example, there exist a rule analogous to (2.4) for the IPDT model (2.2b) that reads

$$K = \frac{0.9}{\mu_M D_M}, \qquad T_i = 3.33D_M, \tag{2.5}$$

and again employs a tangent-based model parameterisation procedure.

Along the idea of constraining the closed-loop damping, or somehow equivalently the maximum overshoot of the set point step response, many methods were introduced in the following years. Two notable proposals are that by Chien, Hrones and Reswick [9] and by Cohen and Coon [10]. The Chien et al. method has the merit of being (probably) the first to distinguish "servo" control problems (where the goal is to track the set point) and "regulatory" ones (where the set point is substantially considered constant, and the problem is to effectively reject load disturbances). Notice, incidentally, that the work by Ziegler and Nichols did not propose such a distinction but looked particularly at disturbance responses. In many subsequent works the statement can be found that the Ziegler and Nichols rules provide "too oscillatory a set point response": true, but tracking was not their primary goal.

The Chien rules refer to (2.2a) and (2.3a) and read

$$\text{Servo, 0\% overshoot} \qquad K = \frac{0.35 T_M}{\mu_M D_M}, \qquad T_i = 1.17 T_M, \tag{2.6a}$$

$$\text{Servo, 20\% overshoot} \qquad K = \frac{0.6 T_M}{\mu_M D_M}, \qquad T_i = T_M, \tag{2.6b}$$

$$\text{Regulatory, 0\% overshoot} \qquad K = \frac{0.6 T_M}{\mu_M D_M}, \qquad T_i = 4 D_M, \tag{2.6c}$$

$$\text{Regulatory, 20\% overshoot} \qquad K = \frac{0.7 T_M}{\mu_M D_M}, \qquad T_i = 2.33 D_M, \tag{2.6d}$$

and are applicable for $0.1 < D_M/T_M < 1$. Notice the structural similarity to the Ziegler and Nichols ancestor and also that the integral time is made dependent on the model time constant in the servo case and on the model delay in the regulatory one. With the same model, a servo problem thus results in a smaller PI gain than a regulatory one. On the contrary, the servo integral time is smaller than the regulatory one only for processes where the rational dynamics definitely dominates the delay, i.e. for D_M/T_M smaller than $1.17/4 \approx 0.29$ and $1/2.33 \approx 0.43$ in the 0% and 20% overshoot cases, respectively. Observe also that the work by Chien et al. still specifies the model parameterisation procedure (here too, the tangent method).

Also, the Cohen and Coon method refers to (2.2a) and (2.3a). It takes the form

$$K = \frac{1}{\mu_M}\left(0.083 + 0.9\frac{T_M}{D_M}\right), \qquad T_i = T_M\left(\frac{3.33\frac{D_M}{T_M} + 0.31(\frac{D_M}{T_M})^2}{1 + 2.22\frac{D_M}{T_M}}\right), \tag{2.7}$$

and is applicable for $0 < D_M/T_M \leq 1$. The goal is quarter closed-loop damping, and once again, the model parameterisation procedure is specified to be tangent-based.

Observing (2.4) through (2.7), a progressively more articulated use of the model parameters can be observed, corresponding to a deeper structuring of the possible control problems. In particular, the D_M/T_M ratio, sometimes also called the "controllability index", emerges as a quantity with particular relevance. Such an idea is probably one of the starting points for subsequent elaborations that will be formalised later on when the Internal Model Control (IMC) principle will be applied to MBAT.

In the following years a number of rules similar to those just quoted were proposed and are omitted here for brevity. In parallel, however, a stream of methods started emerging, the goal of which is to optimise some integral index referring to a convenient closed-loop response, instead of punctual quantities such as a decay ratio or an overshoot. Among the most used indices are the Integral of the Squared Error (ISE), the Integral of the Absolute Error (IAE), and the Integral of Time times the Absolute Error (ITAE), defined as

$$\text{ISE} = \int_0^\infty e(t)^2\,dt, \qquad \text{IAE} = \int_0^\infty \left|e(t)\right|dt, \qquad \text{ITAE} = \int_0^\infty t\left|e(t)\right|dt, \tag{2.8}$$

where $e(t)$ is the error (many other indices are used in the literature, but there is no space here for a discussion). Two pioneering works on the matter are those by Murrill [54] and Rovira [60]. Both refer to (2.2a) and (2.3a) and use a tangent-based parameterisation procedure. The Murrill rules aim at minimising the ISE, IAE or ITAE in the regulatory case, i.e., referring to the unit load disturbance closed-loop step response. Denoting from now on by θ_M the normalised delay expressed as D_M/T_M, those rules are valid for $0.1 \le \theta_M \le 1$, taking the form

$$\text{ISE} \qquad K = \frac{1.305}{\mu_M \theta_M^{0.959}}, \qquad T_i = \frac{T_M\,\theta_M^{0.739}}{0.492}, \tag{2.9a}$$

$$\text{IAE} \qquad K = \frac{0.984}{\mu_M \theta_M^{0.986}}, \qquad T_i = \frac{T_M\,\theta_M^{0.707}}{0.608}, \tag{2.9b}$$

$$\text{ITAE} \qquad K = \frac{0.859}{\mu_M \theta_M^{0.977}}, \qquad T_i = \frac{T_M\,\theta_M^{0.68}}{0.674}. \tag{2.9c}$$

The Rovira rules conversely minimise the IAE or the ITAE in the servo case, i.e., for the closed-loop unit step set point response. They are valid for $0.1 \le \theta_M \le 1$ and read

$$\text{IAE} \qquad K = \frac{0.758}{\mu_M \theta_M^{0.861}}, \qquad T_i = \frac{T_M}{1.02 - 0.323\theta_M}, \tag{2.10a}$$

$$\text{ITAE} \qquad K = \frac{0.586}{\mu_M \theta_M^{0.916}}, \qquad T_i = \frac{T_M}{1.03 - 0.165\theta_M}. \tag{2.10b}$$

The stream of rules aiming at optimising integral indices is still flourishing, also thanks to the availability of computational resources that one could hardly dare to dream at the Murrill and Rovira times. For the scope of this chapter, it is however more interesting to notice how the type of control problem started (further) calling for structurally different relationships between the model and the controller parameters. Quite intuitively, although up to here the history has touched almost exclusively

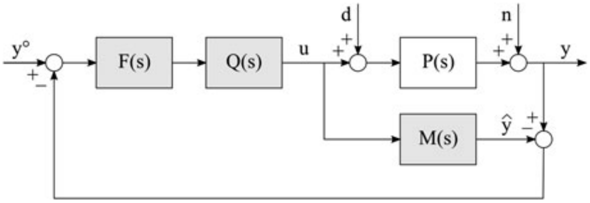


Fig. 2.2 The basic Internal Model Control (IMC) scheme: y° is the set point, y the controlled variable, u the control signal, d and n respectively a load disturbance and an output noise

FOPDT-based methods, there are extensions to other structures, basically the IPDT and the FOIPDT. There is however no space to comment the matter with sufficient detail, and the conceptual aspects that this chapter aims at discussing can emerge even with such an omission. The reader interested in a complete panorama can refer e.g. to [55].

In fact, more or less in the 1960s, the idea started emerging of tuning the controller by impressing not some characteristics of the obtained closed-loop responses, be it a punctual value or an integral index, but the form of the closed-loop dynamics in the whole, for example by requiring that the transfer function from set point (or load disturbance) to controlled variable be "as similar as possible" to some reference one. A historical MBAT rule conceived that way and still used in many applications under the name of "λ-tuning" was proposed in 1968 by Dahlin [12]. Starting from the FOPDT model (2.2a), Dahlin's idea was to make the transfer function from the set point to the controlled variable resemble a first-order one with unity gain, the same delay as the process model, and a specified time constant, which becomes the method's design parameter (sort of another innovation, notice). Denoting the said time constant with λ, whence the method's name, the idea corresponds to tuning the regulator so as to approximate the ideal (but not rational) transfer function

$$R_{\mathrm{ID}}(s) = \frac{1+sT_M}{\mu_M(1+s\lambda-e^{-sD_M})}. \tag{2.11}$$

If e^{-sD_M} is replaced by its (1,0) Padé approximation $1-sD_M$, the resulting controller is a PI, and the Dahlin rules are thus

$$K = \frac{T_M}{\mu_M(D_M+\lambda)}, \qquad T_i = T_M. \tag{2.12}$$

In fact and more in general, having the Dahlin rule as an anticipation, between the 1970s and the 1980s, time became ripe for the introduction in MBAT of the already mentioned IMC principle that originated a vast family of methods. There is quit a bit of debate on which was the first work on that matter, and since we do not intend here to enter said debate, we prefer to illustrate the IMC principle in a view to its usefulness in MBAT, then show some of the most widely used methods, and later on employ the principle for some methodological discussions on model error and robustness.

In a nutshell, the IMC idea is explained by the block diagram of Fig. 2.2 and thinking for now to the case of an asymptotically stable process $P(s)$: if the model $M(s)$ of that process is perfect and if there is neither load disturbance d nor output noise n, then apparently the process output y and the model output $\widehat{y}$ are equal, the scheme is open-loop, and the transfer function from the set point y° to the controlled variable y is $F(s)Q(s)M(s)$. If *in addition* $Q(s)$ can be taken as the inverse of $P(s)$, then the desired closed-loop dynamics from set point to controlled variable can be chosen arbitrarily by selecting $F(s)$. Finally, the IMC controller (the grey blocks in Fig. 2.2) is immediately shown to be equivalent to the feedback one given by the error-to-control transfer function

$$R(s) = \frac{F(s)Q(s)}{1 - F(s)Q(s)M(s)}. \tag{2.13}$$

Of course the above hypotheses are in general unrealistic, and if the IMC principle is to be used with a high-fidelity process model, and thereby accepting a controller the structure of which depends on that of said model, there are a number of aspects to address and a vast literature. If the IMC principle is to be applied to MBAT, however, things are somehow simpler, in that the principle has to be viewed basically as a flexible means to obtain explicit tuning rules for the model structures of interest, provided that the said structures allow (exactly if possible, and in the opposite case with reasonable approximation) to derive a controller of the desired form. For example, if one takes as $M(s)$ the FOPDT model (2.2a), as $Q(s)$ the inverse of its minimum-phase part, i.e. $(1 + sT_M)/mu_M$, and sets $F(s) = 1/(1 + s\lambda)$, the Dahlin rule is re-obtained and could be called the first IMC-PI one. The IMC principle however allows for more insight into the problem, leading to the "improved" rules mentioned here in the following and to some robustness-related considerations reported later on. On the other hand, however, as the literature began focusing on IMC-based and similar methods, attention was progressively shifted on the characteristics of the control problem involving the process *model*, and works accounting for the particular model parameterisation procedure used (or even mentioning it) ceased to be the majority.

Coming back to the mainstream history, a notable "improved IMC" rule for the PI was proposed by Rivera, Skogestad and Morari in 1986 [59] in a view to improve performance especially in the case of significant process delays, which reads

$$K = \frac{T_M + D_M/2}{\mu_M \lambda}, \qquad T_i = T_M + \frac{D_M}{2} \tag{2.14}$$

and is presented in the quoted paper together with a wealth of considerations on IMC-based tuning that is impossible to summarise here but is highly advisable for reading. Another successful IMC improvement, known as "SIMC", was proposed by Skogestad [63] with the aim (simplifying the matter a bit) of avoiding excessive values of the integral time, thus sluggish transients due to poor control activity. The SIMC rules for the PI are

$$K = \frac{T_M}{\mu_M(D_M + \lambda)}, \qquad T_i = \min\bigl(T_M, 4(D_M + \lambda)\bigr). \tag{2.15}$$

Another relevant research line, closely related to IMC-based tuning, is the so-called "Direct Synthesis" (DS), exemplified by [8], that shares with the IMC the possibility of being employed with different model structures, thereby unifying in a single design framework a matter that with older approaches needed treating on a per-structure basis.

To end this section, at least another tuning *rationale* needs mentioning, namely that based on interpolation of results obtained via numerical optimisation. A successful result of the said approach is the AMIGO PI proposed by Hägglund and Åström in 2002 [24] and is somehow borderline with respect to classical MBAT since it extends the MIGO method previously proposed by the same authors. The MIGO assumes the process transfer function to be known, while the AMIGO tunes based on three parameters (the gain, the apparent dead time and time constant, and the inflection point slope) deducible from a step response under the hypotheses that it is essentially monotonic and is therefore considered by the authors a "revisitation" of the Ziegler and Nichols rule. The AMIGO goal is to maximise the integral gain subject to a constraint on the maximum sensitivity, in an attempt to solve the performance/robustness tradeoff—a subject that has been gaining increasing interest in the last years. The method is applicable to asymptotically stable and integrating processes, see [24] for its description.

The last tuning method here mentioned, based again on interpolation, is the so-called "kappa–tau" one [23] that obtains the PI parameters as

$$K = \frac{A_0}{k_M} e^{(A_1 \tau_M + A_2 \tau_M^2)}, \qquad T_i = B_0 D_M e^{(B_1 \tau_M + B_2 \tau_M^2)}, \tag{2.16}$$

where $k_M = \mu_M \theta_M$, and $\tau_M = D_M/(D_M + T_M)$ is another definition of "normalised delay", differing from θ_M as it lies in the 0–1 range; the constants $A_{0,1,2}$ and $B_{0,1,2}$, tabulated in [23], are obtained by numerically optimising performance under a robustness constraint expressed on the magnitude margin M_s, defined as

$$M_s = \max_{\omega} \left| \frac{1}{1 + L(j\omega)} \right|, \tag{2.17}$$

where $L(s)$ is the open-loop transfer function, and for which the values of 1.4 and 2.0 provide two sets of constants A and B, respectively corresponding to conservative and aggressive tunings.

2.3.4 Some MBAT Methods for the PID

Coming to the PID controller, more or less the same story just followed for the PI can be told, with two main differences. First, given the richer controller structure, more model structures are present. Second, some methods refer to the ideal PID—i.e. to (2.3b) with $N = \infty$—and some to the real one, the first type of methods being not the totality, but a significant majority.

Here too, the ancestor is the work by Callender et al. [7]. It refers to the ideal PID and the FOPDT model (2.2a) and takes the form

$$K = \frac{1.066}{\mu_M D_M}, \qquad T_i = 1.418 D_M, \qquad T_d = a D_M, \quad a = [0.353, 0.47]. \quad (2.18)$$

Much more known, as for the PI case, is however the method by Ziegler and Nichols [75], also referring to the ideal PID and providing its three parameters as

$$K = \frac{a T_M}{\mu_M D_M}, \qquad T_i = 2 D_M, \qquad T_d = \frac{D_M}{2}, \quad a = [1.2, 2] \qquad (2.19)$$

for a FOPDT model parameterised with the tangent method, aiming at quarter decay ratio. Similar proposals can be found in the following years for other model structures. An example for the IPDT is the work by Ford [17], whose rules are

$$K = \frac{1.48}{\mu_M D_M}, \qquad T_i = 2 T_M, \qquad T_d = 0.37 D_M, \qquad (2.20)$$

with the goal of a 2.7:1 decay ratio, the model being assumed known (i.e. the parameterisation procedure not being thought of as part of the tuning). The quoted work by Chien et al. [9] provides overshoot-related rules also for the PID that we omit here for brevity.

The use of integral indices emerged a few years later, like for the PI, and still continues. An "old" example given by the Murrill [54] is minimum IAE rules for the FOPDT model in the regulatory case

$$K = \frac{1.4835}{\mu_M \theta_M^{0.981}}, \qquad T_i = \frac{T_M \theta_M^{0.749}}{0.878}, \qquad T_d = 0.482 T_M \theta_M^{1.137} \qquad (2.21)$$

with a tangent-based parameterisation procedure. A more recent example was proposed for both the servo and regulatory case and the IPDT model by Visioli [72], referring to the various indices. For example, the minimum servo ISE rules are

$$K = \frac{1.37}{\mu_M D_M}, \qquad T_i = 1.69 D:M, \qquad T_d = 0.59 D_M \qquad (2.22)$$

with the model assumed already parameterised.

The idea of tuning the controller by impressing the form of the closed-loop dynamics in the whole emerged also for the PID, and here too the work by Dahlin [12] provides historical MBAT rule for the FOPDT model. If (2.11) is rationally approximated by using not a (1,0) but a (1.1) Padé approximation of the delay term, i.e. $(1 - sD_M/2)/(1 + sD_M/2)$, the resulting controller is a PID, whence the Dahlin (or "PID λ-tuning") rules

$$K = \frac{T_M + D_M/2}{\mu_M (D_M + \lambda)}, \qquad T_i = T_M + \frac{D_M}{2}, \qquad T_d = \frac{T_M D_M/2}{T_M + D_M/2}, \qquad (2.23)$$

referring to the ideal controller.

The IMC principle came then into play, and in the PID case it was exploited more extensively then in the PI one. As before, in fact, the Dahlin rule can be considered the first (ideal) IMC-PID one, but much more has been done. For example, a wide variety of applications of the IMC principle to PID MBAT can be found in [59], where all the model structures (2.2a)–(2.2e), plus several others, are considered. For some structures, the PID is augmented with a lag term and thus turned into a real one. This is not done on a general basis, however. On the other hand, Leva and Colombo proposed a tuning rule invariantly producing a real PID [43], referring (2.2a) and (2.3b), and composed of the relationships

$$\begin{aligned} T_i &= T_M + \frac{D_M^2}{2(D_M+\lambda)}, & K &= \frac{T_i}{\mu_M(D_M+\lambda)}, \\ N &= \frac{T_i(D_M+\lambda)}{\lambda T_i} - 1, & T_d &= \frac{\lambda D_M N}{2(D_M+\lambda)}, \end{aligned} \tag{2.24}$$

to be used in sequence; the design parameter λ is interpreted as the desired closed-loop dominant time constant. The number of IMC-derived tuning rules is impressive and impossible to review, and the same applies to the similar DS principle, some applications of which to the PID can be found in the paper [8] quoted above. A major merit of both approaches, but in particular—see the comparative discussion in [8]—of the IMC, is the possibility of conducting robustness analysis in a sense compatible with MBAT: some words on that important matter will be spent later on.

Continuing the panorama, the interpolation-based approach was applied to the PID too, and there exist corresponding versions of the quoted AMIGO [24] and “kappa–tau” [23] methods, plus many others.

As can be seen, the PI and PID stories are very parallel. It is now time to discuss the announced differences. First, the two zeroes of the PID make it more suited to the PI for second- and even higher-order systems, see e.g. the discussions on “which controller is adequate” in many works such as [5]. This was recognised long ago and gave rise to many considerations on the role of unmodelled dynamics that general frameworks such as the IMC and the robust control theory, see e.g. [14, 53] as references of more or less that period, subsequently comprehended in a unitary treatise.

From this point of view, two historical methods are worth mentioning, with some considerations on their potential and pitfalls as expressed by their authors at the time the research was published. The firs method was proposed by Haalman in 1965 [20]. The method has the goal of making the nominal open-loop transfer function $L(s) = R(s)M(s)$ resemble the desired one

$$L_{\mathrm{ID}}(s) = \frac{2e^{-sD_M}}{3D_M s}, \tag{2.25}$$

which corresponds to achieving a cutoff frequency ω_c of $2D_M/3$ and a phase margin of 50° approximately. If the (non-integrating) process model is in the form (2.2a), it is advised to select a PI, while if it is in the form (2.2d), an ideal PID is chosen, and the tuning formulæ are

$$K = \frac{2(T_{M1}+T_{M2})}{3\mu_M D_M}, \qquad T_i = T_{M1}+T_{M2}, \qquad T_d = \frac{T_{M1}T_{M2}}{T_{M1}+T_{M2}}. \tag{2.26}$$

Haalman (correctly) stated, right from the paper's title, that the method is very suited for processes with overdamped response and significant delay. In fact, being ω_c inversely proportional to D_M, the requested response might become too fast if D_M is small. It is additionally worth noticing that the method does not use any model mismatch information, thus there is no way of guiding its operation on the basis of the model accuracy.

Such a problem had however started receiving attention a few years before, as testified by the second method here mentioned, namely that introduced by Kessler in 1958 [38, 39] and known as the "Symmetric Optimum" (SO). In fact, despite being older than the Haalman method, the SO one contains several ideas that have been widely developed in the following years. The most important one is to assume that the (non integrating) process model be

$$M(s) = \frac{\mu e^{-sD}}{\prod_{k=1}^{m}(1+sT_k)\prod_{h=1}^{n}(1+sT_h)}, \tag{2.27}$$

i.e. either in the form (2.2a) if $m=1$ or (2.2d) if $m=2$, but with some other poles accounting for the process dynamics not described by the model. It is furthermore assumed that the time constants T_k dominate the process dynamics, i.e. that

$$T_k \gg D + \sum_{h=1}^{n}(1+sT_h) \quad \forall k. \tag{2.28}$$

The quantity $L + \sum_{h=1}^{n}(1+sT_h)$ can then be interpreted as the time constant of a transfer function representing the unmodelled process dynamics, which is still rough but extremely foreseeing way to account for model mismatch. Denoting by T_{um} the above quantity, the SO method takes as approximate model

$$M'(s) = \frac{\mu e^{-sD}}{(1+sT_{um})\prod_{k=1}^{m}(sT_k)} \tag{2.29}$$

and designs the controller so that the nominal cutoff frequency be $1/2T_{um}$ (thus reducing the demand as the mismatch increases, i.e. automating a very wise practice) and that the nominal open-loop magnitude $|R(j\omega)M'(j\omega)|$ have a slope of –20 dB/dec in the frequency interval from $1/4mT_{um}$ to $1/T_{um}$. The SO tuning formulæ (also the PI ones for completeness) are given in Table 2.1.

The SO method was immediately recognised to have both strengths and weaknesses. It performs very well, provided that the process delay is small since the time constants T_k must dominate also the delay—see (2.28)—thus it is especially suited for electromechanical systems, despite being keen to generate low-frequency

Table 2.1 The SO tuning formulæ

	K	T_i	T_d
PI	$\dfrac{T_1}{2\mu T_{um}}$	$4T_{um}$	
PID ($T_2 \geq 8T_{um}$)	$\dfrac{T_1 T_2}{8\mu T_{um}^2}$	$16T_{um}$	$4T_{um}$
PID ($T_2 \geq 8T_{um}$)	$\dfrac{T_1(T_2+4T_{um})}{8\mu T_{um}^2}$	T_2+4T_{um}	$\dfrac{4T_2 T_{um}}{T_2+4T_{um}}$

regulator zeroes, i.e. overshoots in the set point responses. In any case the idea of "tuning also on the basis of unmodelled dynamics" is very powerful. Apart from the methodological consequences discussed late on, for example, the SO method has led to several evolutions, maybe the most successful being a similar one called BO or "Betrags Optimum", often translated as "magnitude optimum" [68]. The application of the PID to higher-order model structures has also received attention in general, as shown e.g. by [34].

The second peculiarity of the PID story is the presence or absence of the "derivative filter", the term $1/(1+sT_d/N)$ in (2.3b), in the tuning formulæ. Curiously enough, such a relevant issue was not brought to a systematic attention, at least to the best of the authors' knowledge, until quite recent years with respect to the overall MBAT history, see e.g. [51]. As noticed in subsequent more extensive works such as [35], the derivative filter effect is invariantly a modification of the high-frequency PID behaviour, which improves noise insensitivity and robustness versus model errors acting "slightly above" the cutoff frequency, at a generally affordable cost in terms of stability degree (e.g. phase margin). However, if this is the sole effect in the case of "series" PID realisations, where the filter is cascaded to *all* the controller dynamics, the same is not true for parallel ones like the ISA (2.3b), where the derivative filter also causes the zeroes of the real PID to not coincide with those of the ideal form anymore. Hence, in the PID form probably most widely used in the applications for a number of reasons (e.g. the ease of switching on and off the individual control actions) that it is impossible to discuss here, default values for the derivative filter parameter N, quoting from the abstract of [35], "are much less natural" and can sometimes cause undesired behaviours that are definitely hard to understand for the typical personnel using those controllers.

2.3.5 Summarising the Story

Apart from the recent research issues discussed in Sect. 2.5, the story told so far can be assumed to represent quite well the panorama of assessed and industrially used MBAT methods. In the said story, also in view to better relate modern issues to previously emerging problems, three periods can be broadly distinguished.

At the dawn of MBAT, methods tended (and quite intuitively *needed*) to be extremely simple, owing both to the limited computational resources available and to the fact that much control-theoretical subjects nowadays familiar were at the time being studied, and even if some methodological results were available, their industrial application was far to come—think of optimal and robust control, just to give a sample. In that period specifications were typically expressed as punctual quantities (damping, overshoot, and so on), or it was required to minimise some integral index or cost function, rules were almost invariantly structure-specific, but (probably owing at least partially to a research attitude privileging application-related aspects with respect to formal analysis) the model parameterisation procedure was often treated as an integral part of the overall MBAT method. Industrial realisations appear.

In the subsequent period, a transition can be observed towards specifications expressed, broadly speaking, in the form of some reference model (sometimes mentioned explicitly, like in works denoted by the "model matching" or similar concepts like [1]). The idea of accounting somehow for the limited capability of the model to represent the process emerges with increasing strengths, and a visible convergence starts with methodologies (such as the mentioned robust control) that have reached a sufficient maturity. At the same time, however, the focus tends to shift away from the model parameterisation phase, concentrating instead on the analysis of the nominal control problem, i.e. that containing the regulator ant the model used for its tuning. Industrial realisations become more frequent, especially integrating MBAT in existing controllers. The end of this period may be considered to coincide with the introduction of DS and especially IMC-based techniques.

More recently, the DS and IMC analysis capabilities have led to revisit older methods in view to use more expressive specifications (e.g. gain or phase margins), and the increasing availability of computing power has permitted the use of interpolation. More attention to robustness is paid, and some discussions start emerging on what is to be meant by "robustness" in the MBAT context.

As a result of such an evolution, many MBAT industrial products exist, while however many questions are still open for modern research, as will be discussed in the following.

2.3.6 Guidelines for a Minimal Taxonomy

Based on the discussion above, a minimal taxonomy is easily provided that classifies the methods, and the following four axes can be proposed.

The first one is the type of regulator addressed (PI, ideal or real PID, and so on): such information allows us to initially relate the method to preferred classes of problems. To do so, one can use the considerations on controller selection given e.g. in [5] and/or more implementation-related knowledge, e.g. the poor suitability of an ideal PID to highly noisy measurements, and so forth.

The second axis is the type of model used, bearing in mind (the consequences of this will emerge later on) that the structure of the said model de facto dictates that of the regulator or, taking an alternative viewpoint, is dictated by it, making in any case the model/regulator structure choice a unitary fact. Again, the capability of the used model to represent some particular dynamics (e.g. the FOPDT one is poorly suited to oscillatory processes) helps relate the method to classes of problems from another point of view.

The third axis is the way specifications are accepted, and basically one can distinguish characteristics of some closed-loop response, parameters of some reference closed-loop model, and the objective of minimising some integral index (e.g. the ISE). This aspect of the proposed taxonomy allows one to relate the *rationale* of a method—a third point of view—to the types of control problems for which it is best suited (for example, set point tracking versus disturbance rejection).

The fourth axis is the mechanism behind the tuning rules, that can be broadly classified into analytical synthesis, optimisation, and possibly interpolation (and also soft computing methodologies, although space limitations oblige to omit them here). Such information helps us estimate the computational weight and the type of information needed to use the method, i.e., helps us judge on its suitability for the particular application at hand.

One could also think of a fifth axis, distinguishing whether or not (and in the affirmative case in what sense) the model parameterisation phase is considered a part of the overall tuning procedure, but such a subject has quite recently appeared in the literature, see e.g. some considerations in [30], although first noticed practically at the origins of MBAT, and is thereby deferred to the following of this chapter.

It is finally worth noticing that other taxonomies were attempted in the literature, e.g. [55], as well as a number of works evaluating and comparing (thus implicitly classifying or helping classify) PI/PID tuning rules in several ways, see e.g. [3, 15, 16, 66] and many others. The authors do not believe that the one proposed herein is in any sense superior but find this way of thinking to be slightly better as a means to select the "best" method to use in a certain type of control problems. It has of course to be acknowledged that subjective opinions play a relevant role in statements like that just proposed.

2.3.7 A Few Samples of What Was Left out Here

As shown e.g. in the introductory sections of [55], there is a full population of PID controller forms: series, parallel, interacting, noninteracting and so forth. Although here the ISA form was adopted for brevity, some rules were conceived and designed having a certain PID form in mind. Apart from the need of converting parameters from one form to another, which is totally straightforward, the *operation* of an MBAT procedure can sometimes be affected by the way the PID is written, especially for what concerns the commutation between the "normal operation" and the "tuning" modes. The matter has potentially relevant implications, in that if not accounted properly in the controller engineering, it can cause undesired and hard to explain malfunctions but is of scarce conceptual interest, thus it was omitted (suffice to include here a *caveat* not to underestimate such facts, directed to anybody possibly wishing to implement an autotuner).

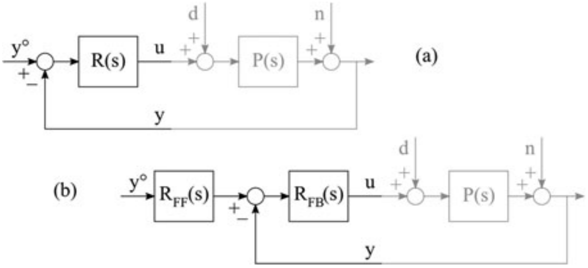


Fig. 2.3 Schemes with a 1-dof (**a**) and a 2-dof (**b**) controller

Also, there are numerous tuning methods (also of the MBAT type) that employ soft computing techniques such as neural networks, fuzzy logic, genetic algorithms and so on. On an average basis such techniques are less used in the applications than in more classical ones, and in any case too impossible to analyse in detail in a chapter like this. That matter was thus omitted.

Abandoning soft computing, it is then to be noticed that a number of other techniques were necessarily left out such as pole placement, prescription of gain and phase margins, and much more. The main reason is that all of them fall into the defined categories and would just have lengthened the list, but the choices made here must in no sense be considered a "ranking" of techniques, and, as anticipated, are largely subjective.

A final word is worth spending on two-degree-of-freedom (2-dof) controller tuning methods, corresponding to the scheme of Fig. 2.3 and that were omitted only for space reasons.

The idea of exploiting the 2-dof nature of many industrial PIDs, also frequently called the "set point weighing" functionality, is however very interesting, as it allows us to focus the tuning of the feedback path on stability and disturbance rejection, subsequently employing the feedforward path to recover and/or improve set point tracking. Also, if the 2-dof ISA form

$$U = K\left(bY^\circ - Y + \frac{1}{sT_i}\left(Y^\circ - Y\right) + \frac{sT_d}{1+sT_d/N}\left(cY^\circ - Y\right)\right) \tag{2.30}$$

is adopted, where b and c are the set point weights in the proportional and derivative actions, the feedback and feedforward blocks denoted in Fig. 2.3 as R_{FB} and R_{FF} become respectively a standard 1-dof ISA PID in the form (2.3b)—or which standard rules are readily applied—and a unity-gain set point pre-filter, namely

$$R_{\mathrm{FF}}(s) = \frac{1 + s(bT_i + T_d/N) + s^2 T_i T_d (c + b/N)}{1 + s(T_i + T_d/N) + s^2 T_i T_d (1 + 1/N)}. \tag{2.31}$$

This allows us to set up two-step MBAT procedures, first tuning R_{FB} for stability, robustness and disturbance rejection, and then R_{FF} for tracking. In other words, using MBAT in this way is an effective means to counteract the "overemphasis on the set point response" that is observed in very engineering-oriented works like [61].

2.3.8 A Final Remark

The reader may have noticed that the review of methods shown so far practically stops some years ago. This is however consistent with the idea of separating “well assessed” and “innovative” methods. From a similar point of view, it is extremely interesting to look at the great classification effort published in 2005 and 2006 by Ang, Chong and Li [4, 47]. The quoted papers list the main products for PI/PID tuning, in the form of both modules for control environments and independent software packages, together with the patents filed on PID tuning in the United States, Japan, Korea, and by the World Intellectual Property Organisation; also, there is a list of autotuning PID hardware modules.

From so extensive a survey, three facts are worth noticing. First, the release years of the mentioned hardware modules span the range 1985–2001 [4, Table V]. Second, out of 42 software packages listed, 32 support MBAT [*ibid.*, Table IV]. Third, in patents from 1971 to 2000 one can observe an increasing percentage of methods based on “non-excitation” (i.e. that do not deliberately stimulate the system); also in patents, the most used tuning approach is “formula” (i.e. simplifying a bit explicit tuning relationships) followed by “rule” (more complex mappings from process information to controller parameters, from example heuristic relationships up to fuzzy logic) and with an increasing percentage of “optimisation” [*ibid.*, Figs. 6 and 7].

If one assumes that (a) hardware objects tend to incarnate well-established technology, and only a significant methodological advance results in a new hardware generation, (b) independent software packages are typically conceived so as to deliver advanced functionalities, not convenient and/or impractical to realise in hardware with sufficient flexibility and ease of use, and (c) patents are clearly filed before the contained claims are industrially exploited, the following conclusions—in the opinion of the authors—can be drawn.

- MBAT is felt as a promising technology, but its use is mostly limited to “quite advanced” tools, typically requiring some user interaction—thus competence—especially to drive the identification phase, see e.g. the LOOP-PRO® product suite[1] [11].
- Many hardware modules do encompass MBAT, typically in the form of very simple methods, and quite often complement it with tuning maps or rule-based synthesis. At the same time, however, a significant amount of industrial products (especially, but not only, low-end ones) stick to alternative approaches like relay-based tuning. A probable reason for that is the practical absence, in the relay-based context, of any arbitrariness and/or ambiguity on how the process information—frequency response points—is obtained and treated [74].
- Much MBAT industrial research is underway on the model identification front, aiming especially at reducing the process upset and the tuning time. However, see also the item above, the said research has not (yet) fully unleashed its potential.

[1] Loop-Pro is a registered trademark of Control Station, Inc., One Technology Drive, Tolland, CT 06084.

- Modern computational capabilities make it affordable to include optimisation in the loop controllers directly—another recently started research field.

Projecting the above remark onto the scope of this chapter, it is therefore not surprising that among current research lines, particular effort is devoted to the possible effects of a model selection and/or parameterisation that is in some sense "improper" for the used tuning rule in the addressed problem. This motivates the choice of the topics discussed in Sect. 2.5.

2.4 Tuning Assessment, Monitoring and Retuning

Before treating such modern issues, however, three aspects of (MBAT) autotuners' operation, the latter two deeply intertwined, need discussing.

The first aspect is how the obtained tuning can be assessed before making it effective. Given the simple models used, it is not complicated to check the tuning in nominal conditions. The real question is whether or not the desired properties (think for example to some stability and performance level) carry over from the nominal system—that containing the model—to the real one. This clearly leads to consider unmodelled dynamics, and apart from the mentioned SO historical example, the natural theory to be brought in is that of robust control.

When declining the idea of robustness in MBAT, however, an important distinction is necessary. To set up a robustness problem, one has to specify which property has to be (made) robust, with respect to the variation of what, in which set. Assuming that the property is specified (e.g. stability) and the varying object is the process transfer function, in MBAT the set is not available by definition. In real-life cases, one makes just a single experiment, and any model error estimate (the matter will be discussed soon) can at most measure the model's inability to explain the data. No information can be obtained on the effects of a process variation, because in a single experiment of acceptable duration, there can be no such variation. In MBAT, therefore, the problem of "robustness" needs splitting into two. One is to guarantee that the PID tuned on the model will control "well enough" the real process as it was when data was collected, and this can be tackled by using model error information. The other is to quantify the "amount of model error" that can be tolerated while still preserving the property, and this can only be done a priori. It is important to bear such a distinction in mind, although the literature (no criticism intended) is quite often silent on the matter.

Model error estimates can only be obtained by analysing the identified model response in conjunction with the identification data. In order to simplify a potentially complex matter, attempts were made to treat the model error as a parametric error for the model, or more generally, to assume some structure for the model error itself, deduced by some reference response, see e.g. [50]. However, for a rigorous treatise, the model error description must be nonparametric, as shown in [42] together with a procedure to obtain a magnitude overbound for the additive error in the frequency domain. Based on nonparametric overbounds, works like [43], referring to the IMC

PID method, indicate how to guarantee that at least the real process as it was at the time of the experiment will be well controlled. On a similar front, the recent paper [70] provides a very neat problem formulation, based on two tuning parameters for performance (specifically, closed-loop time constant) and robustness, also clarifying how complex the underlying optimisation problem can be and how to obtain a simple tuning rule [*ibid.*, p. 66] serving the purpose.

Coming to the a priori quantification of the tolerable model error, in the past years this was practically the only problem addressed, and the typical tools were the classical stability degree indicators, namely the gain and phase margins, see e.g. [18, 27–29, 73]. More recently, a shift can be observed towards better indicators like the peak (nominal) sensitivity M_s, defined as

$$M_s = \max_{\omega} \left| S(j\omega) \right|, \quad S(j\omega) = \frac{1}{1 + R(j\omega)M(j\omega)}, \tag{2.32}$$

and the performance/robustness tradeoff is managed both by evaluating existing rules, as in [71], and proposing a new one with that specific purpose. Incidentally, accounting for robustness poses an interesting identification-related question: does the model have reproduce the identification data as closely as possible, thus minimising the error in the frequency domain on an average basis, or concentrate its fidelity on some band, for example to loosen the acceptable error bound in the vicinity of the cutoff? Some words on the matter will be spent later on when discussing the so-called “contextual tuning”.

The second aspect touched here is how to monitor the tuning on-line, which is strongly connected to the third one, i.e. how to decide when a re-tuning is advisable. The subject has significant connection with fault detection, as recognised since many years [36], but in the MBAT context it is more frequent, for implementation convenience, to encounter “detectors” for specific problems. Two notable examples, covering by the way the most relevant facts to detect, are the methods proposed by Hägglund [21] to detect oscillations by studying the magnitude of the IAE between subsequent zero crossings of the error, and in [22] to spot sluggish loops by the so called “idle index” that (roughly) indicates that the controlled and control variables’ derivatives have the same sign for too long during a step-generated transients. Another interesting work is [64] that uses normalised (dimensionless) settling time and IAE to detect sluggish or non-optimal loop behaviours, referring to the IMC rules, while from a more methodological standpoint, [32] attempts to cast the matter into a comparative framework with minimum variance control. Recent evolutions of the loop monitoring idea, including suggestions for controller retuning, can be found e.g. in [69].

2.5 Modern Research Issues

Despite various decades of past research, in the MBAT context many questions still stand open. In the impossibility of even just mentioning all of them, the choice is made here to discuss those that the authors feel as more relevant, admitting in advance that the choice is somehow influenced by their research interests.

Specifically, whilst up to some years ago MBAT was basically viewed as a handy methodology to devise tuning recipes, modern research is progressively concentrating on the high-level information conveyed by the presence of a model. Stretching the lexicon a bit, one could say that an MBAT-based autotuner is "more conscious" of the controlled process, in the same way as possessing a model of some object implies deeper a conscience of that object than just knowing how it responds to tome stimulus. This "consciousness" is being nowadays devoted a great effort, and the fronts on which the said effort is exerted are (in the opinion of the authors) the major actual research lines.

In the first place, right from the beginning MBAT rules were conceived for some tuning objective (quarter damping, 0% overshoot and so on). As anticipated above when proposing taxonomic guidelines, the type of specifications accepted is strongly connected to the tuning objectives, thereby helping classify the method. An interesting issue is to conversely classify the *control problem* and then act accordingly for the regulator tuning. A proposal in this sense can be found in [41], where first some quantities are defined that characterise the nominal control problem, i.e., the (model PID) couple as emerging from the chosen MBAT method. The suggested quantities, apart form the nominal cutoff frequency ω_c and phase margin φ_m, are the PID high-frequency gain, the ratio between the integral time and the model dominant time constant, the PID frequency response magnitude at ω_c, the maximum PID phase lead, the ratio between the frequency corresponding to the said maximum lead and ω_c, and finally the ratio between the closed-loop settling time, roughly estimated as $5/\omega_c$, and that of the model (assumed asymptotically stable) in open loop. Based on those quantities and a set of weights characterising the tuning desires (e.g. fast response versus high damping) and the qualitative amount of noise, a decision mechanism allows to choose, within a pre-specified set of MBAT rules, the one that best suits the tuning desires in the case at hand. Further discussions on the idea of automatically selecting a tuning rule, and correspondingly of quantitatively characterising a tuning problem, can be found in [44]. Most likely, endowing industrial autotuners with the capability of selecting the "best" rule to use will enhance their success.

On a similar front, research is addressing "multi-objective" tuning methods as a way to manage the typical MBAT tradeoffs, namely (to quote the most relevant) that opposing "servo" to "regulatory" tuning, i.e. set point tracking to disturbance rejection, and performance to robustness. As noticed since some years, see e.g. [19], this problem has quite a lot to do with optimisation, since a natural way of trading two conflicting figures of merit versus one another is to optimise one of the two subject to a constraint on the latter. The arising optimisation problems are however difficult to formalise, since many of the involved entities are more keen to a qualitative than a quantitative descriptions. Attempts were thus made to use soft computing techniques: for example, [40] proposes a method based on clonal selection accounting for diversity, distributed computation, adaptation and self-monitoring function and contains interesting comparisons to similar proposals. Alternatively, the quoted paper [19] introduces an LMI-based optimisation framework, while [67] numerically maximises “the shortest distance from the Nyquist curve of the open-loop transfer function to the critical point –1” and thus optimises the peak sensitivity as per (2.32).

Avoiding a lengthy list of references, in the particular context of multi-objective tuning a significant presence of soft computing techniques can be observed, with particular reference to genetic algorithms, and it can also be noticed that several works directly refer to a specific application, see e.g. [57]. More classical techniques, at least in the first years in which the subject received interest, tend conversely to focus on the mentioned LMI idea, since it was shown a few years before that well-assessed formalisms, such as the $\mathcal{H}_\infty$ or the μ-synthesis ones, give rise to intractable problems when the controller structure has to be constrained into the PID form [37].

In parallel, however, works such as the book [13] or [62] had systematically exploited, among others, the concept of “stabilising regions”. Given a (nominal) model, the idea is to analytically determine the region of the PID parameter space that guarantees stability, thus providing the search space for possible subsequent optimisations. Similar ideas can be found in [26], where the problem of designing for robust performance is cast into the simultaneous stabilisation of a family of complex polynomials.

More recently, a quest for simpler approaches to the multi-objective tuning problem can be observed, somehow as a consequence of the research path just sketched, together with some *renaissance*, in this new and better formalised context, of old tuning indices. A very interesting example is the paper [49] that addresses different model structures by minimising the IAE for step an load responses (which is well aligned with traditional research) in view however to manage “robustness, performance and control effort” in a coordinated manner; the interested reader can refer in particular to Sect. 4 of the paper.

In this scenario, modern trends are finally well exemplified by the mentioned works [70, 71], In the first one, an ISA PID MBAT rule is presented in which two design parameters govern respectively the closed-loop time constant and the degree of robustness, i.e. of acceptable error, and default values for the said parameters are also devised—an important matter to enhance simplicity and thus industrial acceptability. The second paper analyses how widely used MBAT rules comply with their claimed robustness specification, indicating that for some set of plants, the said specifications are hardly or not attained, while for others, there is even too wide a margin. The outcome, and certainly a direction to research upon, is that not only robustness specifications need including in MBAT (in the sense just discussed, and somehow envisaged in previous works such as [42, 43]) but checked a posteriori.

Indeed, all the above considerations could be collectively grouped in what appears to be a re-consideration of the role of the tuning model in MBAT. As pointed out e.g. in [45], when a tuning parameter is related to robustness, its choice cannot be done on the sole basis of the nominal model—quite intuitive but curiously e.g. in the FOPDT-based IMC context there are several proposals to select λ based on the model parameters. Along the same reasoning, it was also shown that the said “default” parameter choices normally tend to be excessively conservative, to the detriment of performance—an idea that subsequent and more complete investigations such as the quoted paper [71] did confirm.

In synthesis, the idea has been emerging that one relevant MBAT issue is the necessity for the model to be precise especially near the cutoff, which however is a *result* of the tuning, not a prior information. From a different but analogous perspective, in fact, the model should be evaluated based on the tuning results it produces rather than on the capability of reproducing the identification data. As shown in the quoted work, a major MBAT problem is that the identification phase contains numerous decisions that are substantially arbitrary: the model structure, the way the process is possibly stimulated and data is obtained, and the parameterisation procedure. Since relay feedback was introduced, it become clear that the said procedure produces local information—typically, one or a few Nyquist curve points—but is practically free of such arbitrary choices. As such, attempts were made to join relay-based identification and MBAT, an interesting and somehow pioneering paper being [31]. Such research is still ongoing, and a recent result is the so-called "contextual tuning"approach proposed in [46] as a more systematic treatise. The quoted paper shows that, in principle, any MBAT rule can be used by fitting the model to be exact at a certain frequency, determined by suitably driving a relay experiment. That frequency will become the cutoff one, and if the selected MBAT rule is used in such a way to nominally produce exactly that cutoff frequency, the result is the simultaneous parameterisation of the PID and the tuning model. The contextual approach has proven to produce tendentiously better degrees of robustness with respect to those of the same MBAT rule used with non-contextual parameterisation methods such as the method of areas and appears to be a promising research subject.

Another relevant topic, particularly in recent years, is that of "fragility", see e.g. [2]. Quoting from that reference, "if robustness of the control loop indicates the margin of variation in which the plant characteristics with a fixed controller may vary, the controller fragility has a similar meaning in terms of the variation of its own parameters", based on previous research reported e.g. in [13], the maximum ℓ_2 norm of the PID parameter vector can be suggested as a measure of "controller parametric stability", and from that, [2] proposes an *absolute fragility index* and shows that, besides robustness as traditionally meant, also fragility is a quantity that may vary significantly among different MBAT rules when applied to similar cases. While the mainstream research on fragility concentrates on the non-nominalities that can be introduced by imperfections in the controller components if analogue, or numerical errors in its realisation if digital, the obtained results can *de facto* be considered applicable whatever the cause is for the PID having "slightly different parameters than it should", owing e.g. to some measured outliers adversely affecting the model parameterisation.

Concerning again the model identification phase, a relevant topic is the investigation of automatic structure selection. Given the poor excitation typical of MBAT, classical techniques based on the prediction error whiteness, and also most methods coming from the identification for control domain, are in fact difficult to apply and often inadequate. A promising idea is to detect the "correct" model structure based on the recognition of some patterns in reference response, for example, an oscillating step response calls (in the continuous time domain) for a couple of complex

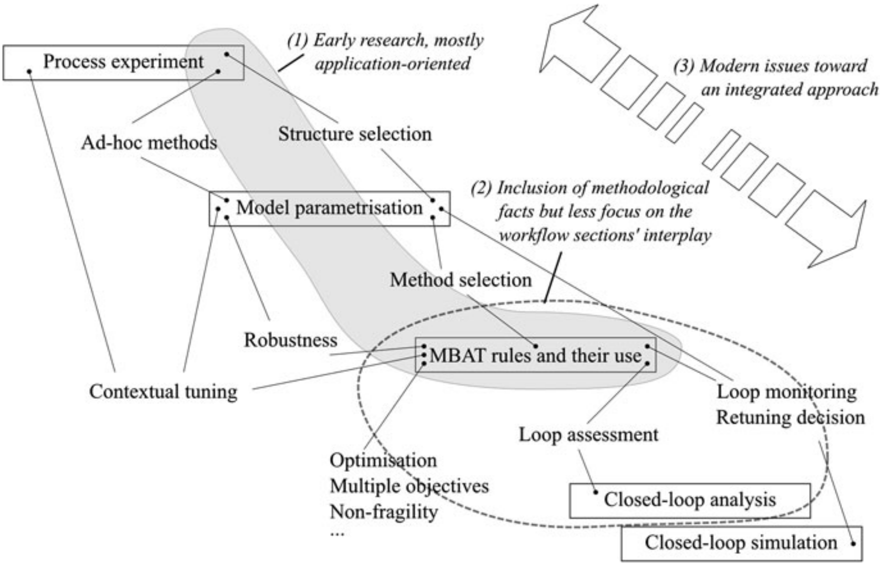


Fig. 2.4 Thel MBAT workflow revisited in the light of research developments

poles. A way to accomplish the task can be the use of neural networks coupled to suitable data pre-processing, as in [58]. Incidentally, the "structural information" provided by such techniques is potentially richer than that coming from more classical ones: sticking to the example just sketched, in fact, not only can one say that a second-order model is advised, but also that its poles should be complex—another research issue to address.

Finally, it is worth noticing that modern computation tools allow for MBAT rules that would have been impossible to realise only a few years ago. In fact, many literature proposals are de facto based on the interpolation of results obtained with some optimisation mechanism that at present can be directly plugged into the process hardware. Needless to say, this gives rise to very interesting perspectives, although from more an engineering than a strictly methodological point of view.

To end this section, in Fig. 2.4 a scheme is proposed that shows again the typical MBAT phases, like Fig. 2.1 did very synthetically at the beginning of this chapter, but also outlines the main open problems and their collocation in the overall workflow, together with mapping onto it the main research phases outlined before.

2.6 Conclusions and Future Perspectives

After presenting a necessarily brief and partial review of MBAT, it is now the time to draw some conclusions that can be summarised in the following items and implicitly express the authors' opinions on future perspectives.

- MBAT is a powerful tool because the presence of a process model allows one to tell more on the closed loop than it is possible with other tuning approaches.
- However, the model must be suitable to do so, or the conclusions drawn on the forecast loop behaviour, even if a more or less satisfactory tuning was actually achieved, may be heavily incorrect. In one word, the model can exert a great power but bears an equally great responsibility
- Better still, the *model identification* and the *tuning assessment* bear such responsibility. MBAT poses very peculiar issues as for both, and there is still room for a lot of research.
- From the methodological standpoint, beside improvements in the various phases of the MBAT process, an *integrated* approach said process is necessary, nowadays envisaged quite clearly and pursued by modern research lines.
- From the application-related standpoint, MBAT has not yet unleashed all of its potential. Most likely, one reason for that is some lack still present in the aforementioned integrated view of the process. Indeed, the interest for MBAT research definitely comes also from engineering issues.
- On the same front, the impressing increase in the computational power available "on the plant floor" will allow one to realise solutions that only some years ago were just wishful thinking, and possibly also to port the said solutions directly into the loop controllers—i.e., not only in centralised software packages used in the control room. There will be much to think and design also in the user interface and ergonomy of the so envisaged product, since a wide usage can be foreseen, also on the part of non-specialist personnel.

To conclude, the authors believe that, for sure, MBAT has some pitfalls, but within the various possible approaches to (PID) autotuning, it is probably the one that—while being researched upon—led to the deepest insight into the (auto)tuning problem and also into the way a tuning procedure has to be engineered. The authors' hope is that the few and partial considerations reported herein may be helpful for the community who devote their effort to such a fascinating research and engineering subject.

References

1. Aguirre, L.A.: PID tuning based on model matching. Electron. Lett. **28**(25), 2269–2271 (1992)
2. Alfaro, V.M.: PID controllers' fragility. ISA Trans. **46**(4), 555–559 (2007)
3. Alvarez-Ramirez, J., Monroy-Loperena, R., Velasco, A., Urrea, R.: Optimality of internal model control tuning rules. Ind. Eng. Chem. Res. **43**(24), 7951–7958 (2004)
4. Ang, K.H., Chong, G.C.Y., Li, Y.: PID control system analysis, design, and technology. IEEE Trans. Control Syst. Technol. **13**(4), 559–576 (2004)
5. Åström, K.J., Hägglund, T.: Advanced PID Control. Instrument Society of America, Research Triangle Park (2006)
6. Åström, K.J., Hägglund, T., Hang, C.C., Ho, W.K.: Automatic tuning and adaptation for PID controllers—a survey. Control Eng. Pract. **1**(4), 699–714 (1993)
7. Callander, A., Hartree, D.R., Porter, A.: Time lag in a control system. Philos. Trans. R. Soc. Lond. **235**, 415–444 (1935)

8. Chen, D., Seborg, D.E.: PI/PID controller design based on direct synthesis and disturbance rejection. Ind. Eng. Chem. Process Des. Dev. **41**(19), 4807–4822 (2002)
9. Chien, K.L., Hrones, J.A., Reswick, J.B.: On the automatic control of generalised passive systems. Trans. Am. Soc. Mech. Eng. **74**(2), 175–185 (1952)
10. Cohen, G.H., Coon, G.A.: Theoretical consideration of retarded control. Trans. Am. Soc. Mech. Eng. **75**(6), 827–834 (1953)
11. Cooper, D.: Practical Process Control Using Control Station. Control Station, Inc., Storrs (2004)
12. Dahlin, E.G.: Designing and tuning digital controllers. Instrum. Control Syst. **41**(6), 77–81 (1968)
13. Datta, A., Ho, M., Bhattacharyya, S.P.: Structure and Synthesis of PID Controllers. Springer, Berlin (2000)
14. Doyle, J.C., Francis, B.A., Tannenbaum, A.R.: Feedback Control Theory. MacMillan, Basingstoke (1992)
15. Foley, M.W., Julien, R.H., Copeland, B.R.: A comparison of PID controller tuning methods. Can. J. Chem. Eng. **83**, 712–722 (2005)
16. Foley, M.W., Ramharack, N.R., Copeland, B.R.: Comparison of PI controller tuning methods. Ind. Eng. Chem. Res. **44**(17), 6741–6750 (2005)
17. Ford, R.L.: The determination of the optimum process-controller settings and their confirmation by means of an electronic simulator. IEE Proc., Part 2 **101**(80), 141–155 and 173–177 (1953)
18. Fung, H.W., Wang, Q.G., Lee, T.H.: PI tunig in terms of gain and phase margins. Automatica **34**(9), 1145–1149 (1998)
19. Ge, M., Chiu, M.S., Wang, Q.G.: Robust PID controller design via LMI approach. J. Process Control **12**(1), 3–13 (2002)
20. Haalman, A.: Adjusting controllers for a deadtime process. Control Eng. **65**, 71–73 (1965)
21. Hägglund, T.: A control-loop performance monitor. Control Eng. Pract. **3**(11), 1543–1551 (1995)
22. Hägglund, T.: Automatic detection of sluggish control loops. Control Eng. Pract. **7**(12), 1505–1511 (1999)
23. Hägglund, T., Åström, K.J.: Automatic Tuning of PID Controllers. In: Levine, S. (ed.) The Control Handbook, pp. 817–826. CRC Press, Boca Raton (1996)
24. Hägglund, T., Åström, K.J.: Revisiting the Ziegler–Nichols rules for PI control. Asian J. Control **4**(4), 364–380 (2002)
25. Hartree, D.R., Porter, A., Callander, A., Stevenson, A.B.: Time lag in a control system—II. Proc. R. Soc. Lond. Ser. A **161**, 460–476 (1937)
26. Ho, M.T., Lin, C.Y.: PID controller design for robust performance. IEEE Trans. Autom. Control **48**(8), 1404–1409 (2003)
27. Ho, W.K., Gan, O.P., Tay, E.B., Ang, E.L.: Performance and gain and phase margins of well-known PID tuning formulas. IEEE Trans. Control Syst. Technol. **4**(4), 473–477 (1996)
28. Ho, W.K., Hang, C.C., Cao, L.S.: Tuning of PID controllers based on gain and phase margin specifications. Automatica **31**(3), 497–502 (1995)
29. Ho, W.K., Hang, C.C., Zhou, J.: Self-tuning PID control of a plant with under-damped response with specifications on gain and phase margins. IEEE Trans. Control Syst. Technol. **5**(4), 446–452 (1997)
30. Huang, H.P., Chen, C.L., Lai, C.W., Wang, G.B.: Autotuning for model-based PID controllers. AIChE J. **42**(9), 2687–2691 (1996)
31. Huang, H.P., Jeng, J.C., Luo, K.Y.: Auto-tune system using single-run relay feedback test and model-based controller design. J. Process Control **15**(6), 713–727 (2005)
32. Hugo, A.J.: Performance assessment of single-loop industrial controllers. J. Process Control **16**(8), 785–794 (2006)
33. Gyöngy, I.J., Clarke, D.W.: On the automatic tuning and adaptation of PID controllers. Control Eng. Pract. **14**(2), 149–163 (2006)
34. Isaksson, A.J., Graebe, S.F.: Analytical PID parameter expression for higher order systems. Automatica **35**(6), 1121–1130 (1999)

35. Isaksson, A.J., Graebe, S.F.: Derivative filter is an integral part of PID design. IEE Proc., Control Theory Appl. **149**(1), 41–45 (2002)
36. Isermann, R.: Process fault detection based on modeling and estimation methods—a survey. Automatica **20**(4), 387–404 (1984)
37. Iwasaki, T., Skelton, R.E.: All fix-order $\mathcal{H}_\infty$ controllers: observer-based structure and covariance bounds. IEEE Trans. Autom. Control **40**(3), 512–516 (1995)
38. Kessler, C.: Das Symmetrische Optimum, Teil I. Regelungstechnik **6**(11), 395–400 (1958) (in German)
39. Kessler, C.: Das Symmetrische Optimum, Teil II. Regelungstechnik **6**(12), 432–436 (1958) (in German)
40. Kim, D.H., Cho, J.H.: Robust PID controller tuning using multiobjective optimization based on clonal selection of immune algorithm. Knowl.-Based Intell. Inf. Eng. Syst. **3213**, 50–56 (2004)
41. Leva, A.: Automated selection of model-based PID tuning methods based on the nominal control problem. In: Proc. ECC2007, Kos, Greece, pp. 5341–5347 (2007)
42. Leva, A., Colombo, A.M.: Estimating model mismatch overbounds for the robust autotuning of industrial regulators. Automatica **36**(12), 1855–1861 (2000)
43. Leva, A., Colombo, A.M.: On the IMC-based synthesis of the feedback block of ISA-PID regulators. Trans. Inst. Meas. Control **26**(5), 417–440 (2004)
44. Leva, A., Donida, F.: Quality indices for the autotuning of industrial regulators. IET Control Theory Appl. **3**(2), 170–180 (2009)
45. Leva, A., Schiavo, F.: On the role of the process model in model-based autotuning. In: Proc. 16th IFAC World Congress, Praha, Czech Republic (2005)
46. Leva, A., Negro, S., Papadopoulos, A.V.: PI/PID autotuning with contextual model parametrisation. J. Process Control **20**(4), 452–463 (2010)
47. Li, Y., Ang, K.H., Chong, C.Y.: Patents, software, and hardware for PID control—an overview and analysis of the current art. IEEE Control Syst Mag 42–54, February 2006
48. Lin, M.G., Lakshminarayanan, S., Rangaiah, G.P.: A comparative study of recent/popular PID tuning rules for stable, First-Order Plus Dead Time, Single-Input Single-Output processes. Ind. Eng. Chem. Res. **47**(2), 344–368 (2008)
49. Madhuranthakam, C., Elkamel, A., Budman, H.: Optimal tuning of PID controllers for FOPTD, SOPTD and SOPTD with lead processes. Chem. Eng. Process. **47**(2), 251–264 (2008)
50. Maffezzoni, C., Rocco, P.: Robust tuning of PID regulators based on step response identification. Eur. J. Control **3**(2), 125–136 (1997)
51. Mellquist, M., Crisafulli, S., MacLeod, I.M.: Derivative filtering issues for industrial PID controllers. In: Proc. IEAust Control-97, Sydney, Australia, pp. 486–491 (1997)
52. Moradi, M.H.: New techniques for PID controller design. In: Proc. 2003 IEEE Conference on Control Applications, Istanbul, Turkey, pp. 903–908 (2003)
53. Morari, M., Zafiriou, E.: Robust Process Control. Prentice-Hall, Upper Saddle River (1989)
54. Murrill, P.W.: Automatic Control of Processes. International Textbook Co., Scranton (1967)
55. O'Dwyer, A.: Handbook of PI and PID Controller Tuning Rules. Imperial College Press, London (2006)
56. O'Dwyer, A.: PID control: the early years. In: Control in the IT Sector Seminar, Cork Institute of Technology, Cork, Ireland, May 2005
57. Pedersen, G.K.M., Yang, Z.: Multi-objective PID-controller tuning for a magnetic levitation system using NSGA-II. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary computation, pp. 1737–1744 (2006)
58. Piroddi, L., Leva, A.: Step response classification for model-based autotuning via polygonal curve approximation. J. Process Control **17**(8), 641–652 (2007)
59. Rivera, D.E., Skogestad, S., Morari, M.: Internal model control 4: PID controller design. Ind. Eng. Chem. Process Des. Dev. **25**(1), 252–265 (1986)
60. Rovira, A.A., Murrill, P.W., Smith, C.L.: Tuning controllers for setpoint changes. Instrum. Control Syst. **42**(12), 67–69 (1969)

61. Shinskey, F.G.: Process control: as taught versus as practiced. Ind. Eng. Chem. Res. **41**(16), 3745–3750 (2002)
62. Silva, G.J., Datta, A., Bhattacharyya, S.P.: New results on the synthesis of pid controllers. IEEE Trans. Autom. Control **47**(2), 241–252 (2002)
63. Skogestad, S.: Simple analytic rules for model reduction and PID controller tuning. J. Process Control **13**(4), 291–309 (2003)
64. Swanda, A.P., Seborg, D.E.: Controller performance assessment based on setpoint response data. In: Proc. ACC 1999, San Diego, CA, pp. 3863–3867 (1999)
65. Taiwo, O.: Comparison of four methods of on-line identification and controller tuning. IEE Proc., Control Theory Appl. **140**(5), 323–327 (2006)
66. Tan, W., Liu, J., Chen, T., Marquez, H.J.: Comparison of some well-known PID tuning formulas. Comput. Chem. Eng. **30**(9), 1416–1423 (2006)
67. Toscano, R.: A simple robust PI/PID controller design via numerical optimization approach. J. Process Control **15**(1), 81–88 (2005)
68. Umland, J.W., Safiuddin, M.: Magnitude and symmetric optimum criterion for the design of linear control systems: what is it and how does it compare to the others? IEEE Trans. Ind. Appl. **26**(3), 489–497 (1990)
69. Veronesi, M., Visioli, A.: Performance assessment and retuning of PID controllers. Ind. Eng. Chem. Res. **48**(5), 2616–2623 (2009)
70. Vilanova, R.: IMC based robust PID design: tuning guidelines and automatic tuning. J. Process Control **18**(1), 61–70 (2008)
71. Vilanova, R., Alfaro, V.M., Arrieta, O., Pedret, C.: Analysis of the claimed robustness for PI/PID robust tuning rules. In: Proc. 18th Mediterranean Conference on Control & Automation, Marrakech, Morocco, pp. 658–662 (2010)
72. Visioli, A.: Optimal tuning of PID controllers for integral and unstable processes. IEE Proc., Control Theory Appl. **148**(2), 180–184 (2001)
73. Wang, Q.G., Fung, H.W., Zhang, Y.: PID tuning with exact gain and phase margins. ISA Trans. **38**(3), 243–249 (1999)
74. Yu, C.C.: Autotuning of PID Controllers: A Relay Feedback Approach. Springer, London (2006)
75. Ziegler, J.G., Nichols, N.B.: Optimum settings for automatic controllers. Trans. Am. Soc. Mech. Eng. **64**, 759–768 (1942)

Chapter 3
PI/PID Controllers Design for Integrating and Unstable Systems

A. Seshagiri Rao and M. Chidambaram

3.1 Introduction

In industry, for many processes such as heating boilers, batch chemical reactors, liquid storage tanks, liquid level system with a constant outflow, the dynamic response is very slow with a large dominant time constant. It is observed that in a distillation column, the dynamics of the bottoms level control shows a large time constant and can be described as an integrating process [12]. There also exists industrial processes such as aerospace control systems, DC motors and high-speed disk drives whose dynamics show the characteristics of double integrator types (i.e., the process model transfer function is in the form of $ke^{-\theta s}/s^2$ or $ke^{-\theta s}/s^2(\tau s+1)$) [34]. These types of processes are approximated as integrating processes for the purpose of designing the controllers. Designing controllers based on integrating processes give a superior closed-loop performance than that of designing controller based on a first-order plus time delay process for both the nominal and model mismatch conditions. Several practical examples of integrating processes and controller design methodologies are described in [82]. A system whose transfer function has at least one pole lying in the right half-plane is known as an unstable system. Open-loop instability means that the system will move away from the steady state even for a small perturbation of the system parameters or operating conditions. Linearization of the mathematical model equations of such systems around the operating point will give a transfer function, which has at least one pole in the right half of the s-plane (RHP). The response of such transfer function models for a small perturbation may be ever

A. Seshagiri Rao
Department of Chemical Engineering, National Institute of Technology, Tiruchirappalli 620 015, India
e-mail: seshagiri@nitt.edu

M. Chidambaram (✉)
Department of Chemical Engineering, Indian Institute of Technology Madras, Chennai 600 036, India
e-mail: chidam@iitm.ac.in

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_3, © Springer-Verlag London Limited 2012

increasing, but in the case of nonlinear systems (i.e., real systems), the perturbation will cause the system steady state to move to another but stable steady state. In the case where the system has a single steady state that is an unstable one, the actual response will be oscillatory around this unstable steady-state point. Unstable processes occur because of several reasons and are listed below.

3.1.1 Processes with Steady State Multiplicities

Many real systems exhibit multiple steady states due to certain nonlinearity of the system(s). Some of the steady states may be unstable. Chemical reactors might present multiple steady states and oscillatory solutions, depending on particular operating conditions. Numerous works have shown that chemical reacting systems may present steady-state multiplicity, periodic solutions, and more complex behavior. Complex dynamic behavior has been found not only in chemical systems [2, 28, 43] but also in biological systems [45], separation processes [75], and recycle systems [38–40]. Sometimes it may be necessary to operate the system at an unstable steady state for economical and/or safety reasons.

3.1.2 Processes Exhibiting Sustained Oscillations

There is a class of systems which exhibits sustained oscillations (around the operating point) in the output in spite of constant input conditions. Crystallizers, Aerosol reactors, Continuous stirred tank reactor (CSTR), Polymerization reactors, and Bioreactors are examples of such systems. For example, an isothermal continuous crystallizer exhibits sustained oscillations in crystal size distribution (CSD) in spite of constant input conditions [13]. It may be important to maintain the crystal size distribution because poorly shaped crystals may promote agglomeration, caking and pose storage and processing problems. At certain operating conditions, the CSTR also shows sustained oscillations in its temperature. The linearization of the nonlinear model equations for such systems around the unstable steady state gives a transfer function model with unstable pole(s) [43]. An excellent overview on the physical occurrence of unstable processes and PID controller design methods is given in [8, 68]. Here, the reported mathematical model equations of an isothermal chemical reactor and bioreactor along with the model parameters of the transfer function models are summarized.

Isothermal CSTR The mathematical model of an isothermal CSTR is given as [36]

$$\frac{dc}{dt} = \left(\frac{Q}{V}\right)(c_f - c) - \left[\frac{k_1 c}{(k_2 c + 1)^2}\right], \tag{3.1}$$

where Q is the inlet flow rate, V is the volume of reactor, c is the reactant concentration, k_1 and k_2 are the kinetic rate constants, and c_f is the feed concentration. The values of the operating conditions are given by $Q = 0.03333$ 1/s, $V = 1$ l, $k_1 = 10$ 1/s, and $k_2 = 10$ 1/mol. For the nominal value of $c_f = 3.288$ mol/l, the steady-state solution of the model equation gives the following two stable steady states at $c = 1.7673$ mol/l and 0.01424 mol/l. There is one unstable steady state at $c = 1.3065$ mol/l. Feed concentration is the manipulated variable. Linearization of the model equation around this operating condition $c = 1.3065$ mol/l gives the following unstable transfer function model relating the reactor concentration to feed concentration by considering a measurement delay of 20 s:

$$\frac{\Delta c(s)}{\Delta c_f(s)} = \frac{3.326e^{-20s}}{99.69s - 1}.$$

Bioreactor A nonlinear continuous bioreactor exhibits output multiplicity behavior. The model equations are given by [3]

$$\frac{dX}{dt} = (\mu - D)X,$$
$$\frac{dS}{dt} = (S_f - S)D - \frac{\mu X}{\gamma},$$

where X and S are the concentrations of the cell and the substrate, respectively, $\mu = \mu_m S/(K_m + S + K_l S^2)$. The model parameters are $\gamma = 0.4$ g/g, $S_f = 4$ g/l, $\mu_m = 0.53h^{-1}$, $D = 0.3h^{-1}$, $K_m = 0.12$ g/l, $K_l = 0.4545$ l/g. The reactor exhibits steady states ($X = 0$, $S = 4$), (0.9951, 1.5122), and (1.5301, 0.1746). It is desired to operate the reactor at unstable steady state ($X = 0.9951$, $S = 1.5122$). The dilution rate D is used as a manipulated variable. A delay of 2.4 hours is considered in the measurement of X. When the nonlinear model equations are linearized around the unstable operating point ($X = 0.9951$, $S = 1.5122$), the following transfer function model is obtained:

$$\frac{\Delta X(s)}{\Delta D(s)} = \frac{-5.89e^{-2.4s}}{5.86s - 1}.$$

Table 3.1 gives unstable transfer function models reported in the literature for various systems. To achieve good closed-loop performances for unstable and integrating systems, the controller should be designed properly. However, there are some difficulties involved for these types of systems when compared to that of stable systems.

3.1.3 Difficulties Involved with Unstable Time Delay Processes

The dynamics of many integrating systems are represented by pure integrating processes, integrating first-order plus time delay, double integrating process, and the

Table 3.1 Unstable processes reported in the literature

S.No.	Process	Controlled and manipulated variables	Transfer function
1	Isothermal CSTR [36]	Outlet concentration, feed concentration	$3.326e^{-20s}/(99.69s-1)$
2	Bioreactor [3]	Biomass concentration, dilution rate	$-5.89e^{-2.4s}/(5.86s-1)$
3	Dimerization reactor [4]	Reactor temperature, collant flow rate	$-0.017e^{-2.4s}/(5.8s-1)$
4	Gas phase polyolefin reactor [57]	Reactor temperature, cooling water temperature	$k_p/(s^2+a_1s+a_0)$, the denominator coefficients a_1, a_0 change their values and signs depending on the operating conditions
5	Fluidized bed reactor [27]	Temperature of the bed, coolant flow rate	$1/(s+0.8695)(s-0.0056)$
6	Nonideal CSTR [36]	Outlet concentration, feed concentration	$2.21(1+11.133s)e^{-20s}/(98.3s-1)$
7	Fluid catalytic cracker [2]	Riser top temperature, catalyst circulation rate	$18(s-0.0019)/(s-0.00025)$
8	X-29 air craft [14]	Altitude, actuator signal	$(s-26)/(s-6)$
9	Klein's unridable bicycle [29]	Steering angle, tilt angle	$3.5(5-0.7s)/(s^2-6.867)$
10	Autocatalytic CSTR [9]	Outlet concentration, feed concentration	$0.2679(1-41.667s)$ $/(41.667s^2+279.03s+1)$
11	Distillation column [23]	Top product composition, reflux rate	$0.003551(s-0.00492)$ $/(s-0.00276)(s+0.0114)$
12	Jacketed CSTR [6]	Reactor temperature, jacket temperature	$(0.8714s+6.963)$ $/(s^2+2.848s-1.132)$
13	Crystallizer [26]	Number of crystals, inlet feed concentration	$0.03(s+1)/(0.516s^2-0.0945s+1)$
14	Modified cart and pole problem [33]	Cart position, control input	$(0.98s^2-14.36)/s^2(s^2-15.79)$
15	Vertical takeoff airplane [16]	Angle, commanded voltage for the power amplifiers	$b/s^2(s+a)$
16	Fly by wire helicopter [78]	Pitch attitude, actuator drive signal	$-13.48(s+0.55)(s-0.005)$ $/(s+12.58)(s^2+1.26s+0.49)$ $\times(s^2-1.2s+0.09)$
17	Autocatalytic CSTR [31]	Reactor temperature, coolant flow	$(-0.0378s-0.0001)$ $/(s^3+0.6907s^2$ $-0.0592s+0.0012)$
18	Adiabatic CSTR carrying out Van de Vusse reaction [80]	Concentration, feed flow rate	$(-0.624s^2+77.8476s+4.2157)$ $/(s^3-123.96s^2-105.888s$ $-16.1412)$

dynamics of many unstable systems are represented by unstable first-order plus time delay (FOPTD) or second-order plus time delay (SOPTD) transfer function models. Some processes also consist of numerator zeros. The performance of the control system is limited for integrating and unstable processes compared to that of the stable processes. Controller design for such systems is more difficult than those of the stable systems. Performance specifications like overshoot and settling time are larger compared to those of the stable systems. For unstable systems with time delay, there exists a minimum value of controller gain below which the closed-loop system cannot be stabilized. There also exists a maximum value of controller gain above which the closed-loop system cannot be stabilized. The design value of the controller gain is the average of these two limiting values. The values of maximum and minimum controller gain narrow down as the time delay increases, restricting the performance of the closed-loop system. The presence of a positive zero in the transfer function model increases the overshoot and the inverse response. The control system performance is further complicated by the presence of a zero.

3.2 Literature Review on PI/PID Controllers Design

In the open literature, several PI and PID controllers design methods have been proposed for the control of integrating and unstable processes with and without the time delay. The methods can be broadly classified as (i) Methods based on stability analysis, (ii) Methods based on phase margin and gain margin specifications, (iii) Direct synthesis method, (iv) Pole placement method, (v) Optimization based approaches, (vi) Model reference method, (vii) First stabilized by P or PD controller, then designing PID for the stabilized system, (viii) IMC method, and (ix) Equating coefficients method. Here, some important methods are reviewed. The literature on controller design for integrating systems is given first followed by unstable systems. As it is difficult to cover all the literature, more emphasis is given here to the literature published based on direct synthesis and IMC and related methods after the year 2000 for single-input single-output (SISO) integrating and unstable systems.

Tyreus and Luyben [71] showed that the IMC-based PI controller can lead to poor control performance if the closed-loop time constant is not chosen properly and proposed simple tuning rules based on the classical frequency response method to achieve maximum closed-loop log modulus of 2 dB. Luyben [41] extended the method of Tyreus and Luyben [71] for designing a PID controller. The settings given in the paper are for the series form of PID controller with a first-order filter. According to their method, the integral time constant is equal to 2.2 times the ultimate period, the derivative time constant is equal to the reciprocal of the ultimate frequency, the controller gain is equal to 0.46 times the ultimate gain, and filter time constant is equal to 0.1 times the derivative time. Lee et al. [32] proposed tuning rules based on IMC method and Maclaurin's series expansion after converting integrating process into unstable process. Visioli [77] proposed tuning methods for integrating systems based on minimizing ISE (integral square error), ISTE, and ITSE performance indices. A genetic algorithm is used for the optimization problem. The resulted controller settings are fitted by simple equations. The optimization for the servo response gives a PD controller. For a servo problem, Visioli [77] could not get any PID settings. However, for the regulatory problem with the assumption that the load transfer function is the same as the process transfer function, PID settings are obtained by ISE cost function minimization using a genetic algorithm.

Wang and Cai [79] have derived PID settings using a two-degree-of-freedom controller and specifying phase margin and gain margin for the stabilized system in the inner loop. Chidambaram and Sree [11] proposed tuning rules based on equating coefficient method. Skogestad [63] proposed PID tuning rules based on IMC principles. Tuning rules are proposed for integrating, double integrating, and integrating with first-order plus time delay process. These tuning rules are named as SIMC tuning rules and became popular since 2003. Sree and Chidambaram [67] designed a PID controller based on equating coefficient method. Arbogast and Cooper [1] proposed PID controller in series with a first-order filter based on IMC principles and recommended that the IMC tuning parameter may be taken as 10. They have experimentally verified the PID settings on a level control tank. Shamsuzzoha and Lee [58, 59] proposed PID controller in series with lead/lag filters based on IMC principles and provided guidelines for selection of the tuning parameter and analyzed robustness based on maximum sensitivity.

Song et al. [64] proposed a stabilization algorithm for integrating systems based on D-partition and graphical techniques. Eriksson et al. [15] developed PID tuning rules for time varying delays with maximum amplitude of the varying delay and Jitter Margin. The jitter margin is an upper bound for any additional delay that can be added into a closed-loop system while maintaining stability. The delay can be of any type (constant, time-dependent, random). Rao et al. [55] designed controller based on direct synthesis method and obtained improved control performances. Minimization of (ISE) criterion results in a PD controller for integrating processes. The PD controller gives good servo response but fails to reject the load disturbances. To overcome this problem, Ali and Majhi [5] proposed a method based on ISE minimization with the constraint that the slope of the Nyquist curve has a specific value at the gain crossover frequency. They have provided guidelines for selecting the gain crossover frequency and the slope of the Nyquist curve. Pai et al. [49] proposed direct synthesis for disturbance-rejection-based tuning formulae for integrating processes. Foley et al. [17] compared several tuning formulae for controlling integrating time delay processes and concluded that derivative action is required to improve the achievable performance unless the signal-to-noise ratio is very low.

Veronesi and Visioli [76] developed a PD controller if the objective is only set point tracking and different PID controller tuning rules by assuming that the objective is both set point tracking and load disturbance rejection and only disturbance rejection respectively. They used reduced order integrating model, closed-loop index (CI), and IMC method to develop the controller settings, where, CI is the ratio of IAE if a PD controller is used to the actual IAE of the closed-loop system. Shamsuzzoha and Skogestad [62] proposed new tuning rules based on the closed-loop set point response. A proportional controller is used initially to get desired overshoot, and based on this closed-loop response, PI controller settings are provided.

Recently, Hu and Xiao [21] developed tuning rules similar to those of Shamsuzzoha and Skogestad [62] and proposed guidelines for initial selection of the proportional gain to get desired percentage levels of the overshoot. Recently, simple tuning rules are provided based on the IMC principles by Rao and Sree [56] for integrating systems, which gives an improved performance over many existing methods.

Lee et al. [32] proposed methods of designing PI/PID controllers for unstable systems without and with a negative zero based on IMC principles, and they have used the Maclaurin's series expansion to get a PID controller. Though Lee et al. [32] have given a general methodology to design a PID controller, the derivation is tedious due to the Maclaurin's series expansion particularly for the unstable SOPTD system with a zero. Kwak et al. [30] proposed stabilizability conditions and analyzed the structural limitations of the PID controller and proposed a two-stage design procedure for unstable processes: a PD controller is used for stabilization, and then for the stabilized system, PID controller is designed using an optimization criterion. Jhunjhunwala and Chidambaram [24] proposed a design method based on minimization of the ISE criterion for set point response and developed tuning rules for the PID controller. They have developed the tuning rules based on time delay to time constant ratio. Wang and Cai [79] have derived PID settings using two-degree-of-freedom controller and specifying phase margin and gain margin for the stabilized system in the inner loop. Since two-degree-of-freedom controller is equivalent to a set-point weighted PID controller, they have given equation to calculate the set-point weighting parameter.

Sree et al. [69] developed tuning rules based on equating coefficient method for first-order unstable process with time delay. Rao and Chidambaram [52] developed a design strategy for unstable second-order processes with time delays. Zhou et al. [83] compared several tuning formulae for unstable time delay processes; however, simple PID controllers are not compared; rather they compared two-degree-of-freedom control schemes like the modified Smith predictor and modified IMC schemes with more than two controllers. Shamsuzzoha and Lee [60] proposed IMC-based tuning rules for second-order unstable processes with a zero and obtained improved performance. Chen et al. [7] provided new tuning relations for selection of the set-point weighting tuning parameter and showed improved performance. Shamsuzzoha and Lee [61] proposed IMC-based tuning rules for PID controller with lead/lag filter for enhanced disturbance rejection. Panda [50] provided tuning rules based on the IMC method and Laurent's series expansion for PID controller for both integrating and unstable processes. The tuning rules developed by Shamsuzzoha and Skogestad [62] and Hu and Xiao [21] also are applicable for unstable processes. Nie et al. [46] proposed a two-loop controller design procedure based on new gain and phase margin specifications. Both upper and lower values of the gain margins are considered, and solution is provided from intersection of the curves generated between two coupled nonlinear equations, which are obtained from amplitude and phase relations.

To enhance the closed loop-performance, two-degree-of-freedom control schemes are proposed. Here, two degree-of-freedom control schemes in the form of modified Smith predictor structures are briefly reviewed. Majhi and Atherton [42] proposed a design methodology based on autotuning. Tan et al. [70] proposed a design method based on IMC methodology. Later, several authors [20, 35, 37, 47, 51, 53, 73, 81] proposed design methodologies for the modified Smith predictor. Further, the design methodologies are extended for cascade control schemes [72, 74]. Govindhakannan and Chidambaram [18] reported the design of PI (PID) centralized and decentralized PI controllers for 2×2 multivariable unstable systems with time delays. For 2×2 systems with unstable interactions (unstable poles) present in both ways, the decentralized PI controllers do not stabilize the system, whereas the centralized PI controllers stabilize the system. Govindhakannan and Chidambaram [19] presented a study on the stabilization of multivariable unstable systems first by centralized proportional controllers followed by the design of decentralized or centralized PID controllers for the stabilized systems. Two-stage centralized controller design method gives improved performance than the single-stage centralized controllers.

From the analysis of the literature, it is observed that most of the PI/PID controller tuning method(s) have at least one tuning parameter. Selection of this tuning parameter is crucial to obtain good closed-loop performances. Different authors have selected different performance indices such as (i) minimization of IAE, ISE, ITAE, ITSE, (ii) peak value of the sensitivity function, and (iii) minimization of overshoot for selection of the tuning parameter. Each method has some advantages and some limitations. Some methods address the design of simple PI or PID controllers, and some methods address the design of PI or PID controller with lead/lag filters. Further, two-degree-of-freedom controllers are also proposed by several authors. Note that two-degrees-of-freedom is a general word and there are many control structures such as (i) stabilizing with an inner loop with P or PD controller followed by PI or PID controller in the outer loop, (ii) treating set-point weighting or set-point filter as additional degrees of freedom, and (iii) methods based on modified IMC and Smith predictor schemes under this category. Most of the methods are developed for SISO systems. There are very few methods available for the design of PI/PID controllers for MIMO unstable and integrating systems.

In the next sections, PI/PID controller design methods for integrating and unstable processes based on direct synthesis method [52, 55], IMC method [68], and Equating coefficient method [11, 67, 69] are discussed in detail.

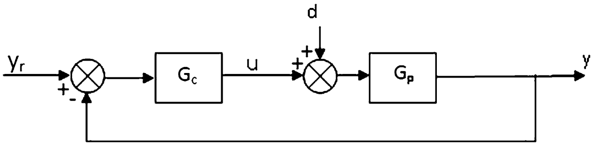
3.3 Design Based on the Direct Synthesis Method

The direct synthesis method is a model based approach in which a suitable closed loop transfer function for the servo problem should be specified for the design of the controller.

3.3.1 Controller Design for Integrating Systems [55]

Typically, the classes of integrating processes with time delay can be represented by any of the following transfer function models: pure integrating process plus time

Fig. 3.1 Feedback control loop



delay (IPTD),

$$G_p = ke^{-\theta s}/s; \tag{3.2}$$

integrating plus first-order plus time delay (IFOPTD),

$$G_p = ke^{-\theta s}/s(\tau s + 1); \tag{3.3}$$

integrating plus unstable first-order plus time delay (IUFOPTD),

$$G_p = ke^{-\theta s}/s(\tau s - 1); \tag{3.4}$$

pure double integrating plus time delay (DIPTD),

$$G_p = ke^{-\theta s}/s^2; \tag{3.5}$$

double integrating plus first-order plus time delay (DIFOPTD),

$$G_p = ke^{-\theta s}/s^2(\tau s + 1). \tag{3.6}$$

If the integrating process is of higher order, then it can be reduced to the form of any of the above-mentioned process models by using an appropriate identification method. The simple feedback block diagram showing the process and the controller is shown in Fig. 3.1, where G_p is the transfer function of the integrating process, and G_c is the transfer function of the controller, y is the closed loop output, y_r is the reference point, d is the disturbance.

The closed-loop relation for set point changes is given by

$$\frac{y}{y_r} = \frac{G_c G_p}{1 + G_c G_P}. \tag{3.7}$$

According to the direct synthesis method, from (3.7), the controller is given by

$$G_c = \frac{(y/y_r)_d}{G_p[1 - (y/y_r)_d]}, \tag{3.8}$$

where $(y/y_r)_d$ is the desired closed-loop transfer function (DCLTF) for set point changes. With this, the controller is designed for different types of integrating processes ((3.2)–(3.6)) after specifying the DCLTF. Note that selection of $(y/y_r)_d$ is important and it should be selected in such a way that it results in a realizable controller and the obtained controller should provide good nominal and robust performances. For selection of the DCLTF, (a) the order of the denominator in the DCLTF should be greater than that of the process for which the controller is going to be designed, (b) if the process has positive (right half-plane) zeros, then all RHP zeros should be considered in the DCLTF, (c) if the process has time delay, this time delay should be considered, and (d) after doing this, the numerator of the DCLTF should be selected, and proper tuning relations should be provided for the numerator parameters. Here, the desired closed-loop responses are selected to achieve this objective after conducting many simulation studies.

Case (i) If the process is of (3.2), i.e., $G_p = ke^{-\theta s}/\mathrm{s}$, the desired closed-loop transfer function is considered as

$$(y/y_r)_d = (\eta s + 1)e^{-\theta s}/(\lambda s + 1)^2. \tag{3.9}$$

Upon substituting and with a first-order Pade approximation for the time delay $[e^{-\theta s} = (1 - 0.5\theta s)/(1 + 0.5\theta s)]$, the controller is obtained by considering $\eta = 2\lambda + \theta$ as

$$G_c = k_c\left(1 + \frac{1}{\tau_i s} + \tau_d s\right)\frac{1}{\tau_f s + 1}, \tag{3.10}$$

where

$$k_c = \frac{2\lambda + 1.5\theta}{k(\lambda^2 + 2\lambda\theta + 0.5\theta^2)}, \qquad \tau_i = 2\lambda + 1.5\theta,$$

$$\tau_d = \frac{\theta\lambda + 0.5\theta^2}{2\lambda + 1.5\theta}, \qquad \tau_f = \frac{0.5\theta\lambda^2}{(\lambda^2 + 2\lambda\theta + 0.5\theta^2)}.$$

Case (ii) If the process is of $G_p = ke^{-\theta s}/s(\tau s + 1)$, the desired closed-loop transfer function is considered as

$$(y/y_r)_d = \left(\eta_2 s^2 + \eta_1 s + 1\right)e^{-\theta s}/(\lambda s + 1)^3. \tag{3.11}$$

Using a first-order Pade approximation for the time delay, after simplification, the controller is obtained as

$$G_c = k_c\left(1 + \frac{1}{\tau_i s} + \tau_d s\right)\frac{\alpha s + 1}{\beta s + 1}, \tag{3.12}$$

where

$$k_c = \frac{\eta_1}{k(3\lambda^2 + 1.5\lambda\theta + 0.5\theta\eta_1 - \eta_2)}, \qquad \tau_i = \eta_1,$$

$$\tau_d = \frac{\eta_2}{\eta_1}, \qquad \alpha = 0.5\theta, \qquad \beta = \frac{0.5\theta\lambda^3}{\tau(3\lambda^2 + 1.5\lambda\theta + 0.5\theta\eta_1 - \eta_2)},$$

with $\eta_1 = 3\lambda + \theta$ and $\eta_2 = \frac{(0.5\theta-\tau)\lambda^3+(3\tau^2-1.5\theta\tau)\lambda^2+3\theta\tau^2\lambda+0.5\theta^2\tau^2}{\tau(0.5\theta+\tau)}$.

Case (iii) If the process is of $G_p = ke^{-\theta s}/s(\tau s - 1)$, the controller design procedure is same as explained in case (ii), the controller structure is obtained according to (3.12) with the controller parameters

$$\begin{aligned} &k_c = \frac{\eta_1}{k(\eta_2 - 3\lambda^2 - 1.5\lambda\theta - 0.5\theta\eta_1)}, \qquad \tau_i = \eta_1, \qquad \tau_d = \frac{\eta_2}{\eta_1}, \\ &\alpha = 0.5\theta, \qquad \beta = \frac{0.5\theta\lambda^3}{\tau(\eta_2 - 3\lambda^2 - 1.5\lambda\theta - 0.5\theta\eta_1)} \end{aligned} \tag{3.13}$$

with $\eta_1 = 3\lambda + \theta$ and $\eta_2 = \frac{(0.5\theta+\tau)\lambda^3+(3\tau^2+1.5\theta\tau)\lambda^2+3\theta\tau^2\lambda+0.5\theta^2\tau^2}{\tau(\tau-0.5\theta)}$.

Case (iv) If the process is of double integrating with time delay, i.e., $G_p = ke^{-\theta s}/s^2$, the desired closed-loop transfer function is assumed as given in (3.11). Substituting (3.11) into (3.8) and simplifying by approximating the time delay as a first-order Pade, the controller structure is obtained as (3.12) with the parameters

$$\begin{aligned} &k_c = \frac{\eta_1}{k(\lambda^3 + 1.5\lambda^2\theta + 0.5\theta\eta_2)}, \qquad \tau_i = \eta_1, \qquad \tau_d = \frac{\eta_2}{\eta_1}, \\ &\alpha = 0.5\theta, \qquad \beta = \frac{0.5\theta\lambda^3}{(\lambda^3 + 1.5\lambda^2\theta + 0.5\theta\eta_2)} \end{aligned} \tag{3.14}$$

where $\eta_1 = 3\lambda + \theta$ and $\eta_2 = 3\lambda^2 + 1.5\theta\lambda + 0.5\theta\eta_1$.

Case (v) If the process is of $G_p = ke^{-\theta s}/s^2(\tau s + 1)$, the desired closed-loop transfer function is considered as

$$(y/y_r)_d = \big(\eta_2 s^2 + \eta_1 s + 1\big)e^{-\theta s}/(\lambda s + 1)^4. \tag{3.15}$$

By substituting (3.6) and (3.15) into (3.8) and using a first-order Pade approximation for time delay, the controller is obtained after considering $\eta_1 = 4\lambda + \theta$ and $\eta_2 = 6\lambda^2 + 2\theta\lambda + 0.5\theta\eta_1$ as

$$G_c = k_c\left(1 + \frac{1}{\tau_i s} + \tau_d s\right)\frac{\alpha_2 s^2 + \alpha_1 s + 1}{\beta_2 s^2 + \beta_1 s + 1}, \tag{3.16}$$

where

$$k_c = \frac{\eta_1}{k(4\lambda^3 + 3\lambda^2\theta + 0.5\theta\eta_2)}, \qquad \tau_i = \eta_1, \qquad \tau_d = \frac{\eta_2}{\eta_1}, \qquad \alpha_2 = 0.5\theta\tau,$$

$$\alpha_1 = 0.5\theta + \tau, \qquad \beta_2 = \frac{0.5\theta\lambda^4}{(4\lambda^3 + 3\lambda^2\theta + 0.5\theta\eta_2)},$$

$$\beta_1 = \frac{\lambda^4 + 2\theta\lambda^3}{(4\lambda^3 + 3\lambda^2\theta + 0.5\theta\eta_2)}.$$

The primary requirement for selection of λ is that the resulting controller gains should be positive for positive values of k. Hence, to get positive values of controller gain (k_c) for cases (ii) and (iii), the constraints to be followed are $\eta_2 <$

$3\lambda^2 + 1.5\theta\lambda + 0.5\theta\eta_1$ and $\eta_2 > 3\lambda^2 + 1.5\theta\lambda + 0.5\theta\eta_1$, respectively. Further, λ should be selected in such way that the resulting controller gives good nominal and robust control performances. After carrying out several simulation studies, the recommended range of the tuning parameter that gives good control performances is 0.8θ to 3θ. For robust design (good closed-loop performance in the presence of process uncertainties), it is recommended that the value of β obtained is considered as '0.1β'. To reduce large overshoot, set-point weighting is suggested [68], and thus the PID controller in (3.10, 3.12, 3.16) should be implemented in the form

$$u(t) = k_c\left[(\sigma y_r - y) + \left(\frac{1}{\tau_i}\right)\int e\,dt + \tau_d\frac{de}{dt}\right], \tag{3.17}$$

where $\sigma = y_r - y$ is the set-point weighting parameter. The set-point weighting parameter should be selected in such way that the resulting controller should not give large overshoot responses. Based on extensive simulations conducted on different types of integrating processes, it is suggested that σ can be taken in the range of 0.3 to 0.4 [54].

Stability and Robustness For any closed-loop control system, it is necessary to analyze the stability and robustness for uncertainties in the process, and this is analyzed here using the small gain theorem. The general definitions of the small gain theorem are given here. The closed-loop system is robustly stable if and only if [44]

$$\left\| l_m(j\omega)T(j\omega)\right\| < 1 \quad \forall\omega \in (-\infty, \infty), \tag{3.18}$$

where $T(j\omega)$ is the complementary sensitivity function, and $l_m(j\omega)$ is the bound on the process multiplicative uncertainty. The process uncertainty can be represented as

$$l_m(j\omega) = \left|\frac{G_p(j\omega) - G_m(j\omega)}{G_m(j\omega)}\right|,$$

where $G_m(j\omega)$ is the model of the integrating process. To show the stability and robustness analysis more qualitatively, let us consider the integrating process (3.2) for which the complementary sensitivity function of the closed-loop with the designed controller (3.10) is

$$T(j\omega) = \frac{kk_c(1 + \tau_i s + \tau_i\tau_d s^2)e^{-\theta s}}{s(\tau_f s + 1) + kk_c(1 + \tau_i s + \tau_i\tau_d s^2)e^{-\theta s}},$$

where the controller parameters k_c, τ_i, τ_d and τ_f are the functions of the tuning parameter λ. If uncertainty exists in the time delay, then the tuning parameter should be selected such that

$$\left\| T(j\omega)\right\|_\infty < \frac{1}{|e^{-\Delta\theta s} - 1|}.$$

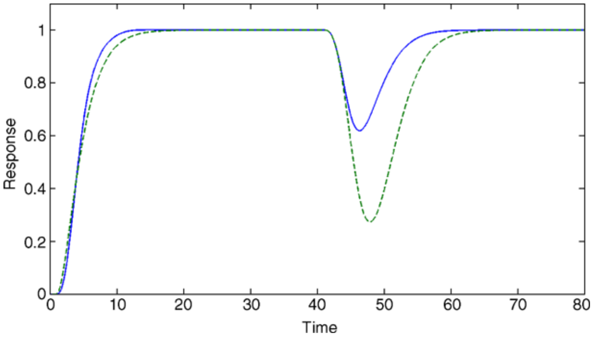


Fig. 3.2 Responses for perfect model for the process $G_p = e^{-s}/s^2(s+1)$, *solid*—proposed, *dashed*—Liu et al. [34]

If the uncertainty exists in the gain, then the tuning parameter should be selected in such a way that

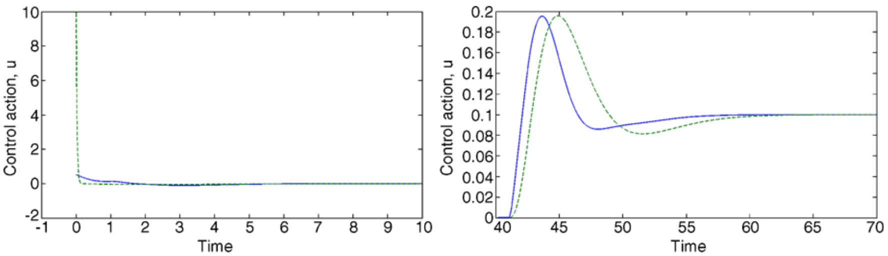
$$\left\|T(j\omega)\right\|_\infty < \frac{k}{|\Delta k|}.$$

Also for ensuring that the closed-loop performance is robust, the constraints to be followed by the sensitivity and complementary sensitivity functions are (3.18) and [44]

$$\left\|l_m(j\omega)T(j\omega) + w_m(j\omega)\bigl(1 - T(j\omega)\bigr)\right\| < 1. \tag{3.19}$$

Here, $w_m(j\omega)$ is the uncertainty bound on the sensitivity function which is given by $1 - T(j\omega)$. The controller should satisfy the robust stability and robust performance constraints ((3.18) and (3.19)). Similar stability and robustness analyses can be done for the remaining types of integrating processes with time delay.

Simulation Study For simulation purposes, a double integrating process [55] is considered as $G_p = e^{-s}/s^2(s+1)$. The controller parameters obtained for the proposed method are $k_c = 0.256$, $\tau_i = 6.6$, $\tau_d = 2.7061$, $\alpha_2 = 0.4$, $\alpha_1 = 1.5$, $\beta_2 = 0.074$, and $\beta_1 = 0.074$ with the tuning parameter selected as $\lambda = 1.4\theta = 1.4$. The set-point weighting parameter is chosen as $\sigma = 0.3$. Using these values the performance is evaluated by giving unit step change in the set point and a negative step input of 0.1 in the load (d) at $t = 40$ s, respectively. For comparison, the method proposed by Liu et al. [34] is considered. Figure 3.2 shows the responses for perfect model, and Fig. 3.3 shows the corresponding control action responses for set-point tracking and disturbance rejection. The proposed method of this chapter section performs better. Figure 3.4 shows the responses for a perturbation of $+15\%$ in the process time delay and -15% in the time constant, and Fig. 3.5 shows the corresponding control action responses. Note that the control action responses are given separately for set-point tracking and disturbance rejection for clear illustration. Here also, the proposed method gives improved performances. Similar performance improvement is observed for perturbations in the process gain.



(a) Setpoint tracking

(b) Disturbance rejection

Fig. 3.3 Control action responses for perfect model for the process $G_p = e^{-s}/s^2(s+1)$, *solid*—proposed, *dashed*—Liu et al. [34]

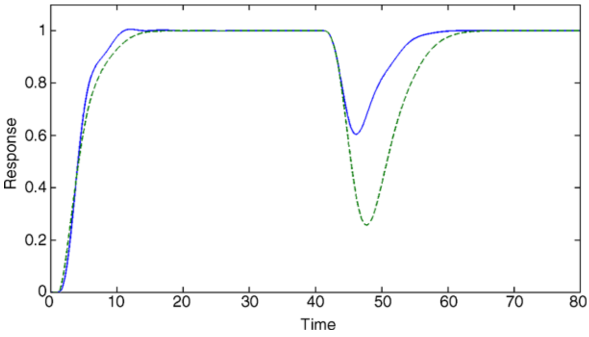
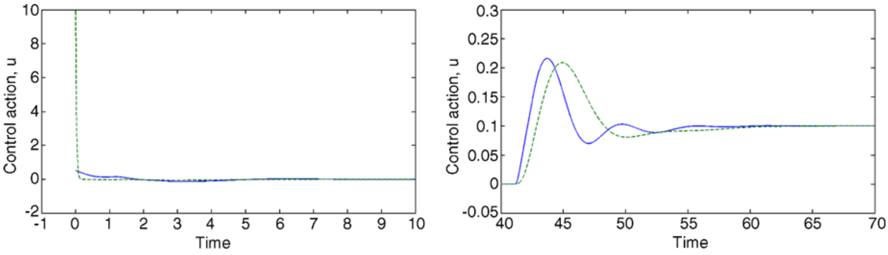


Fig. 3.4 Responses for perturbation of $+15\%$ in time delay and -15% in time constant for the process $G_p = e^{-s}/s^2(s+1)$, *solid*—proposed, *dashed*—Liu et al. [34]



(a) Setpoint tracking

(b) Disturbance rejection

Fig. 3.5 Control action responses for perturbation of $+15\%$ in time delay and -15% in time constant for the process $G_p = e^{-s}/s^2(s+1)$, *solid*—proposed, *dashed*—Liu et al. [34]

3.3.2 Controller Design for Unstable Systems [68]

This section presents a method for the selection of closed loop transfer function and associated controller settings for (i) unstable FOPTD system, (ii) unstable SOPTD system with a zero. If (3.8) is applied as such to unstable systems, the controller turns out to be unstable. To overcome this problem, it is suggested that the closed loop system is assumed to be a second order transfer function model with a numerator dynamics term [48]. The same idea is followed here for the design of controllers for unstable processes.

3.3.2.1 Case-1: Unstable FOPTD Systems

The process transfer function is considered as

$$G_p = ke^{-\theta s}/(\tau s - 1). \tag{3.20}$$

The closed-loop transfer function is assumed as

$$(y/y_r)_d = (\eta s + 1)e^{-\theta s}/(\lambda s + 1)^2,$$

where λ is the tuning parameter. With this, after approximating the time delay term as $e^{-\theta s} = (1 - \theta s)$ and simplifying, the controller is obtained as

$$G_c = k_c\left(1 + \frac{1}{\tau_i s}\right), \tag{3.21}$$

where

$$k_c = \frac{1}{k(\eta - 2\lambda - \theta)}, \qquad \tau_i = \eta, \quad \eta = \frac{\lambda^2 + 2\lambda\tau + \theta\tau}{\tau - \theta}.$$

If the $e^{-\theta s}$ term is approximated as $e^{-\theta s} = (1 - 0.5\theta s)/(1 + 0.5\theta s)$, the expression for G_c is obtained as

$$G_c = k_c\left(1 + \frac{1}{\tau_i s} + \tau_d s\right)\frac{1}{\beta s + 1}, \tag{3.22}$$

where

$$k_c = \frac{(\eta + 0.5\theta)}{k(\eta - 2\lambda - \theta)}, \qquad \tau_i = 0.5\theta + \eta, \qquad \tau_d = \frac{0.5\eta\theta}{0.5\theta + \eta}$$

with $\beta = \frac{0.5\lambda^2\theta}{\tau(\eta - 2\lambda - \theta)}$ and $\eta = \frac{\lambda^2\tau + \lambda\theta\tau + 0.5\lambda^2\theta + 2\lambda\tau^2 + \theta\tau^2}{\tau - 0.5\theta}$.

In order to provide an analytical guideline for the selection of the tuning parameter λ, many simulation studies are carried out on various unstable processes for minimum of integral of absolute error (IAE) cost function, and analytical relations are fitted as functions of $\varepsilon = \theta/\tau$ and are obtained as follows.

For PID:

$$\lambda/\tau = \begin{cases} 2.03\varepsilon^2 + 0.4958\varepsilon + 0.0326 & \text{if } 0 \le \varepsilon \le 0.8, \\ 4.5115\varepsilon^{5.661} & \text{if } 0.8 < \varepsilon < 1.0. \end{cases}$$

For PI:

$$\lambda/\tau = \begin{cases} 0.025 + 1.75\varepsilon & \text{if } \varepsilon \le 0.1, \\ 2\varepsilon & \text{if } 0.1 \le \varepsilon \le 0.5, \\ 2\varepsilon + 5\varepsilon(\varepsilon - 0.5) & \text{if } 0.5 \le \varepsilon \le 0.7. \end{cases}$$

The PI and PID settings are calculated for various values of ε, and the settings are fitted by the following equations.

For PID:

$$k_c k_p = \begin{cases} 4282\varepsilon^2 - 1334.6\varepsilon + 101 & \text{if } \varepsilon < 0.2, \\ 1.1161\varepsilon^{-0.9427} & \text{if } 0.2 \le \varepsilon \le 1.0, \end{cases}$$

$$\tau_i/\tau = \begin{cases} 36.842\varepsilon^2 - 10.3\varepsilon + 0.8288 & \text{if } 0 \le \varepsilon \le 0.8, \\ 76.241\varepsilon^{6.77} & \text{if } 0.8 < \varepsilon \le 1.0, \end{cases}$$

$$\tau_d/\tau = 0.5\varepsilon \quad \text{if } 0 \le \varepsilon \le 1.0.$$

For PI:

$$k_c k_p = \begin{cases} 49.535e^{-21.644\varepsilon} & \text{if } \varepsilon < 0.1, \\ 0.8668\varepsilon^{-0.8288} & \text{if } 0.1 \le \varepsilon \le 0.7, \end{cases}$$

$$\tau_i/\tau = 0.1523e^{7.9425\varepsilon} \quad \text{if } 0 \le \varepsilon \le 0.7.$$

Analytical expressions are also fitted for set-point weighting parameter (σ) (refer (3.6)) for PID and PI controllers separately and are obtained as follows.

For PID:

$$\sigma = \begin{cases} 10.17\varepsilon^2 - 3.426\varepsilon + 0.495 & \text{if } \varepsilon \le 0.2, \\ -0.232\varepsilon + 0.287 & \text{if } 0.2 \le \varepsilon \le 1.0. \end{cases}$$

For PI:

$$\sigma = \begin{cases} 0.33 & \text{if } \varepsilon \le 0.1, \\ 0.3646 - 0.437\varepsilon & \text{if } 0.1 \le \varepsilon \le 0.7. \end{cases}$$

Simulation Application to Isothermal CSTR Model A PID controller is designed and simulated for an isothermal CSTR model exhibiting multiple steady-state solutions (3.1). For this unstable first-order plus time delay system, a PID controller is designed by the present method as: $k_c = 1.616$, $\tau_i = 85.73$, and $\tau_d = 8.813$. For the purpose of comparison, the method proposed by Huang and Chen [22] is considered. The regulatory responses for these two PID settings for a step disturbance in Q from 0.03333 to 0.03 1/s are evaluated on the nonlinear model, and the responses are shown in Fig. 3.6. The corresponding manipulated variable responses are shown in Fig. 3.7. The regulatory responses for an uncertainty in time delay are also evaluated (24 s delay in the process, whereas the controllers settings are

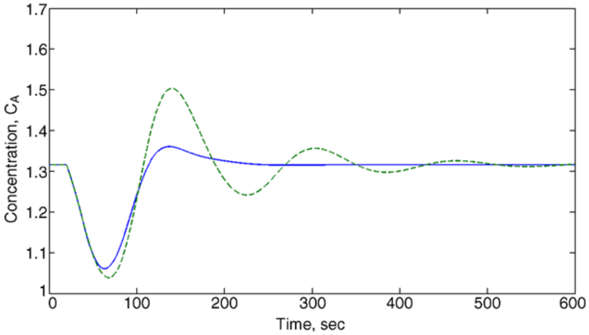


Fig. 3.6 Regulatory response of isothermal CSTR for a step change in q from 0.03333 to 0.03 $1/\mathrm{s}$, *solid*—present method, *dashed*—Huang and Chen [22]

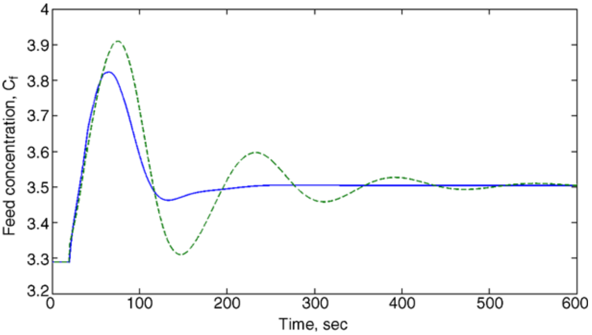


Fig. 3.7 Manipulated variable (feed concentration) response of isothermal CSTR, *solid*—present method, *dashed*—Huang and Chen [22]

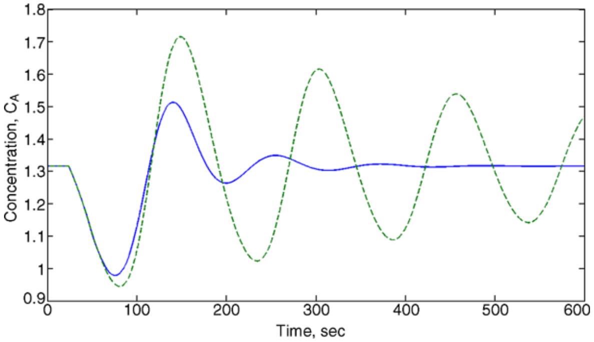
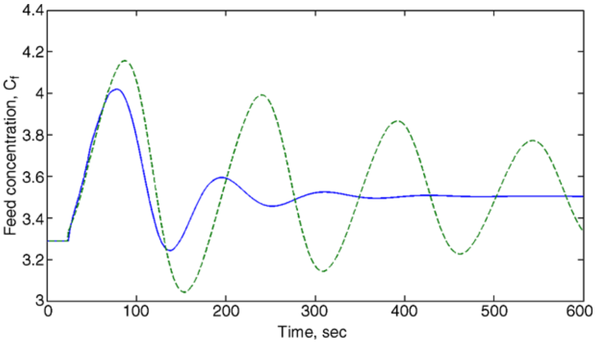


Fig. 3.8 Regulatory response of isothermal CSTR for +20% uncertainty in time delay, *solid*—present method, *dashed*—Huang and Chen [22]

designed for 20 s delay), and the responses are shown in Fig. 3.8. The corresponding manipulated variable responses are shown in Fig. 3.9. The present method gives more robust performance than that of Huang and Chen [22].

Fig. 3.9 Manipulated variable (feed concentration) response of isothermal CSTR for +20% uncertainty in time delay, *solid*—present method, *dashed*—Huang and Chen [22]



3.3.2.2 Case-2: Unstable SOPTD Systems [52]

The typical unstable SOPTD processes that exist in most of the chemical and biological systems can be represented by any of the following transfer function models:

$$G_p = ke^{-\theta s}/(\tau_1 s + 1)(\tau_2 s - 1),$$

$$G_p = ke^{-\theta s}/(\tau_1 s - 1)(\tau_2 s - 1),$$

$$G_p = ke^{-\theta s}/s(\tau s - 1),$$

$$G_p = k(1 \pm ps)e^{-\theta s}/(\tau_1 s \pm 1)(\tau_2 s - 1).$$

Of all the processes, the one that is difficult to control is the unstable SOPTD process with two RHP poles and an RHP zero. Hence, in the present work, this process is considered for the controller design. If the unstable SOPTD process is of the other type, then the designed controller can be easily extended for those types of processes by neglecting the corresponding terms. For generalization, the process is considered for the design of the controller as

$$G_p = k(1 - ps)e^{-\theta s}/\left(a_1 s^2 + a_2 s + 1\right), \tag{3.23}$$

where $a_1 > 0$, $a_2 < 0$, and the open-loop RHP poles of G_p may be real or complex. The desired closed-loop transfer function is assumed as

$$(y/y_r)_d = \left(\eta_2 s^2 + \eta_1 s + 1\right)(1 - ps)e^{-\theta s}/(\lambda s + 1)^3.$$

According to (3.8), the controller is obtained after simplification as

$$G_c = k_c\left(1 + \frac{1}{\tau_i s} + \tau_d s\right)\frac{\alpha s + 1}{\beta s + 1}, \tag{3.24}$$

where

$$k_c = \eta_1/kh, \qquad \tau_i = \eta_1, \qquad \tau_d = \eta_2/\eta_1, \qquad \alpha = 0.5\theta,$$

$$\beta = x_1/a_1, \qquad h = 3\lambda + \theta - \eta_1 + p,$$

$$x_1 = \left(0.5\theta\lambda^3 - 0.5\theta\eta_2 p\right)/h,$$
$$x_2 = \left(\lambda^3 + 1.5\theta\lambda^2 + \eta_2 p + 0.5\theta\eta_2 - 0.5\theta\eta_1 p\right)/h,$$
$$x_3 = \left(3\lambda^2 + 1.5\theta\lambda - \eta_2 + \eta_1 p + 0.5\theta\eta_1 - 0.5p\theta\right)/h,$$
$$\eta_1 = (y_2 z_2 - z_1 y_4)/(y_2 y_3 - y_1 y_4),$$
$$\eta_2 = (y_3 z_1 - z_2 y_1)/(y_2 y_3 - y_1 y_4),$$
$$y_1 = -a_1^2 + 0.5\theta p a_1, \qquad y_2 = -0.5\theta p a_2 - a_1 p - 0.5\theta a_1,$$
$$y_3 = -a_1 a_2 - a_1 p - 0.5\theta a_1, \qquad y_4 = -0.5\theta p + a_1,$$
$$z_1 = a_1\lambda^3 + 1.5\theta a_1\lambda^2 - 0.5\theta a_2\lambda^3 - 3\lambda a_1^2 - p a_1^2,$$
$$z_2 = 3a_1\lambda^2 + 1.5\theta a_1\lambda - 0.5p\theta a_1 - 0.5\theta\lambda^3 - 3a_1 a_2\lambda - a_1 a_2\theta - a_1 a_2 p.$$

Note Based on many simulations, for the unstable SOPTD processes without any zero, the value of β is considered directly as '0.1β', and for the unstable SOPTD processes with a zero, the value of β obtained directly from the controller design procedure is retained.

Selection of the Tuning Parameter It is well known that there is always a trade off in selecting the desired closed-loop tuning parameter (λ). For stable systems fast speed of response and good disturbance rejection are favored by choosing a small value of λ; however, stability and robustness are favored by a large value of λ. This rule is not always true for unstable systems and hence, proper analysis based on performance specifications and operator experience is required. In fact, this is the case for any controller design method based on the closed-loop tuning parameter such as the IMC method, direct synthesis method, etc. Based on many simulation studies, it is observed that the starting value of λ can be considered to be slightly greater than the process time delay, i.e., λ can be considered as 1.2 times the time delay. If both the nominal and robust control performances are achieved with this value, then this value for λ can be taken as final value. If not, the value should be increased carefully until both the nominal and robust control performances are achieved.

Simulation Application to a Model of CSTR with Autocatalytic Reaction Consider an isothermal continuous stirred tank reactor (CSTR) carrying out an autocatalytic reaction $A + 2B \rightarrow 3B$ with a deactivation reaction $B \rightarrow C$. The transient equations for the reactor are given by

$$\frac{dc_A}{dt} = \left(\frac{F}{V}\right)(c_{Af} - c_A) - k_1 c_A c_B^2,$$
$$\frac{dc_B}{dt} = \left(\frac{F}{V}\right)(c_{Bf} - c_B) - k_1 c_A c_B^2 - k_2 c_B,$$

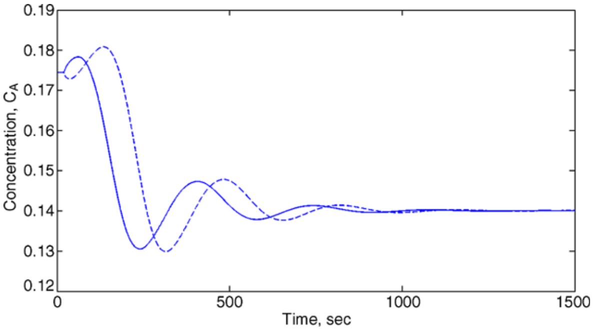


Fig. 3.10 Responses for a step change in the concentration from 0.1744 to 0.140 for CSTR with autocatalytic reaction problem, *solid*—with set-point, *dashed*—without set-point weighting

where c_A is the concentration of the reactant, c_{Af} is the feed concentration, c_B is the concentration of the intermediate, c_{Bf} is the initial concentration of the intermediate, F is the feed flow rate, V is the volume of the reactor, and k_1 and k_2 are the kinetic parameters. The parameters of the CSTR model are given as $k_1 = 1\ \mathrm{l}^2/\mathrm{mol}^2 \cdot \mathrm{s}$, $k_2 = 1/50\ \mathrm{s}^{-1}$, $F/V = 1/250$ s, $c_{Af} = 1$ gmol/l, and $c_{Bf} = 0$. At steady state, the concentrations obtained are $c_A = 0.1744$ mol/l and $c_B = 0.1376$ mol/l. Here c_A is the controlled variable, and F is the manipulated variable. On linearization of the model equations around this steady state operating condition gives the transfer function model as (3.23) with $k_p = -0.2679$, $p = 41.6667$, $a_1 = 279.03$, and $a_2 = -2.9781$. A measurement time delay of $\theta = 20$ s is considered. One can also consider higher values of the time delay. The open-loop poles of the unstable SOPTD process are $0.0053 \pm j0.0596$. The proposed method gives the controller parameters $k_c = 0.868$, $\tau_i = -41.439$, $\tau_d = 1.846$, $\alpha = 10$, and $\beta = 3.78$. Simulation studies are carried out for the original nonlinear system of equations of the CSTR. Figure 3.10 shows the responses for a step change in the concentration from $c_A = 0.1744$ to 0.140 with and without the set-point weighting. The improvement due to the set-point weighting in the responses can be clearly observed.

3.4 PI/PID Controller Design Based on IMC Method

Internal Model Control (IMC) is an established method for the controller synthesis. In this section, a method of tuning PID controllers (i) for an unstable FOPTD system with a positive zero, (ii) for an unstable FOPTD system with a negative zero, and (iii) for an unstable SOPTD system is described. Designing a PID controller by the synthesis and IMC methods have some similarities. For the unstable transfer function models, the value of numerator time constant of IMC controller (numerator time constant of closed-loop transfer function model in case of synthesis method) is selected in such a way that the unstable poles of the process which appear in the numerator of the controller transfer function get canceled by some of the factors of the denominator of the controller transfer function. This step, along with the Pade approximation for the time delay, avoids the use of Maclaurin series expansion to get PID controller parameters. The block diagram of the IMC scheme is shown

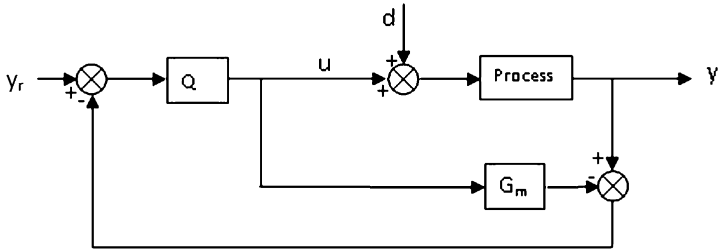


Fig. 3.11 IMC control scheme

in Fig. 3.11, where Q is the IMC controller $[Q = G_{m+}^{-1} f]$ in which G_{m+}^{-1} is the invertible portion of the process model G_m, and f is the IMC filter. Figure 3.11 can be converted into a single-loop feedback control scheme as shown in Fig. 3.1 with the equivalent single-loop feedback controller $G_c = Q/(1 - QG_m)$ [68].

3.4.1 Unstable FOPTD System with a Positive Zero

The process transfer function of unstable FOPTD system with a positive zero is considered as

$$G_p = k(1 - ps)e^{-\theta s}/(\tau s - 1). \tag{3.25}$$

Using the Pade approximation for the time delay, the process transfer function can be written as $G_p = k(1 - ps)(1 - 0.5\theta s)/[(\tau s - 1)(1 + 0.5\theta s)]$, and the IMC controller is obtained as

$$Q = \left(\frac{(\tau s - 1)(1 + 0.5\theta s)}{k}\right)\left(\frac{\eta s + 1}{(\lambda s + 1)^3}\right).$$

Upon simplification, the equivalent conventional controller is obtained as a PID controller with a lag filter (3.22) as

$$k_c = \frac{(\eta + 0.5\theta)}{-k(3\lambda + p + 0.5\theta - \eta)}, \qquad \tau_i = \eta + 0.5\theta,$$

$$\tau_d = \frac{0.5\eta\theta}{\eta + 0.5\theta}, \qquad \beta = \frac{0.5\theta p\eta - \lambda^3}{\tau(3\lambda + p + 0.5\theta - \eta)},$$

where $\eta = [\lambda^3 + (3\lambda + p + 0.5\theta)\tau^2 + \tau(3\lambda^2 - 0.5p\theta)]/[0.5p\theta + \tau^2 - p\tau - 0.5\theta\tau^2]$. Note that the design of controller by the synthesis method gives a PID controller with a second-order filter which is complex when compared to the IMC method of design.

3.4.2 Unstable FOPTD System with a Negative Zero

The process transfer function of unstable FOPTD system with a negative zero is considered as

$$G_p = k(1 + ps)e^{-\theta s}/(\tau s - 1). \tag{3.26}$$

Using the Pade approximation for the time delay, the process transfer function can be written as $G_p = k(1 + ps)(1 - 0.5\theta s)/[(\tau s - 1)(1 + 0.5\theta s)]$. Based on this transfer function model, an IMC controller is designed. After obtaining the IMC controller, the equivalent conventional controller is obtained as a PID controller with a first-order filter (3.22) with the controller parameters as

$$k_c = \frac{(\eta + 0.5\theta)}{-k(2\lambda + 0.5\theta - \eta)}, \qquad \tau_i = \eta + 0.5\theta, \qquad \tau_d = \frac{0.5\eta\theta}{\eta + 0.5\theta}, \qquad \beta = p,$$

where $\eta = [\lambda^2 + 2\lambda\tau + 0.5\theta\tau]/[\tau - 0.5\theta]$. Note that the design of controller by the synthesis method also gives a PID controller with a first-order filter.

3.4.3 Unstable SOPTD System with a Positive Zero

The process transfer function of an unstable SOPTD system with a positive zero is considered as

$$G_p = k(1 - ps)e^{-\theta s}/\bigl(a_1 s^2 + a_2 s + 1\bigr). \tag{3.27}$$

Using the Pade approximation for the time delay, the process transfer function can be written as $G_p = k(1 - ps)(1 - 0.5\theta s)/[(a_1 s^2 + a_2 s + 1)(1 + 0.5\theta s)]$, and with this the IMC controller is obtained. After simplification, the equivalent conventional controller is obtained as a PID controller with a lead lag filter of the form

$$G_c = k_c\left(1 + \frac{1}{\tau_i s} + \tau_d s\right)\frac{(\alpha s + 1)}{(\beta_1 s^2 + \beta_2 s + 1)}, \tag{3.28}$$

where

$$\begin{aligned}
&k_c = z_2/k(5\lambda + p + 0.5\theta - z_2), \qquad \tau_i = z_2, \qquad \tau_d = z_1/z_2, \qquad \alpha = 0.5\theta,\\
&\beta_1 = \lambda^5/a_1(5\lambda + p + 0.5\theta - z_2),\\
&\beta_2 = \bigl[\bigl(10\lambda^2 - 0.5p\theta - z_1 + pz_2 + 0.5z_2\theta\bigr)/(5\lambda + p + 0.5\theta - z_2)\bigr] - a_2 z_2\\
&\quad\;\; = (x_3x_4 - x_1x_6)/(x_4x_2 - x_1x_5),\\
&z_1 = (x_3 - x_2z_2)/x_1,\\
&x_1 = 0.5p\theta a_1, \qquad x_2 = pa_1^2 + 0.5\theta a_1^2 + a_2a_1^2,
\end{aligned}$$

$$
\begin{aligned}
x_3 &= 5a_1\lambda^4 - a_2\lambda^5 - 5\lambda a_1^2 - pa_1^2 - 0.5\theta a_1^2 + 5a_2\lambda a_1^2 + 5p\lambda + 2.5\lambda\theta,\\
x_4 &= pa_1 + 0.5\theta a_1 + a_1 a_2,\\
x_5 &= a_1^2 - a_1 a_2^2 - pa_1 a_2 - 0.5\theta a_1 a_2 - 0.5p\theta a_1,\\
x_6 &= \lambda^5 + 10a_1 a_2\lambda^2 - 0.5pa_1 a_2\theta - 5a_1 a_2^2\lambda - pa_1 a_2^2 - 0.5a_1 a_2^2\theta - 10a_1\lambda^3\\
&\quad + 10a_1^2\lambda^2 - 0.5p\theta a_1^2.
\end{aligned}
$$

Design of controller by synthesis method gives a PID controller with first-order filter as discussed earlier. Note that the role of the filter in the IMC method and desired closed-loop transfer function in the direct synthesis method is similar for obtaining the final form of the controller.

3.5 Design Based on Equating Coefficient Method

This method is based on matching the corresponding coefficients of s, s^2, s^3 in the numerator and denominator of the closed-loop transfer function for a servo problem. It is a simple and effective tuning method [68].

3.5.1 Design for Integrating Systems [11]

Here, a method of designing PI, PD, and PID controllers for integrating systems with a time delay is addressed. To improve the performance of the controller under uncertainty in model parameters, the concept of two tuning parameters is used. The robustness of the controller is evaluated by using Kharitonov's theorem.

3.5.1.1 Case-1: Integrating Time Delay Systems [11]

An integrator with time delay system is represented by $ke^{-\theta s}/s$. The closed-loop transfer function relating the output (y) to the set point (y_r) is given by

$$
\frac{y(\tilde{s})}{y_r(\tilde{s})} = \frac{(k_1\tilde{s} + k_2 + k_3\tilde{s}^2)e^{-\tilde{s}}}{[\tilde{s}^2 + (k_1\tilde{s} + k_2 + k_3\tilde{s}^2)e^{-\tilde{s}}]}, \tag{3.29}
$$

where $k_1 = k_c k\theta$, $k_2 = k_1/(\tau_i/\theta)$, $k_3 = k_1(\tau_d/\theta)$, $\tilde{s} = s\theta$. Here s is the Laplace operator. The exponent $e^{-\tilde{s}}$ in the numerator is removed for further study, since this term only shifts the corresponding time axis and using Pade's approximation for $e^{-\tilde{s}}$ as $[(1-0.5\tilde{s})/(1+0.5\tilde{s})]$ in the denominator, the corresponding coefficients of $\tilde{s}$, $\tilde{s}^2$, and $\tilde{s}^3$ of the numerator with that of the denominator are equated. Since the presence of the integral mode makes the offset zero, the constant term in the numerator and that in the denominator are the same. By equating the corresponding coefficient of

$\tilde{s}$, $\tilde{s}^2$, and $\tilde{s}^3$ of the numerator with that of the denominator, one can obtain $k_2 = 0$, $k_1 = 1$, $k_3 = 0.5$. After simplifying, the controller settings are obtained as

$$k_c k\theta = 1, \qquad \tau_i/\theta = \infty, \qquad \tau_d/\theta = 0.5.$$

This is a PD controller. It is interesting to note that Visioli [77] by minimizing the integral squared error (ISE) for a servo problem using genetic algorithm found that a PD controller results and the settings are also given by the above equations.

Since with a PI or PID controller, the closed-loop response shows some overshoot for the servo response, the value of $y(\tilde{s})/y_r(\tilde{s})$ can be allowed to be more than 1. If $\tilde{s} = 0$, y is automatically equal to y_r because of the presence of the integral action. Therefore, the corresponding coefficients of $\tilde{s}$ of the numerator are equated to α times that of the denominator. Here, the value of α is greater than one, and this parameter is considered as a tuning parameter. The following set of linear algebraic equations is obtained:

$$\begin{gathered}(1-\alpha)k_1 + 0.5(1+\alpha)k_2 = 0,\\ 0.5(1+\alpha)k_1 + (1-\alpha)k_3 = \alpha,\\ (1+\alpha)k_3 = \alpha.\end{gathered}$$

For $\alpha = 1$, as stated earlier, it results in a PD controller. By solving the above equations the PID controller parameters are obtained as

$$\begin{aligned}&k_c k\theta = 4\alpha^2/(1+\alpha)^2, \qquad \tau_i/\theta = 0.5(1+\alpha)/(\alpha-1),\\ &\tau_d/\theta = 0.25(1+\alpha)/\alpha.\end{aligned}$$

Similarly, by considering only PI controller mode, the following equations for the PI controller settings are obtained.

$$k_c k\theta = 2\alpha/(1+\alpha), \qquad \tau_i/\theta = 0.5(1+\alpha)/(\alpha-1),$$

where α is a tuning parameter and should be greater than 1. Similarly to the IMC method, care should be taken in selecting this tuning parameter. Assuming that $\alpha = 1.25$, the following relations for PID controller are obtained:

$$k_c k\theta = 1.2346, \qquad \tau_i/\theta = 4.5, \qquad \tau_d/\theta = 0.45.$$

However, the controller gives an oscillatory response when the uncertainty in the delay is +33%. Therefore, the present method is extended using two tuning parameters α_1 and α_2. Writing $e^{-\tilde{s}}$ in the denominator of (3.29) as $e^{-0.5\tilde{s}}/e^{0.5\tilde{s}}$, the following equation for the closed-loop transfer function is obtained:

$$\frac{y(\tilde{s})}{y_r(\tilde{s})} = \frac{(k_1\tilde{s} + k_2 + k_3\tilde{s}^2)e^{-0.5\tilde{s}}e^{-\tilde{s}}}{[e^{0.5\tilde{s}}\tilde{s}^2 + (k_1\tilde{s} + k_2 + k_3\tilde{s}^2)e^{-0.5\tilde{s}}]}. \tag{3.30}$$

The numerator and the denominator terms using the Taylor series expansion (up to four terms) for $e^{0.5\tilde{s}}$ and $e^{-0.5\tilde{s}}$ are considered. The coefficient of $\tilde{s}$ in the numerator is equated α_1 times to that of denominator of the closed-loop transfer function. The coefficients of $\tilde{s}^2$ and $\tilde{s}^3$ of the numerator are equated to α_2 times to those of the denominator. It has been found by simulation on various transfer function models that $\alpha_2 = 0.6\alpha_1$ and $\alpha_1 = 1.5$ gives best results. Thereby the present method ultimately has no tuning parameters. Solving the equations, k_1, k_2, and k_3 are obtained. Using the definitions of k_1, k_2, and k_3, PID controller settings are obtained as

$$k_c k\theta = 0.8956, \qquad \tau_i/\theta = 2.5, \qquad \tau_d/\theta = 0.5.$$

3.5.1.2 Case-2: Stable FOPTD Systems with an Integrator [67]

A stable FOPTD system with an integrator is represented by $ke^{-\theta s}/[s(\tau s+1)]$. The closed-loop transfer function relating the output (y) to the set point (y_r) is given by

$$\frac{y(\tilde{s})}{y_r(\tilde{s})} = \frac{(k_1\tilde{s} + k_2 + k_3\tilde{s}^2)e^{-\tilde{s}}}{[\tilde{s}^2[(\frac{\tau}{\theta})\tilde{s} + 1] + (k_1\tilde{s} + k_2 + k_3\tilde{s}^2)e^{-\tilde{s}}]}. \tag{3.31}$$

Adopting the similar procedure discussed in the previous section, the following set of linear algebraic equations is obtained:

$$(1-\alpha_1)k_1 + 0.5(1+\alpha_1)k_2 = 0,$$

$$0.5(1+\alpha_2)k_1 + 0.125(1-\alpha_2)k_2 + (1-\alpha_2)k_3 = \alpha_2,$$

$$0.125(1-\alpha_2)k_1 + 0.0208(1+\alpha_2)k_2 + 0.5(1+\alpha_2)k_3 = \alpha_2\big[0.5 + (\tau/\theta)\big].$$

By solving these linear algebraic equations PID controller settings are obtained.

Robustness Analysis A control system is said to be robust if the closed-loop system is stable even when the model parameters of the actual process are different from those used for the controller design. To compare the robustness of the different controller design methods, the range of uncertainty in each of the model parameters for which the controller is stable is to be calculated. The robustness of the closed-loop system for the perturbation separately in time delay, time constant, and process gain is analyzed theoretically by Kharitonov's method. In this method the stability of four equations formed from Kharitonov polynomials is to be checked. The characteristic equation of the system using second-order Pade's approximation for the time delay is

$$P(s) = a_0 + a_1 s + a_2 s^2 + a_3 s^3 + a_4 s^4,$$

where $a_0 = k_c k$, $a_1 = k_c k(\tau_i - 0.5\theta)$, $a_2 = k_c k(0.0833\theta^2 - 0.5\theta\tau_i + \tau_i\tau_d) + \tau_i$, $a_3 = k_c k(0.0833\tau_i\theta^2 - 0.5\theta\tau_i\tau_d) + 0.5\theta\tau_i$, $a_4 = 0.0833\theta^2(k_c k\tau_i\tau_d + \tau_i)$.

Kharitonov's equations for $a_i^- \le a_i \le a_i^+$ $(i = 0, 1, 2, 3, 4)$ are given below, where a_i^- and a_i^+ are the lower and upper bounds for a_i, respectively:

$$a_0^- + a_1^+ s + a_2^+ s^2 + a_3^- s^3 + a_4^- s^4 = 0,$$

$$a_0^- + a_1^- s + a_2^+ s^2 + a_3^+ s^3 + a_4^- s^4 = 0,$$
$$a_0^+ + a_1^- s + a_2^- s^2 + a_3^+ s^3 + a_4^+ s^4 = 0,$$
$$a_0^+ + a_1^+ s + a_2^- s^2 + a_3^- s^3 + a_4^+ s^4 = 0.$$

For fixed values of k and τ, a perturbation in time delay θ, i.e., when $(\theta - \delta\theta) \le \theta \le (\theta + \delta\theta)$ is substituted in the above coefficients and Kharitonov's equations are checked for stability using the Routh–Hurwitz method. Similarly, perturbation in k (for fixed τ and θ) is evaluated, and the stability ranges are analyzed.

3.5.2 Design for Unstable Systems [69]

In this section, the equating coefficient method is extended to unstable FOPTD systems. This method gives equations for the controller settings in terms of the FOPTD model parameters.

3.5.2.1 Design of PID Controller

A first-order plus time delay system $ke^{-\theta s}/(\tau s \pm 1)$ is considered, with + sign for stable systems and "−" sign for unstable systems. A PID controller is considered here. The closed-loop transfer function relating the output variable (y) to the set point (y_r) can be written as

$$\frac{y(\tilde{s})}{y_r(\tilde{s})} = \frac{(k_1\tilde{s} + k_2 + k_3\tilde{s}^2)e^{-0.5\varepsilon\tilde{s}}e^{-\varepsilon\tilde{s}}}{[(\tilde{s} \pm 1]\tilde{s}e^{-0.5\varepsilon\tilde{s}} + (k_1\tilde{s} + k_2 + k_3\tilde{s}^2)e^{-\varepsilon\tilde{s}}e^{-0.5\varepsilon\tilde{s}}]} \tag{3.32}$$

where $k_1 = k_c k$, $k_2 = k_1/(\tau_i/\tau)$, $k_3 = k_1(\tau_d/\tau)$, $\tilde{s} = s\tau$, $\varepsilon = \theta/\tau$. By equating the corresponding coefficients of $\tilde{s}$, $\tilde{s}^2$, and $\tilde{s}^3$ of the numerator with those of the denominator the following equations are obtained:

$$k_2 = (1/\varepsilon), \qquad k_1 = (1/\varepsilon) + 0.5, \qquad 2k_3 = 1 + (\varepsilon/6).$$

By rearranging all the above three equations, the following simple equations for the PID controller settings in terms of the model parameters are obtained:

$$k_c k = (1/\varepsilon) + 0.5, \qquad \tau_i/\tau = 1 + 0.5\varepsilon, \qquad \tau_d/\tau = 0.5\varepsilon(1 + 0.1667\varepsilon)/(1 + 0.5\varepsilon).$$

After equating each of the numerator term (except that of the coefficient of $\tilde{s}^0$) is made equal to α times that of the corresponding denominator term, a set of linear algebraic equations for the PID controller settings are obtained. Then after several simulation studies, the final tuning rules are obtained as

$$k_c k = 1.4183\varepsilon^{-0.9147} \quad \text{if } 0.01 \le \varepsilon \le 0.9,$$

$$\tau_i/\tau = \begin{cases} 16.327\varepsilon^2 + 5.5778\varepsilon + 0.8158 & \text{if } 0.01 \le \varepsilon < 0.6, \\ 196\varepsilon^2 - 247.28\varepsilon + 87.72 & \text{if } 0.6 \le \varepsilon \le 0.9, \end{cases}$$

$$\tau_d/\tau = 0.4917\varepsilon \quad \text{if } 0.01 \le \varepsilon \le 0.9.$$

3.5.2.2 Design of PI Controller

A PI controller is designed using the method discussed in the above section. Here α_1 is tuned to be greater than 1, and α_2 is selected as $\beta\alpha_1$. The values of the parameters α_1 and β are selected by simulation of the process model with PI controller. The value α_1 increases with the value of ε. The settings are fitted by the following simple equations:

$$k_c k = 0.8624\varepsilon^{-0.9744} \quad \text{if } 0.01 \le \varepsilon \le 0.6,$$

$$\tau_i/\tau = 143.34\varepsilon^3 - 73.912\varepsilon^2 + 19.039\varepsilon - 0.2276 \quad \text{if } 0.01 \le \varepsilon \le 0.6.$$

Here also, one can analyze the robustness by Kharitonov's method as explained earlier.

3.6 Set-Point Weighting for PI/PID Controllers [10]

The performance of a PI/PID controller can be improved particularly to reduce overshoot for the servo problem by introducing a set-point weighting parameter in the PI/PID control law. As explained earlier, for a PID controller, the control law is

$$u(t) = k_c\left[e_p + \left(\frac{1}{\tau_i}\right)\int e\,dt + \tau_d\left(\frac{de_d}{dt}\right)\right],$$

where $e = y_r - y$, $e_p = (\sigma y_r - y)$, $e_d = (\gamma y_r - y)$, with γ and σ the set-point weighting parameters. The value σ is between 0 and 1. For $\sigma = 1$ and $\gamma = 1$, the controller is a conventional PID controller. The actual error $(y_r - y)$ is used for integral calculation to make sure that the offset is zero. If the derivative action is taken on the output variable rather than on the error, then $\gamma = 0$. The closed-loop transfer function of a process G_p with set-point weighted PID controller ($\gamma = 0$) is derived as

$$\frac{y}{y_r} = \frac{(1 + \sigma\tau_i s)G_p}{[\tau_i s + (1 + \tau_i s + \tau_d s^2)G_p]}.$$

In the above equation, $(1 + \sigma\tau_i s)$ is the stable zero introduced by the controller due to which overshoot is observed in the closed-loop response. An expression for overshoot as a function of σ is derived, and then an expression for σ is obtained by minimizing the overshoot analytically for unstable FOPTD systems. Simple equations are given to calculate set-point weighting parameter as a function of $\varepsilon = (\theta/\tau)$.

3.6.1 Case-1: Integrating Time Delay Systems

Here, method of selection of set-point weighting parameter is described with PI and PID controllers.

With PI Controller The closed-loop transfer function relating y to y_r of integrating dead-time process with a PI controller is given by [10]

$$\frac{y}{y_r} = \frac{k_c k(\sigma + \frac{1}{\tau_i s})[\frac{e^{-\theta s}}{s}]}{1 + k_c k(\sigma + \frac{1}{\tau_i s})[\frac{e^{-\theta s}}{s}]}. \tag{3.33}$$

Using the approximation $e^{-\theta s} = 1 - \theta s$ in the denominator, it becomes

$$\frac{y}{y_r} = \frac{(\sigma \tau_i s + 1)e^{-\theta s}}{[\tau_e^2 s^2 + 2\tau_e \zeta s + 1]}, \tag{3.34}$$

where $\tau_e^2 = \tau_i[1 - k_c k\theta]/(k_c k)$ and $2\tau_e \zeta = \tau_i - \theta$. By minimizing the overshoot in y, the value of σ is obtained as [10] $\sigma = \zeta \tau_e / \tau_i$, and after simplification, the equation for σ is obtained as $\sigma = 0.5[1 - (\theta/\tau_i)]$. The value of σ varies from 0.3632 to 0.44, depending on the tuning method. Hence, an average value of 0.4 may be recommended for integrator dead-time processes.

With PID Controller For PID controller [for $\gamma = 0$, i.e., the derivative is taken on output and not on the error], the numerator of the closed-loop transfer function is the same as that of (3.33). The expression for the set-point weighting parameter is recommended as

$$\sigma = \zeta \tau_e^2 / \left(1 - \zeta^2\right)^{0.5},$$

where τ_e and ζ are the effective time constant and damping coefficient of the characteristic equation based on the two dominant roots of the denominator of the closed-loop system $\tau_e^2 s^2 + 2\tau_e \zeta s + 1 = 0$. It is recommended that for the Ziegler–Nichols tuning method, whether PI or PID, σ can be considered as 0.35 for integrating dead-time processes.

3.6.2 Case-2: Unstable FOPTD Systems [66]

A PI controller is considered for the unstable system $ke^{-\theta s}/(\tau s - 1)$. The transfer function relating y to y_r is obtained as explained earlier [10]. Using the approximation $e^{-\theta s} = 1 - \theta s$ in the denominator, the closed-loop relation is obtained as

$$\frac{y}{y_r} = \frac{(\sigma \tau_i s + 1)e^{-\theta s}}{[\tau_e^2 s^2 + 2\tau_e \zeta s + 1]}, \tag{3.35}$$

where $\tau_e^2 = \tau_i[\tau - k_c k\theta]/(k_c k)$ and $2\tau_e\zeta = [-\tau_i + k_c k(\tau_i - \theta)]/(k_c k)$. The step response for unit step change in set point is evaluated, minimization of the first overshoot in y is carried out, and the set-point weighting parameter is obtained. The final expressions for a PID controller and PI controller are given below:

$$\sigma = \begin{cases} 0.26 - 0.42\varepsilon + 0.23\varepsilon^2 & \text{for PID controller,} \\ 0.267 - 0.6\varepsilon + 0.36\varepsilon^2 & \text{for PI controller.} \end{cases}$$

Further, Sree and Chidambaram [65] have proposed a simple method of calculating the set-point weighting parameter for unstable systems with a zero by equating the coefficient of s in the closed-loop transfer function model relating y to y_r.

3.7 Introduction to Two-Degree-of-Freedom Control Schemes

In order to further enhance the closed-loop performance, two-degree-of-freedom control schemes are proposed. There are several structures which are called as two-degree-of-freedom control schemes. Here, two-degree-of-freedom controllers based on modified Smith predictor schemes are explained.

3.7.1 Case-1: Integrating Systems [54]

The modified form of the Smith predictor is shown in Fig. 3.12, where G_p is the transfer function of the integrating process, G_m is the transfer function of the integrating process model without time delay, and θ is the time delay of the process model. G_{cs} is the transfer function of the set-point tracking controller, and G_{cd} is the transfer function of the load disturbance rejection controller. The closed-loop transfer functions between the output and the set point and the input disturbance (y_{di}) assuming that the model describes the process exactly (i.e., $G_p = G_m e^{-\theta s}$) are given respectively as

$$\frac{y}{y_r} = \frac{G_{cs} G_m e^{-\theta s}}{1 + G_{cs} G_m}, \tag{3.36}$$

$$\frac{y}{y_{di}} = \frac{(1 + G_{cs} G_m - G_{cs} G_m e^{-\theta s}) G_m e^{-\theta s}}{(1 + G_{cs} G_m)(1 + G_{cd} G_m e^{-\theta s})}. \tag{3.37}$$

It can be observed that the set-point responses are decoupled from the load disturbance rejection responses. From (3.36) it can be observed that the set-point tracking depends on the controller G_{cs}. However, the load disturbance rejection depends on both the controllers. Once G_{cs} is designed for set-point tracking, G_{cd} can be designed for load disturbance rejection. From (3.36) it can be observed that the controller G_{cs} introduces a zero in the y/y_r expression, and this zero introduces an undesirable overshoot in the closed-loop servo response. To reduce the overshoot, either a set point filter or set point weighting is considered.

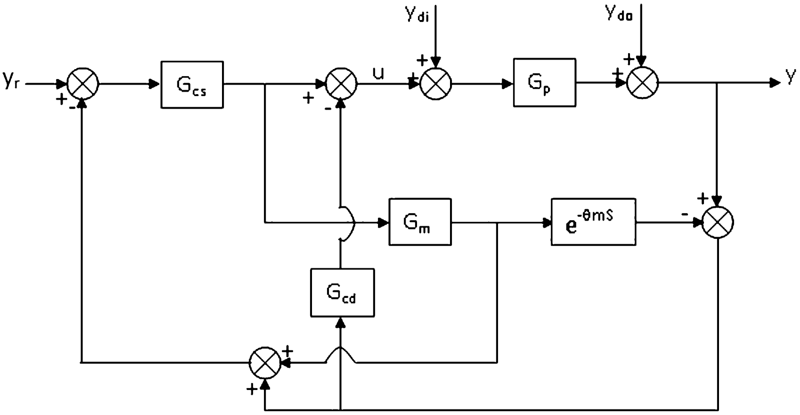


Fig. 3.12 Modified Smith predictor control structure for integrating systems [54]

Controllers Design Here the controller design is given only for one integrating process, and the design for other types of integrating processes can be seen in [54]. The process is considered as $G_p = G_m e^{-\theta s} = k_m e^{-\theta s}/s(\tau_m s + 1)$. The direct synthesis method is used for design of set-point tracking controller and is obtained as

$$G_c = k_c\left(1 + \frac{1}{\tau_i s} + \tau_d s\right), \tag{3.38}$$

where $k_c = \frac{3\tau_m}{k_m \lambda^2}$, $\tau_i = 3\lambda$, $\tau_d = \lambda(1 - \frac{\lambda}{3\tau_m})$.

G_{cd} is chosen as a PD controller in the form $k_d + \tilde{k}_{rd} s$, where k_d is the proportional term, and $\tilde{k}_{rd}$ is the derivative term ($\tilde{k}_{rd} = k_d \tau_d$ when compared with conventional PD controller structure $k_d(1 + \tau_d s)$). G_{cd} is designed based on gain and phase margin criteria. Let us consider the integrating process model with time delay as $\tilde{G}_M = G_m e^{-\theta s}$. The characteristic equation involving G_{cd} is given by $1 + G_{cd}\tilde{G}_M$. The open-loop transfer function is $F(s) = G_{cd}(s)\tilde{G}_M(s)$. From the definitions of gain margin and phase margin, the relations for magnitude and phase angle are given by

$$\arg\left[G_{cd}(j\omega_p)\tilde{G}_M(j\omega_p)\right] = -\pi,$$

$$\left|G_{cd}(j\omega_g)\tilde{G}_M(j\omega_g)\right| = 1,$$

$$A_m = \frac{1}{|G_c(j\omega_p)\tilde{G}_M(j\omega_p)|},$$

$$\Phi_m = \pi + \arg\left[G_{cd}(j\omega_g)\tilde{G}_M(j\omega_g)\right],$$

where ω_p and ω_g are the phase and gain crossover frequencies, and A_m and Φ_m are the gain margin and phase margin, respectively. For a specified gain and phase margin, the above equations can be solved for the controller parameters k_d and $\tilde{k}_{rd}$.

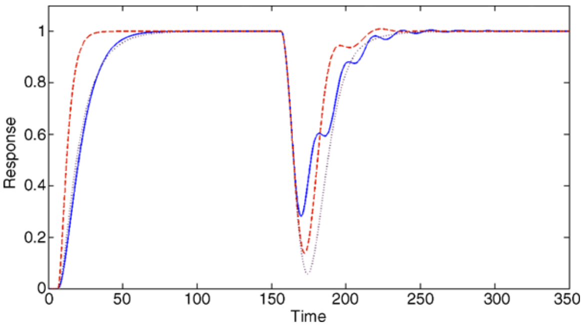


Fig. 3.13 Responses for perfect model for the process $G_p = e^{-6.567s}/s(3.4945s + 1)$, *solid*—proposed, *dashed*—Liu et al. [35], *dotted*—Kaya [25]

Simulation Study Consider the integrating process as

$$G_p(s) = \frac{e^{-6.567s}}{s(3.4945s + 1)}.$$

The set point controller (G_{cs}) parameters are obtained as $k_c = 0.124$, $\tau_i = 27.58$, $\tau_d = 1.130$ with the tuning parameter selected as $\lambda = 1.4\theta_m = 7.3542$. The set-point weighting parameter is chosen as $\sigma = 0.4$. G_{cd} is designed as $0.035 + 1.12s$ by considering the gain margin and phase margin as 1.7 and 60, respectively. Using these controller settings, the performance is evaluated by giving unit step input in the set point and a negative step input of 0.1 in the load (y_{di}) at $t = 150$ s, respectively. To show the improvement, the methods proposed recently by Kaya [25] and Liu et al. [35] are considered. Figure 3.13 shows the responses for perfect model. It can be observed that the proposed method gives better control performances for the load disturbance rejection. A perturbation of +30% in process time delay and the time constant is considered, and the corresponding responses are shown in Fig. 3.14. Figure 3.15 shows the corresponding control action responses. It is clear that the proposed method performs significantly better.

3.7.2 *Case-2: Unstable Systems [53]*

The block diagram of the modified Smith predictor scheme is shown in Fig. 3.16, where $\tilde{G}_p$ is the transfer function of the unstable plant, θ_p is the plant time delay, $\tilde{G}_m$ is the transfer function of the plant model, and θ_m is the model time delay. G_{cs} is the set-point tracking controller and is designed using the direct synthesis method. G_{cd} is meant for stabilization of the unstable process and is the disturbance rejection controller designed using the modified H_∞ method. A first-order filter (G_f) is used to improve the robustness for the predicted disturbance. The closed-loop transfer functions for servo and regulatory problems assuming that the model represents exactly the process $(\tilde{G}_p e^{-\theta_p s} = \tilde{G}_m e^{-\theta_m s})$ are

$$\frac{y}{y_r} = \frac{G_{cs}\tilde{G}_m e^{-\theta_m s}}{1 + G_{cs}\tilde{G}_m}, \tag{3.39}$$

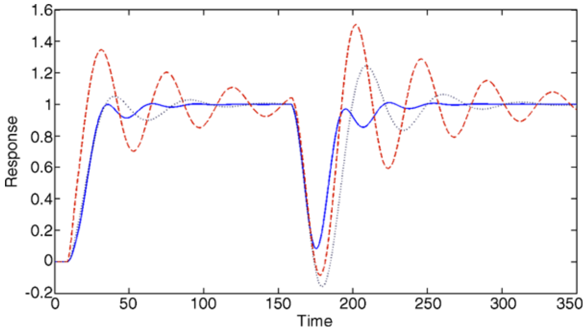


Fig. 3.14 Responses for perturbations of +30% in process time delay and time constant for the process $G_p = e^{-6.567s}/s(3.4945s + 1)$, *solid*—proposed, *dashed*—Liu et al. [35], *dotted*—Kaya [25]

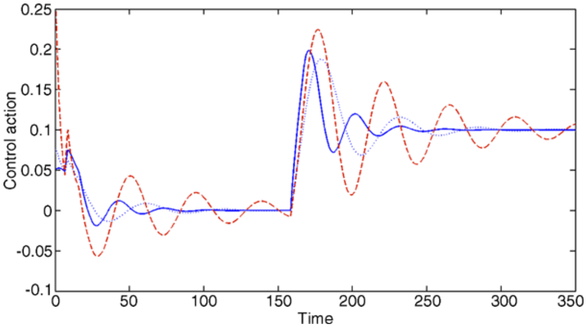


Fig. 3.15 Control action responses for perturbations of +30% in process time delay and time constant for the process $G_p = e^{-6.567s}/s(3.4945s + 1)$, *solid*—proposed, *dashed*—Liu et al. [35], *dotted*—Kaya [25]

$$\frac{y}{y_d} = \frac{(1 + G_{cs}\tilde{G}_m - G_{cs}G_f\tilde{G}_m e^{-\theta_m s})G_m e^{-\theta_m s}}{(1 + G_{cs}\tilde{G}_m)(1 + 1 + G_{cd}\tilde{G}_m e^{-\theta_m s})}. \tag{3.40}$$

Set-point weighting is considered to reduce the undesirable overshoot, as it is simple and gives an improved response.

Controllers Design An unstable FOPTD process is considered whose model is $k_m e^{\theta_m s}/(\tau_m s - 1)$. The set-point tracking controller is obtained as

$$G_{cs} = k_c\left(1 + \frac{1}{\tau_i s}\right),$$

where $k_c = (\lambda + 2\tau_m)/k\lambda$, $\tau_i = (\lambda^2 + 2\lambda\tau_m)/\tau_m$, and the recommended range for λ as $0.7\theta_m - 2\theta_m$. The disturbance rejection controller is obtained as

$$G_{cd} = k_d + \left(\frac{0.7\tau}{k}\right)s,$$

where

$$k_d = \frac{1}{k_m}\left(\frac{0.533}{\theta_m/\tau_m} + 0.746\right) \quad \text{if } \theta_m/\tau_m \le 0.7,$$

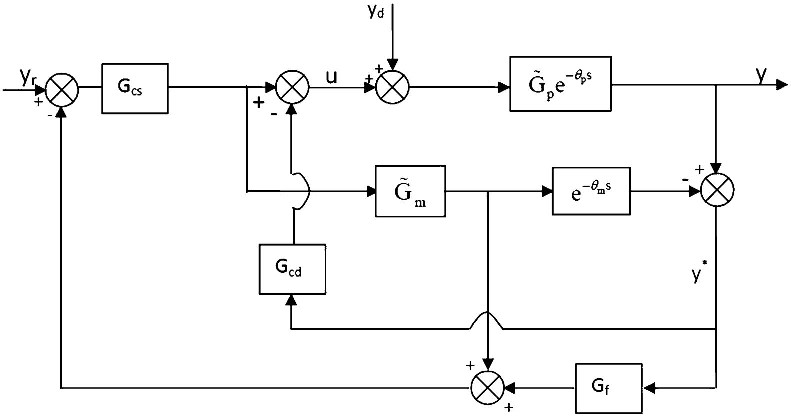


Fig. 3.16 Modified Smith predictor structure for unstable processes [53]

$$k_d = \frac{1}{k_m}\left(\frac{0.49}{\theta_m/\tau_m} + 0.694\right) \quad \text{if } 0.7 < \theta_m/\tau_m \le 1.5.$$

The filter is selected as a first-order one and is $G_f = 1/(\tau_f s + 1)$, where the filter constant τ_f is given as $1.4\theta_m$. The range of set-point weighting parameter is given as 0.4 to 0.6.

Simulation Study An isothermal chemical reactor exhibiting multiple steady-state solutions is considered. The mathematical model is given in (3.1). Based on this model, the controllers G_{cs} and G_{cd} are designed, and the controllers parameters are obtained as $k_c = 2.794$, $\tau_i = 53.58$, $k_d = 1.017$, and $k_d\tau_d = 30.032$, respectively. The closed-loop tuning parameter is selected as $\lambda = 1.2\theta_m$. The set-point weighting parameter is chosen as $\sigma = 0.4$, and the filter time constant is calculated as $\tau_f = 1.4\theta_m = 28$. With these controller settings, the nonlinear system in the closed loop is simulated for a step change from the steady-state value 1.316 to 5, and simultaneously a step disturbance from 0.0333 to 0.04 is introduced at $t = 1000$ s, respectively. The closed-loop response is shown in Fig. 3.17. A perturbation of 30% is considered in the process time delay, and the corresponding responses are also shown in Fig. 3.17. The responses corresponding to perturbations in the process time delay coincide with the response obtained under nominal conditions. It can be observed from the responses that the proposed method gives good control performances. However, all linear controllers work over a limited range of nonlinearity. If the nonlinearity is severe, the controllers need to be retuned frequently corresponding to the nearest operating point.

3.8 Conclusions and Future Perspectives

In this chapter, several PI/PID tuning rules for integrating and unstable systems are reviewed. Methods of designing controllers based on direct synthesis, IMC, and

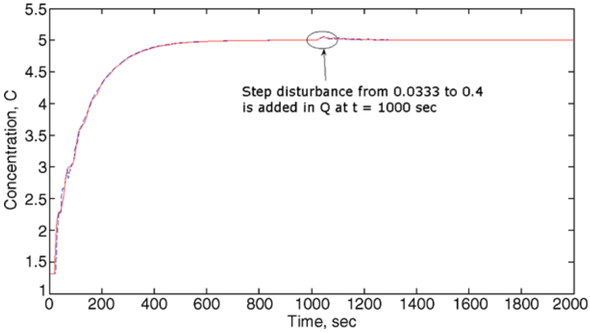


Fig. 3.17 Responses for a step change from 1.316 to 5 at $t = 0$ in the set point and a step change from 0.0333 to 0.4 in the disturbance (Q) at $t = 1000$ s for the isothermal CSTR, *solid*—perfect model, *dashed*—+30% perturbation in the process time delay

equating coefficients are discussed in detail. Further, two degree-of-freedom control schemes have been explained. Some methods are simple, some methods are moderate, and some methods are complex in terms of the mathematical expressions and in terms of the theory of the method. Most of the methods are based on some tuning parameters. Even though guidelines are provided for proper selection of the tuning parameter, in practice, some trials are required to come to the final value of the tuning parameter. Set-point weighting should be considered when dealing with particularly unstable processes to reduce the overshoot. Robustness is analyzed by considering small gain theorem or using Kharitonov's theorem. It can be observed that for integrating systems, the methods perform well for higher values of time delay. However, the methods have a limitation for unstable systems if the time delay is significant.

The conventional PI/PID controllers cannot provide good closed-loop performances for delay dominant unstable processes. If the ratio of time delay to unstable time constant(ε) is ≥ 1.8, the method does not stabilize the unstable process. For all model-based control schemes with a tuning parameter, care should be taken in properly selecting the parameter because, unlike for stable systems, there will be lower and upper bounds for controller gain for unstable systems.

References

1. Arbogast, J.E., Cooper, D.J.: Extension of IMC tuning correlations for non-self regulating (integrating) processes. ISA Trans. **46**, 303–311 (2007)
2. Arbel, A., Rinard, I.H., Shinar, R., et al.: Dynamics and control of fluidized catalytic crackers. 2. Multiple steady states and instabilities. Ind. Eng. Chem. Res. **34**, 3014–3026 (1995)
3. Agarwal, P., Lim, H.C.: Analysis of various control schemes for continuous bioreactors. Adv. Biochem. Biotechnol. **30**, 61–90 (1989)
4. Ali, E., Al-humaizi, K.: Temperature control of ethylene to butene-1 dimerization reactor. Ind. Eng. Chem. Res. **39**, 1320–1329 (2000)
5. Ali, A., Majhi, S.: PID controller tuning for integrating processes. ISA Trans. **49**(1), 70–78 (2010)
6. Bequette, B.W.: Process Control: Modeling, Design and Simulation. Prentice Hall, New York (2003)
7. Chen, C.C., Huang, H.P., Liaw, H.J.: Set point weighted PID controller tuning for time delayed unstable processes. Ind. Eng. Chem. Res. **47**, 6983–6990 (2008)

8. Chidambaram, M.: Control of unstable systems: a review. J. Energy Heat Mass Transf. **19**, 49–57 (1997)
9. Chidambaram, M.: Periodic operation of isothermal plug flow reactors for autocatalytic reactions. Chem. Eng. Commun. **69**, 219–228 (1988)
10. Chidambaram, M.: Set point weighting PI/PID controller for integrating plus dead time processes. In: Proc. National Symp. on Intell. Measurement Contrl, Chennai, pp. 324–331 (2000)
11. Chidambaram, M., Sree, R.P.: A simple method of tuning PID controllers for integrator/dead time processes. Comput. Chem. Eng. **27**, 211–215 (2003)
12. Chien, I.L., Fruehauf, P.S.: Consider IMC tuning to improve performance. Chem. Eng. Prog. **10**, 33–41 (1990)
13. Chiu, T., Christofides, P.D.: Nonlinear control of particulate processes. AIChE J. **45**, 1279–1297 (1999)
14. Clarke, R., Burken, J.J., Bosworth, J.T., et al.: X-29 flight control system: Lessons learned. Int. J. Control **59**(1), 199–219 (1994)
15. Eriksson, L., Oksanen, T., Mikkola, K.: PID controller tuning rules for integrating processes with varying time delays. J. Franklin Inst. **346**(5), 470–487 (2009)
16. Filatov, N.M., Keuchel, U., Unbehauen, H.: Dual control of unstable mechanical plant. IEEE Trans. Control Syst. Technol. **16**(4), 31–37 (1996)
17. Foley, M.W., Julien, R.H., Copeland, B.R.: Proportional-integral-derivative tuning for integrating processes with deadtime. IEE Proc., Control Theory Appl. **4**(3), 425–436 (2010)
18. Govindhakannan, J., Chidambaram, M.: Multivariable PI control of unstable systems. Process Control Qual. **10**, 329–339 (1997)
19. Govindhakannan, J., Chidambaram, M.: Two stage multivariable PID controllers for unstable plus time delay systems. J. Indian Chem. Eng. **42**, 89–93 (2000)
20. Hang, C.C., Wang, Q.G., Yang, X.P.: A modified Smith predictor for a process with an integrator and long dead time. Ind. Eng. Chem. Res. **42**, 484–489 (2003)
21. Hu, W., Xiao, G.: Analytical Proportional-Integral (PI) controller tuning using closed loop set point response. Ind. Eng. Chem. Res. **50**(4), 2461–2466 (2011). doi:10.1021/ie101475n
22. Huang, H.P., Chen, C.C.: Control system synthesis for open loop unstable system with time delay. IEE Proc., Control Theory Appl. **144**, 334–346 (1997)
23. Jacobsen, E.W.: On the dynamics of integrated plants non-minimum phase behavior. J. Process Control **9**, 439–451 (1999)
24. Jhunjhunwala, M.K., Chidambaram, M.: PID controller tuning for unstable systems by optimization method. Chem. Eng. Commun. **185**, 91–113 (2001)
25. Kaya, I.: Two-degree-of-freedom IMC structure and controller design for integrating processes based on gain and phase-margin specifications. IEE Proc., Control Theory Appl. **151**(4), 401–407 (2004)
26. Kausthubh, S.M., Chidambaram, M.: Control of isothermal crystallizers. In: Proc. of Chemcon, Calcutta, India, pp. 13–16 (2000)
27. Kendi, T.A., Doyle, F.J. III: Nonlinear control of a fluidized bed reactor using approximate feedback linearization. Ind. Eng. Chem. Res. **35**, 746–757 (1996)
28. Khinast, J., Luss, D., Harold, M.P., et al.: Continuously stirred decanting reactor: Operability and stability considerations. AIChE J. **44**, 372–387 (1998)
29. Klein, R.E.: Using bicycles to teach system dynamics. IEEE Control Syst. Mag. **9**(3), 4–9 (1989)
30. Kwak, H.J., Sung, S.W., Lee, I.B.: Stabilizability conditions and controller design for unstable processes. Chem. Eng. Res. Des. **78**(A), 549–556 (2000)
31. Lee, J., Park, S.: Adaptive model predictive control for unstable nonlinear processes. J. Chem. Eng. Jpn. **27**, 760–767 (1994)
32. Lee, Y., Lee, J., Park, S.: PID controller tuning for integrating and unstable processes with time delay. Chem. Eng. Sci. **55**, 3481–3493 (2000)
33. Li, K.W., Turksen, IB: Stabilization of unstable and unintuitive plants by fuzzy control. IEEE Trans. Syst. Man Cybern., Part B, Cybern. **27**, 55–67 (1997)
34. Liu, T., Cai, Y.Z., Gu, D.Y., et al.: New modified Smith predictor scheme for integrating and unstable processes with time delay. IEE Proc., Control Theory Appl. **152**(2), 238–246 (2005)

35. Liu, T., Zhang, W., Gu, D.: Analytical design of two degree of freedom control scheme for open-loop unstable processes with time delay. J. Process Control **15**(5), 559–572 (2005)
36. Liou, C.T., Chien, Y.S.: The effect of non-ideal mixing on input multiplicities in a CSTR. Chem. Eng. Sci. **46**, 2113–2116 (1991)
37. Lu, X., Yang, Y.S., Wang, Q.G., et al.: A double two-degree of freedom control scheme for improved control of unstable delay processes. J. Process Control **15**(5), 605–614 (2005)
38. Luyben, W.L.: Dynamics and control of recycle systems. 1. Simple open loop and closed loop systems. Ind. Eng. Chem. Res. **32**, 466–475 (1993)
39. Luyben, W.L.: Dynamics and control of recycle systems. 2. Comparison of alternative process design. Ind. Eng. Chem. Res. **32**, 476–486 (1993)
40. Luyben, W.L.: Dynamics and control of recycle systems. 1. Alternative process design in a ternary system. Ind. Eng. Chem. Res. **32**, 1142–1153 (1993)
41. Luyben, W.L.: Tuning proportional-integral-derivative controllers for integrator/deadtime processes. Ind. Eng. Chem. Res. **35**, 3480–3483 (1996)
42. Majhi, S., Atherton, D.P.: Obtaining controller parameters for a new Smith predictor using auto tuning. Automatica **36**, 1651–1658 (2000)
43. Marlin, T.E.: Process Control: Designing Processes and Control Systems for Dynamic Performance. McGraw-Hill, New York (1995)
44. Morari, M., Zafiriou, E.: Robust Process Control. Prentice Hall, Englewood Cliffs (1989)
45. Namjoshi, A., Kienle, A., Ramakrishna, D.: Steady state multiplicity in bioreactors: Bifurcation analysis of cybernetic models. Chem. Eng. Sci. **58**, 793–800 (2003)
46. Nie, Z.Y., Wang, Q.G., Wu, M., et al.: Lead/lag compensator design for unstable delay processes based on new gain and phase margin specifications. Ind. Eng. Chem. Res. **53**(3), 1330–1337 (2011)
47. Normey-Rico, J.E., Camacho, E.F.: Simple robust dead-time compensator for first-order plus dead-time unstable processes. Ind. Eng. Chem. Res. **47**, 4784–4790 (2008)
48. Oggunaike, B.A., Ray, W.H.: Process Dynamics, Modeling and Control. Oxford Press, New York (1994)
49. Pai, N.S., Chang, S.C., Huang, C.T.: Tuning PI/PID controllers for integrating processes with dead time and inverse response by simple calculations. J. Process Control **20**(6), 726–733 (2010)
50. Panda, R.C.: Synthesis for PID controller for unstable and integrating processes. Chem. Eng. Sci. **64**, 2807–2816 (2009)
51. Rao, A.S., Chidambaram, M.: Enhanced Smith predictor for unstable processes with time delay. Ind. Eng. Chem. Res. **44**, 8291–8297 (2005)
52. Rao, A.S., Chidambaram, M.: Enhanced two-degree-of-freedom control strategy for second order unstable processes with time delay. Ind. Eng. Chem. Res. **45**(10), 3604–3614 (2006)
53. Rao, A.S., Rao, V.S.R., Chidambaram, M.: Simple analytical design of modified smith predictor with improved performance for unstable first order time delay (FOPTD) processes. Ind. Eng. Chem. Res. **46**, 4561–4571 (2007)
54. Rao, A.S., Rao, V.S.R., Chidambaram, M.: Set point weighted modified Smith predictor for integrating and double integrating processes with time delay. ISA Trans. **46**(1), 59–71 (2007)
55. Rao, A.S., Rao, V.S.R., Chidambaram, M.: Direct synthesis based controller design for integrating processes with time delay. J. Franklin Inst. **346**(1), 38–56 (2009)
56. Rao, C.V.N., Sree, R.P.: IMC based controller design for integrating systems with time delay. J. Indian Chem. Eng. **52**(3), 194–218 (2010)
57. Seki, H., Ogawa, M., Ohshima, M.: PID temperature control of an unstable gas-phase polyolefin reactor. J. Chem. Eng. Jpn. **34**, 1415–1422 (2001)
58. Shamsuzzoha, M., Lee, M.: PID controller tuning for integrating processes with time delays. Korean J. Chem. Eng. **25**(4), 637–645 (2008)
59. Shamsuzzoha, M., Lee, M.: Analytical design of enhanced PID filter controller for integrating and first order unstable processes with time delays. Chem. Eng. Sci. **63**(10), 2717–2731 (2008)
60. Shamsuzzoha, M., Lee, M.: Design of advanced PID controller for enhanced disturbance rejection of second order processes with time delay. AIChE J. **54**(6), 1526–1536 (2008)

61. Shamsuzzoha, M., Lee, M.: Enhanced disturbance rejection for open loop unstable processes with time delay. ISA Trans. **48**(2), 237–244 (2009)
62. Shamsuzzoha, M., Skogesatd, S.: The setpoint overshoot method: A simple and fast closed loop approach for PID tuning. J. Process Control **20**(10), 1220–1234 (2010)
63. Skogesatd, S.: Simple analytical rules for model reduction and PID controller tuning. J. Process Control **13**, 291–309 (2003)
64. Song, Y., Mode, M.O., Zhang, T.: Stabilization and algorithm of integrator plus dead time process using PID controller. J. Process Control **19**, 1529–1537 (2009)
65. Sree, R.P., Chidambaram, M.: A simple method of calculating the set point weighting parameter for unstable systems with a zero. Comput. Chem. Eng. **28**, 2433–2437 (2004)
66. Sree, R.P., Chidambaram, M.: Set Point weighted PID controllers for unstable systems. Chem. Eng. Commun. **192**, 1–13 (2005)
67. Sree, R.P., Chidambaram, M.: Simple and robust method of tuning PID controllers for integrator/dead time processes. J. Chem. Eng. Jpn. **38**(2), 113–119 (2005)
68. Sree, R.P., Chidambaram, M.: Control of Unstable Processes. Narosa, New Delhi (2006)
69. Sree, R.P., Srinivas, M.N., Chidambaram, M.: A simple method of tuning PID controllers for stable and unstable FOPTD systems. Comput. Chem. Eng. **28**, 2201–2218 (2004)
70. Tan, W., Marquez, H.J., Chen, T.: IMC design for unstable processes with time delays. J. Process Control **13**, 203–213 (2003)
71. Tyreus, B.D., Luyben, W.L.: Tuning PI controllers for integrator/deadtime processes. Ind. Eng. Chem. Res. **31**, 2625–2628 (1992)
72. Uma, S., Chidambaram, M., Rao, A.S.: Enhanced control of unstable cascade processes with time delays using modified Smith predictor. Ind. Eng. Chem. Res. **48**(6), 3098–3111 (2009)
73. Uma, S., Chidambaram, M., Rao, A.S.: Set-point weighted modified Smith predictor for non-minimum phase integrating processes. Chem. Eng. Res. Des. **88**, 592–601 (2010)
74. Uma, S., Chidambaram, M., Rao, A.S., et al.: Enhanced control of integrating cascade processes with time delays. Chem. Eng. Sci. **65**, 1065–1075 (2010)
75. Van Elk, E.P., Borman, B.C., Kuuipers, J.A.M., et al.: Modeling of gas liquid mass transfer accompanied by irreversible reaction. Chem. Eng. Sci. **54**, 4869–4879 (1999)
76. Veronesi, M., Visioli, A.: Performance assessment and retuning of PID controllers for integral processes. J. Process Control **20**(3), 261–269 (2010)
77. Visioli, A.: Optimal tuning of PID controllers for integral and unstable processes. IEE Proc., Control Theory Appl. **148**(2), 180–184 (2001)
78. Walker, D.J.: Multivariable control of longitudinal and lateral dynamics of a fly by wire helicopter. Control Eng. Pract. **11**, 781–795 (2003)
79. Wang, Y.G., Cai, W.J.: Advanced proportional-integral-derivative tuning for integrating and unstable processes with gain and phase margin specifications. Ind. Eng. Chem. Res. **41**, 2910–2914 (2002)
80. Wu, W.: Stable inverse control for non-minimum phase nonlinear processes. J. Process Control **9**, 171–183 (1999)
81. Zhang, W., Gu, D., Wang, W., et al.: Quantitative performance design of a modified Smith predictor for unstable processes with time delay. Ind. Eng. Chem. Res. **43**, 56–67 (2004)
82. Zhong, Q.C., Visioli, A.: Control of Integral Processes with Time Delays. Springer, London (2011)
83. Zhou, H.Q., Wang, Q.G., Shieh, L.S.: PID control of unstable processes with time delay: A comparative study. J. Chem. Eng. Jpn. **40**(2), 145–163 (2007)

Chapter 4
Robustness in PID Control

Ramon Vilanova, Víctor M. Alfaro, and Orlando Arrieta

4.1 Introduction

Undoubtedly, since its introduction in 1940 [18, 20], PID controllers (proportional integral derivative) are the option most frequently used in different process control applications. Its success is due mainly to the simplicity of its structure (three parameters to tune) and operation, which allows the control engineer a better understanding compared with other advanced control techniques. This has motivated the continuous research efforts aimed at finding alternative approaches to the design and new tuning rules in order to improve the performance of control loops based in PID controllers. Most of these research works that have emerged over the years take the form of design proposals based on simple models and generally give rise to tuning rules that link the parameters of the process model with those of the controller in a direct and simple way. The need for such simple and model-based tuning rules is also encouraged by several control engineering books, some of them specifically on PID control, for example, [16, 36, 58]. A common point that can be found in all of them is the need to incorporate a good understanding of the control problem and its relationship with modeling and knowledge of the process to be controlled.

R. Vilanova (✉)
Departament de Telecomunicació i d'Enginyeria de Sistemes, Escola d'Enginyeria, Universitat Autònoma de Barcelona, Bellaterra, 08193 Barcelona, Spain
e-mail: Ramon.Vilanova@uab.cat

V.M. Alfaro · O. Arrieta
Departamento de Automática, Escuela de Ingeniería Eléctrica, Universidad de Costa Rica, San José 11501-2060, Costa Rica

V.M. Alfaro
e-mail: Victor.Alfaro@ucr.ac.cr

O. Arrieta
e-mail: Orlando.Arrieta@ucr.ac.cr

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_4, © Springer-Verlag London Limited 2012

In this sense, the initial work developed by Ziegler and Nichols [64] constitutes the first attempt for a systematic design of a PID controller on the basis of minimal process information. This work was followed by the also well-known work [23], which presents a modification of the Ziegler and Nichols approach and, more importantly, raises the need to consider different methods for reference tracking and disturbance attenuation. The introduction of ideas about the algebraic approach to design gave rise to the so-called λ-tuning method of [25]. This method, in turn, is closely related to the Smith predictor and the design methods based on Internal Model Control (IMC) of [44]. Among them of particular interest are the considerations that show the IMC control takes the form of a PI or PID controller, depending on the rational approximation used for the time delay. These approaches, however, use the cancellation of the process poles, which can lead to sluggish disturbance responses, especially for processes with large time constants. In [22] a modification is presented in order to avoid such cancellation, while in [47], Skogestad presents a variation of the IMC controller approach, called SIMC, applied to the tuning of PI and PID, which prevents the cancellation by a redefinition of the integral term, in case of systems dominated by large time constants.

The methods based on the application of optimization techniques are an alternative to the analytical ones. The basic idea is to try to capture the different aspects of the desired closed-loop operation through the definition of a given cost functional to be minimized. In [46] and [48], for example, tunings for optimized tracking are provided regarding the integral error criteria ISE, IAE, and ITAE, and in [63] also compact tuning rules for different formulations of the integral criteria are provided. In [57] the appropriate analysis for unstable and integral systems is presented.

Recently, in part because of better accessibility to optimization routines and numerical software, optimization approaches based on multiobjective optimization have appeared. See, for example, the works [29, 53] that set a general approach and exemplify its use for a PID controller. However, the application of these optimization strategies, although effective, depends on the use of quite complex numerical methods and do not result in tuning rules. By its application, what you do get is the tuning of the controller as the solution of the particular optimization problem.

This distinction, which can also be interpreted as a kind of classification, is particularly of application on those approaches that focus on the attainment of a PID controller with certain robustness characteristics. The robustness was an aspect not included as an integral part of the considerations to be included when designing a control system, not only in the case of PID control, but also from a more wide perspective. It is not until the formulation initiated by Zames in [62] that deals with the uncertainty by using the infinity norm that robust control has entered completely into control theory and is by now a fully developed and mature approach, known as the $\mathcal{H}_\infty$ control. This approach is reflected in numerous books that offer a broad vision on the subject and the different variations and approaches that emerged [26, 41, 54].

The ideas emerging from the development of robust control have, of course, finally been specialized to the case of PID control. This permeation has led to different approaches of what we can call *robust* PID control. In this way, we distinguish

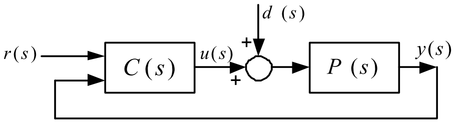


Fig. 4.1 Single-input single-output feedback control system

between the obtaining a robust PID controller, as the result of the solution of a robust control problem, applied to a restricted structure controller, from the widely proposed simple tuning rules to be incorporated into its conception robustness considerations.

In this way, for example, we have methods that are developed on the basis of the internal model control [44], where the resulting tuning rules are parameterized in terms of a variable directly related to the robustness of the system. However, this robustness is not directly linked to a robustness metric or a quantitative robustness measure. On the other hand, there are well-known design strategies that are based on setting the gain and phase margin, initiated in [12] that have given rise to numerous variants and extensions. In this case, the design parameter or specification is directly measuring the desired robustness for the closed-loop system. The idea has spread today to a common use of the maximum of the sensitivity function (commonly called M_S) as a reasonable robustness measure. At this point it is also possible to distinguish between approaches that are attempting to achieve a closed-loop with a particular value of M_S [15] and more flexible approaches providing tuning rules directly parameterized by the target M_S value [8, 11].

The purpose of this chapter is to provide an overview of different approaches to Robust PID design that have arisen during the recent years. These are methods that include explicit considerations of robustness in their conception. A distinction is made between generic tuning rules that are aimed at providing a robust control system but that are not linked to any robustness measure to recent tuning approaches that use explicit robustness terms and definitions in order to measure the level of robustness that is desired for the resulting closed loop. The chapter ends by presenting some considerations about the robustness/performance *tradeoff*.

4.2 Framework and Robustness Measures

In what follows, we will use a single-input single-output control system as it is shown in Fig. 4.1. Note that this is not the control diagram as complete as it could be found, but it reflects the necessary points to facilitate the discussion that follows. The process to be controlled, $P(s)$, is generally represented by a model given by the transfer function:

$$P(s) = \frac{Ke^{-Ls}}{(Ts+1)(aTs+1)}, \qquad \tau_o = L/T, \tag{4.1}$$

where K is the model gain, T its main time constant, a the ratio of the two time constants ($0 \leq a \leq 1.0$), L its dead-time, and τ_o is the model *normalized dead-time*.

Model (4.1) allows us to represent First-Order-Plus-Dead-Time (FOPDT) processes, $a = 0$, over damped Second-Order-Plus-Dead-Time (SOPDT) processes, $0 < a < 1$, and Dual-Pole-Plus-Dead-Time (DPPDT) processes, $a = 1$. A common characterization of the process parameters is made in terms of the normalized dead-time $\tau_o = L/T$ [58]. Note that (4.1) is a more general framework than the one used in traditional works on PI/PID design [13, 16, 58].

On the other hand, the process is controlled by using a PID controller [13] whose output is

$$u(t) = K_p \left\{ e_p(t) + \frac{1}{T_i} \int_0^t e_i(\tau)\, d\tau + T_d \frac{de_d(t)}{dt} \right\} \tag{4.2}$$

with

$$e_p(t) = \beta r(t) - y(t), \tag{4.3}$$

$$e_i(t) = r(t) - y(t), \tag{4.4}$$

$$e_d(t) = \gamma r(t) - y(t), \tag{4.5}$$

where K_p is the controller gain, T_i the integral time constant, T_d the derivative time constant, and β and γ are the set-point weights. The γ parameter is more frequently applied as a derivative mode *switch* (0 or 1) for the set-point signal r. To avoid an extreme instantaneous change, so-called derivative-kick, in the controller output signal when a set-point step change occurs normally γ is set to zero. In this case the controller output (4.2) may be expressed as

$$u(s) = K_p \left\{ e_p(s) + \frac{1}{T_i s} e_i(s) + \frac{T_d s}{T_d/N s + 1} e'_d(s) \right\} \tag{4.6}$$

with

$$e_p(s) = \beta r(s) - y(s), \tag{4.7}$$

$$e_i(s) = r(s) - y(s), \tag{4.8}$$

$$e'_d(s) = -y(s), \tag{4.9}$$

where N is the *derivative filter constant* (usually $N = 10$ [58]). Equation (4.6) may be arranged as

$$u(s) = K_p \left\{ \left(\beta + \frac{1}{T_i s} \right) r(s) - \left(1 + \frac{1}{T_i s} + \frac{T_d s}{0.1 T_d s + 1} \right) y(s) \right\} \tag{4.10}$$

or in the compact form as

$$u(s) = C_r(s) r(s) - C_y(s) y(s), \tag{4.11}$$

where $C_r(s)$ is the *set-point controller* transfer function, and $C_y(s)$ is the *feedback controller* transfer function.

If $C_r(s)$ is equal to $C_y(s)$, it is not possible to specify the dynamic performance of the control system to set-point changes, independently of the performance to load-disturbances changes, and we would have a One-Degree-of-Freedom (1DoF) controller. Otherwise, if we can select both closed-loop transfer functions independently, the controller would be a Two-Degree-of-Freedom (2DoF).

Thus, based on the diagram in Fig. 4.1, the following relationships between the input and output signals are defined:

$$y(s) = T(s)r(s) + P(s)S(s)d(s), \tag{4.12}$$

$$u(s) = C(s)S(s)r(s) + T(s)d(s), \tag{4.13}$$

where the functions $S(s)$ and $T(s)$ are respectively the sensitivity function and complementary sensitivity, which are defined as

$$S(s) \doteq \frac{1}{1 + P(s)C(s)}, \qquad T(s) \doteq \frac{P(s)C(s)}{1 + P(s)C(s)}, \tag{4.14}$$

and, as it will be seen, they play a key role in the determination on the stability and robustness properties of the control system.

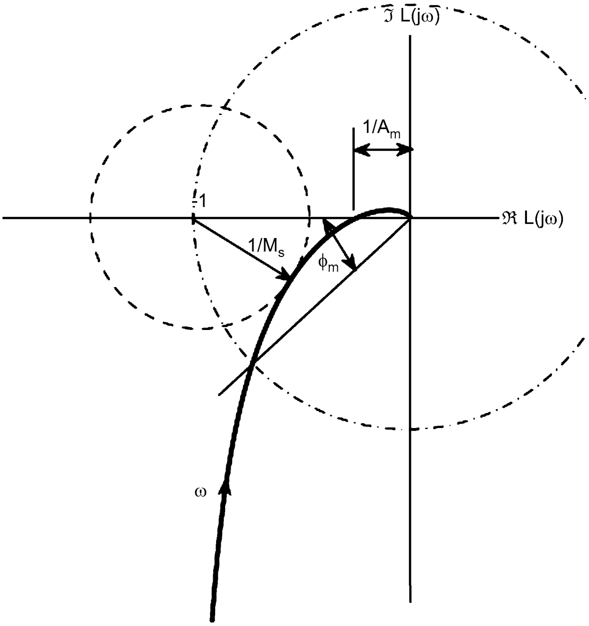
When dealing with the stability of the control system, it is usually understood as absolute stability. That is, the closed-loop transfer function, provided that no cancellation of unstable modes takes place, has all its poles in the left half-plane. However, if it is desired to introduce performance measures regarding stability, then we need the concept of relative stability. Relative stability is used to determine *how stable the system is* or, alternatively, the closeness to the instability. In other words, the distance of the closed-loop to instability. This idea of distance to instability, generates different interpretations as it relates to possible sources of error in the model used for the design of the controller. How much and what type of uncertainty can be allowed before the control system becomes unstable? Given this interpretation, the various measures of relative stability are called *robustness measures*. Thus, the incorporation of these measures in the design procedures and the conception of the tuning rules result in what is known as robust controller design, in our case, a robust PID controller.

4.2.1 Classical Robustness Measures

Robustness measures have evolved over the years, leading to new approaches in robust control. In this way, the gain and phase margin are known as classical relative stability measures. These measures are based on the Nyquist stability criterion and consider the possibility of varying the number of encirclements of the critical point $(-1, 0)$, that may arise from the variation in the gain or phase, respectively, of the system.

The *gain margin*, A_m, is a specification of the point, on the Nyquist plot, at which the frequency response of the loop transfer function must cross the negative

Fig. 4.2 Definitions of the Maximum Sensitivity M_S and the Gain A_m and Phase ϕ_m margins



real axis. This specification defines a multiplying factor for the gain for which the system becomes unstable. Figure 4.2 represents the geometry for the gain margin. The A_m is defined on the basis of the following condition:

$$A_m \left| C(j\omega_{-\pi}) P(j\omega_{-\pi}) \right| = 1; \tag{4.15}$$

therefore,

$$A_m = \frac{1}{|C(j\omega_{-\pi}) P(j\omega_{-\pi})|}. \tag{4.16}$$

The frequency $\omega_{-\pi}$ is the frequency where the gain margin is computed. This is the frequency where the phase of the loop transfer function is $-180°$. The interpretation of the margin as a measure of robustness shows that in the case where the system model is incorrect, the static gain may increase by a factor A_m before the system becomes unstable. Typical values for A_m are $2 \le A_m \le 5$.

On the other hand, the *phase margin*, ϕ_m, specifies by what amount the phase of the system may be delayed so that the corresponding rotation generated on the system's frequency response in the Nyquist plot leads to cross the critical point $(-1, 0)$. The geometry corresponding to the phase margin is shown in Fig. 4.2 and leads to the following condition (in degrees):

$$-\phi_m + \arg\left\{ C(j\omega_1) P(jw_1) \right\} = -180°, \tag{4.17}$$

where

$$\phi_m = 180° + \arg\left\{ C(j\omega_1) P(j\omega_1) \right\}. \tag{4.18}$$

The frequency ω_1 at which the phase margin is evaluated is the frequency at which the system gain equals one. Analogously to the gain margin, the interpretation of the phase margin as a measure of robustness shows that if there is an error in the modeling process, it may suffer an additional phase delay of ϕ_m degrees at the frequency w_1 before the system becomes unstable. Typical values for the phase margin are $30^\circ \leq \phi_m \leq 60^\circ$.

The design based on the specification of A_m and ϕ_m can be considered as the first approach where the robustness is specified in terms of explicit robustness measures. The first design proposal based on A_m and ϕ_m is presented in [12] for A_m =3 and $\phi_m = 60^\circ$.

The problem with designs based on A_m and ϕ_m is that the resulting expressions for computing the controller parameters are highly nonlinear. For its solution, crossover frequencies have also to be computed, and their appearing in the equations along with the $\tan^{-1}$ function prevents its analytical calculation. Because of this basic problem, different approaches have emerged based on approximations which provide relationships that allow both the calculation of the A_m and ϕ_m associated with a particular tuning and the controller parameters given predetermined values for A_m and ϕ_m. In [30], for example, equations for evaluating the A_m and ϕ_m of a control-loop based on a PID controller and a first-order system with delay are provided. These equations are then applied to evaluate the A_m and ϕ_m resulting from several tuning rules such as the classic Ziegler–Nichols [64], Cohen–Coon [24], and the optimal rules derived from (ISE, IAE, ITAE). Recommended values for A_m are between 2 and 5, and between 30^o and 60° for the ϕ_m. An extensive analysis can be found in [30–32, 34].

For Multiple-Input-Multiple-Output (MIMO) systems, all the above considerations should be taken into account, in order to measure the relative stability. However, the well-known methods to calculate the gain and phase margins of a Single-Input-Single-Output (SISO) systems are not applicable for MIMO processes, due the loop interactions [59, 61]. In fact, so far, there seems to be neither satisfactory definitions for MIMO gain and phase margins, nor effective techniques for determining them; however, there are some alternative definitions to achieve these values and to use them for multivariable control systems [19, 33, 60].

Another problem with the use of A_m and ϕ_m as robustness specifications is that both measures are independent, and compliance with both must therefore be ensured. It is for this reason that in recent years, the specification of robustness has focused on the use of the sensitivity function, as a more general, unstructured, measure that, in turn, provides bound values for these two specifications. The next section presents the idea of sensitivity, its role in a feedback control system, and its definition as a robustness measure.

4.2.2 Use of the Sensitivity Function

The sensitivity function, $S(s)$, is today the basis for establishing the robustness level within the different approaches for the design of PID controllers. However, its interpretation as a function to determine the tolerance of the control-loop to changes in the process control is well known from the formulation of the feedback loop as a control strategy. Indeed, if we have a function f that depends on a parameter α, the sensitivity of f to variations in α is denoted by S_α^f and is defined by

$$S_\alpha^f \doteq \lim_{\Delta\alpha \to 0} \left.\frac{\Delta f/f}{\Delta\alpha/\alpha}\right|_{\alpha=\alpha_o} = \left.\frac{\alpha}{f}\frac{\partial f}{\partial \alpha}\right|_{\alpha=\alpha_o}, \tag{4.19}$$

where α_o is the nominal value of α, and Δf and $\Delta\alpha$ represent the deviations of α and f with respect to their nominal values. We will apply this concept to the relationship between the reference and output of a control system for both the open-loop ($T_{\text{ol}}(s)$) and closed-loop ($T_{\text{cl}}(s)$) versions, given respectively by

$$T_{\text{ol}}(s) = P(s)C(s), \qquad T_{\text{cl}}(s) = \frac{P(s)C(s)}{1+P(s)C(s)}. \tag{4.20}$$

Thus, in the presence of uncertainty in the process, $P(s) = P_o(s) + \Delta P$, the previous input–output relationships suffer of this uncertainty and will deviate from their values, obtained on the basis of the application of the controller to the nominal plant: $T_{\text{ol}}(s) = T_{\text{ol},o}(s) + \Delta T_{\text{ol}}$, $T_{\text{cl}}(s) = T_{\text{cl},o}(s) + \Delta T_{\text{cl}}$. Applying the above definition, the sensitivity of $T_{\text{ol}}(s)$ and $T_l c(s)$ to variations in the plant $P(s)$ are determined as

$$S_P^{T_{\text{ol}}} = \lim_{\Delta P \to 0} \frac{\Delta T_{\text{ol}}(s)/T_{\text{ol},o}(s)}{\Delta P(s)/P_o(s)} = 1, \tag{4.21}$$

$$S_P^{T_{\text{cl}}} = \lim_{\Delta P \to 0} \frac{\Delta T_{\text{cl}}(s)/T_{\text{cl},o}(s)}{\Delta P(s)/P_o(s)} = S(s). \tag{4.22}$$

Thus, for the open-loop relationship, modeling errors go entirely to the input–output relationship. On the other hand, for the feedback-based closed-loop configuration, this change in the process is multiplied by the sensitivity function $S(s)$. It is at this point that the controller $C(s)$ can be chosen in such a way that this function has small magnitude within the uncertainty frequency range. This is recognized as one of the key factors for the use of feedback.

The sensitivity function, as presented, is a frequency-dependent function and therefore cannot be used directly as a figure of merit that provides us with a measure of robustness. However, if we notice that $S(s)$ only depends on the direct chain of the control loop, $L(s) = P(s)C(s)$, it is possible to give a geometric interpretation of it in the Nyquist plane. Indeed, since $S(s) = (1 + L(s))^{-1}$, we can represent the complex value $1 + L(j\omega)$ as the vector that goes from the point $(-1, 0)$ to $L(j\omega)$. The sensitivity function will therefore be less than one for those frequencies where the Nyquist curve is outside a circle with center $(-1, 0)$ and unit radius (see Fig. 4.2). Based on this interpretation, we can define the M_S value as

$$\begin{aligned} M_S &\doteq \max_\omega \left|S(j\omega)\right| = \max_\omega \left|\frac{1}{1+C(jw)P(j\omega)}\right| \\ &= \frac{1}{\min_\omega |1 + L(j\omega)|}, \end{aligned} \tag{4.23}$$

the shortest distance from the Nyquist curve to the critical point $(-1, 0)$ being equal to $1/M_S$. Thus, ensuring a certain value for M_S guarantees that the Nyquist locus is away from the critical point a distance equal to $1/M_S$. Therefore, in order to obtain high robustness, small values of M_S will be of interest. Thus, M_S is a robustness measure with widespread use in today's research on robust control systems. This measure is more general than the previously presented gain and phase margins. In fact, it is possible to show that a particular M_S value guarantees, simultaneously, the following bounds for the gain and phase margins:

$$A_m \geq \frac{M_S}{M_S - 1}, \qquad \phi_m \geq 2\sin^{-1}\left(\frac{1}{M_S}\right). \tag{4.24}$$

Another implication of ensuring that the Nyquist curve remains outside a circle centered at the critical point and with radius $1/M_S$ is that stability is guaranteed even though the gain system increases by a factor $M_S/(M_S - 1)$ or reduced by a factor $M_S/(M_S + 1)$. Additionally, it is possible to guarantee stability despite a static nonlinearity, $f(x)$, in the loop, provided that it is ascertained that

$$\frac{M_S}{M_S + 1} < \frac{f(x)}{x} < \frac{M_S}{M_S - 1}. \tag{4.25}$$

Typical values for M_S vary in the range of 1.4 to 2. For $M_S = 2$, we have $A_m \geq 2$ and $\phi_m \geq 29°$ (that is recognized as the minimum acceptable robustness). For the case $M_S = 1.4$, it is guaranteed that $A_m \geq 3.5$ and $\phi_m \geq 41°$. The advantage of using M_S is that it determines the A_m and ϕ_m simultaneously, while these two quantities are completely independent. In turn, M_S also appears in the comparative analysis that is usually made of the robustness achieved by certain approaches, although it was not explicitly considered at the design stage.

4.2.2.1 A Generalized Sensitivity Criteria

From the perspective of robust control theory, stability of the closed-loop control system is maintained despite of changes, $\Delta P(s)$, in the nominal model, $P_o(s)$, used for design provided that the following condition is satisfied:

$$\left|C(j\omega)\Delta P(j\omega)\right| < \left|1 + L(j\omega)\right|, \tag{4.26}$$

an expression that can be rewritten as

$$\left|\frac{\Delta P(j\omega)}{P_o(j\omega)}\right| < \frac{1}{|T(j\omega)|}. \tag{4.27}$$

Thus, variations in the process model can be allowed provided that at the corresponding frequencies, the complementary sensitivity transfer function is *small*. This leads to consider the maximum value of $T(s)$ as an estimate of this tolerance:

$$M_T = \max_{\omega}\left|T(j\omega)\right| = \max_{\omega}\left|\frac{L(j\omega)}{1 + L(j\omega)}\right|. \tag{4.28}$$

In this way, we can formulate a general robustness margin as in [37] and [39], GM_S, defined by

$$GM_S = \max\big\{\|S\|_\infty, \alpha\|T\|_\infty\big\}, \tag{4.29}$$

where the parameter α is defined as $\alpha = M_S^d/M_T^d$ based on the desired values for the maximum of the sensitivity functions. Geometrically, specifying a constraint value for M_T imposes a prohibited area for the Nyquist curve given by a circle centered at $(-M_T^2/(M_T^2-1), 0)$ and radius $M_T/(M_T^2-1)$. Thus, by using a specification for M_S and a specification for M_T, we are constraining the Nyquist locus to stay out of the corresponding circles. The idea of having an area, to be avoided by the Nyquist locus, determined by different circles, is presented in detail in [16].

4.3 Robust-Based PID Control Designs

The use of the gain and phase margins as robustness measures has been replaced by the use of a single indicator, the maximum of the sensitivity function, denoted by M_S.

The use of the sensitivity function in PID control design arises linked to the suggestion to optimize the disturbance rejection by means of imposing some constraints on the sensitivity function. The use of the sensitivity function peak, M_S, was proposed by [43]. However, it was not until the presentation of the *Kappa–Tau* tuning rules [13], that its use spreads, and the development of PID tuning rules that makes use of M_S in an explicit or implicit way starts. This section presents a review of existing methods that have appeared in the literature and that can be considered as robust PID control approaches based on M_S specifications.

4.3.1 Maximum Sensitivity Based Designs

4.3.1.1 *Kappa–Tau* (KT) Method [13]

The method uses an empirical design approach based on the placement of the closed-loop dominant poles, along with a constraint on M_S for robustness. Tuning rules for PI and PID controllers with one and two degrees of freedom (2DoF) are provided. The method uses a particular set of test plants as a representation of the various dynamics that can be found in industrial processes, characterized by its gain K, the dominant time constant T, and its apparent delay L. Starting from these processes, a first-order plus time delay model is obtained from the reaction curve or, instead, their critical gain K_{pu} and period of oscillation T_u. The method establishes two levels of robustness, a minimum level $M_S = 2.0$ and high level by $M_S = 1.4$. The range of application of the method is provided by $0.1 \le L/(L+T) \le 0.85$.

4.3.1.2 *AMIGO* Method [15, 27, 28]

This approach builds on the *MIGO* (“M-constrained Integral Gain Optimization”) method in [17] that optimizes the integral gain of the controller, K_i, and imposes a constraint on the Maximum Sensitivity, $M_S = 1.4$, applied to a benchmark plant set, including multiple pole plants, processes with inverse response, integral processes, etc. From the optimal controller parameters, tuning equations are determined for robust PI and PID controllers in the standard 2DoF form. A review of the AMIGO method (“Approximate MIGO”) is presented in [16]. Although the range of validity for the application of the AMIGO method is $0 \leq L/(L+T) \leq 0.98$, for PID controllers, it is possible to get reliable parameters only within the range $L/(L+T) > 0.5$.

4.3.1.3 KLκ150 Method [37]

This approach is based on a generalized stability criterion that involves a constraint on both the maximum of the sensitivity function and the maximum of the complementary sensitivity. The approach states an optimization problem constrained by the Generalized Maximum Sensitivity, GM_S, criterion. The objective function to be optimized includes the performance for disturbance attenuation, J_d, control effort J_u, and sensitivity to high-frequency measurement noise. Tuning rules are determined for standard PI and PID controllers in terms of the estimated gain of the plant and the frequency at which its phase is $-150°$. With this design, the achieved robustness is $GM_S \approx 1.7$, and performance is quite close to the optimal IAE. The method has the unusual feature of considering the derivative filter constant, as one of the adjustable parameters of the controller [39].

4.3.1.4 Multiple Objective Optimization (MOO) [52]

In a first development [51] the optimal IAE performance to changes in the set point and the disturbance, along with the constraint on the minimal robustness $M_S \leq 2.0$, is considered. The design tackles PI 2DoF controllers from first-order models with delay and integral models with delay. Later on, a MOO method is proposed that, along with the previous considerations, penalizes the variation on the total control effort TV_u and uses a genetic algorithm to solve a multiobjective optimization problem. The controllers obtained with this tuning provide control systems with a nominal robustness $M_S \approx 1.7$ for first-order models with delay into the range

$0.1 \leq \tau_o \leq 2.0$. The MOO equations for first-order models with delay are:

$$\begin{aligned} K_p K &= \frac{1}{6} + \frac{5}{11\tau_o}, \\ \frac{T_i}{T} &= \frac{\frac{3\tau_o}{14} + \frac{7}{6}}{\tau_o + \frac{1}{5}}\tau_o, \\ \beta &= \frac{4\tau_o}{9} + \frac{1}{2}, \end{aligned} \tag{4.30}$$

and, for integral processes with delay,

$$\begin{aligned} K_p K &= \frac{5}{11\tau_o}, \\ \frac{T_i}{T} &= \frac{35\tau_o}{6}, \\ \beta &= \frac{1}{2}. \end{aligned} \tag{4.31}$$

4.3.1.5 PI2_{Ms} Method [9]

Based on the optimum parameters obtained by the Non-Oscillatory Robust Method (NORT) [7], for first-order processes with time delay and $0.1 \leq \tau_o \leq 2.0$, this method provides tuning rules for a 2DoF PI controller for four levels of robustness, $M_S \in \{1.4, 1.6, 1.8, 2.0\}$. The non-oscillatory output for reference input changes and disturbances, imposed on the design, also ensure a smooth behavior of the control signal. The main characteristic of the method is to achieve four levels of robustness for all the range of application. PI2_{Ms} tuning rules for first-order models with delay are fitted to very simple equations as follows:

$$\begin{aligned} \kappa_p \doteq K_p K &= a_0 + a_1\tau_o^{a_2}, \\ \tau_i \doteq \frac{T_i}{T} &= b_0 + b_1\tau_o^{b_2}, \\ \beta &= c_0 + c_1\tau_o^{c_2}, \end{aligned} \tag{4.32}$$

where the constants $\{a_i, b_i, c_i\}$ are found in Table 4.1.

4.3.2 A Generic M_S-Based Tuning

In [11], there is proposed a 1DoF PID tuning resulting from the optimization of a joint criteria that treats the tradeoff problem between the performance for servo

Table 4.1 Constants for the PI$_{2M_S}$ method [7]

M_S	a_0	a_1	a_2
1.4	0.0674	0.3775	−0.9623
1.6	0.1687	0.4724	−0.9805
1.8	0.2118	0.5633	−0.9823
2.0	0.3208	0.5613	−1.0380
M_S	b_0	b_1	b_2
1.4	1.440	−0.1744	−0.659
1.6	8.672	−7.247	−0.04929
1.8	−3.952	5.426	0.08661
2.0	−2.105	3.595	0.1476
M_S	c_0	c_1	c_2
1.4	0.3803	0.7794	0.6851
1.6	0.2611	0.5763	0.4345
1.8	0.2296	0.4711	0.3588
2.0	0.2107	0.4043	0.3122

and regulation operation and that also takes into account the attainment of an arbitrary robustness level or value. In this sense, the following cost objective function is formulated:

$$J_{rd} = \sqrt{\left(J_r^{rd} - J_r^o\right)^2 + \left(J_d^{rd} - J_d^o\right)^2}, \tag{4.33}$$

where J_r^o and J_d^o are the optimal values for servo and regulation control, respectively, and J_r^{rd}, J_d^{rd} are the performance indexes for the *intermediate* tuning considering both operational modes. The index (4.33) represents a balanced performance for both operation modes, where the main idea is to be closer, as much as possible, to the "ideal" point (J_r^o, J_d^o), which means the minimum performance values taking both possible operational modes, servo and regulation, into account. However, this point is unreachable due the differences in the dynamics for each of the objectives of the control operational modes. Therefore, the aim is to get the minimum resulting distance, meaning the best balance between the operational modes.

The cost functional (4.33) just takes into account characteristics of performance. However, there is a need to include a certain robustness for the control loop. In that sense, it used (4.23) as a robustness measure. So, the optimization problem is subject to a constraint of the form

$$\left|M_S - M_S^d\right| = 0, \tag{4.34}$$

where M_S and M_S^d are the Maximum Sensitivity and the desired Maximum Sensitivity functions, respectively. This constraint tries to guarantee the selected robustness value for the control system. A broad classification is established, using specific val-

Table 4.2 PID settings for servo/regulation tuning with robustness consideration [11]

Constant	M_s^d-free	$M_s^d = 2.0$	$M_s^d = 1.8$	$M_s^d = 1.6$	$M_s^d = 1.4$
a_1	1.1410	0.7699	0.6825	0.5678	0.4306
b_1	−0.9664	−1.0270	−1.0240	−1.0250	−1.0190
c_1	0.1468	0.3490	0.3026	0.2601	0.1926
a_2	1.0860	0.7402	0.7821	0.8323	0.7894
b_2	0.4896	0.7309	0.6490	0.5382	0.4286
c_2	0.2775	0.5307	0.4511	0.3507	0.2557
a_3	0.3726	0.2750	0.2938	0.3111	0.3599
b_3	0.7098	0.9478	0.7956	0.8894	0.9592
c_3	−0.0409	0.0034	−0.0188	−0.0118	−0.0127

ues for M_S, within the suggested range between 1.4–2.0. This allows a qualitative specification for the control system robustness. So, the rating is described here as follows:

- Low robustness level—$M_s = 2.0$.
- Medium-low robustness level—$M_s = 1.8$.
- Medium-high robustness level—$M_s = 1.6$.
- High robustness level—$M_s = 1.4$.

According to this principle, the above-mentioned four values for M_S are used here as desirable robustness values, M_S^d in the robustness constraint (4.34) for the optimization problem. Additionally, the unconstrained optimization case is known as the M_S^d-free case.

With the solution of the optimization problem, a fitted tuning rule for PID controllers, expressed in terms of the FOPDT model, was devised as:

$$\begin{aligned} K_p K &= a_1 \tau^{b_1} + c_1, \\ \frac{T_i}{T} &= a_2 \tau^{b_2} + c_2, \\ \frac{T_d}{T} &= a_3 \tau^{b_3} + c_3, \end{aligned} \tag{4.35}$$

where the constants a_i, b_i, and c_i are given in Table 4.2, according to the desired robustness level for the control system.

With the aim to give more completeness to the tuning method, an extension of the approach is presented. Taking the advantage of the simplicity and homogeneity offered by tuning (4.35), a simple tuning rule that allows us to specify, in an explicit way, an arbitrary value for the robustness parameter within the range

$M_S \in [1.4, 2.0]$. In this case, (4.35) can be rewritten as:

$$\begin{aligned} K_p K &= a_1(M_S^d)\tau^{b_1(M_S^d)} + c_1(M_S^d), \\ \frac{T_i}{T} &= a_2(M_S^d)\tau^{b_2(M_S^d)} + c_2(M_S^d), \\ \frac{T_d}{T} &= a_3(M_S^d)\tau^{b_3(M_S^d)} + c_3(M_S^d), \end{aligned} \tag{4.36}$$

where the constants are expressed as functions of M_S^d. Therefore, from Table 4.2 all constant a_i, b_i, and c_i are generated from a generic second-order M_S^d-dependent polynomial as

$$\begin{aligned} a_1 &= -0.3112(M_S^d)^2 + 1.6250(M_S^d) - 1.2340, \\ b_1 &= 0.0188(M_S^d)^2 - 0.0753(M_S^d) - 0.9509, \\ c_1 &= -0.1319(M_S^d)^2 + 0.7042(M_S^d) - 0.5334, \\ a_2 &= -0.5300(M_S^d)^2 + 1.7030(M_S^d) - 0.5511, \\ b_2 &= -0.1731(M_S^d)^2 + 1.0970(M_S^d) - 0.7700, \\ c_2 &= -0.0963(M_S^d)^2 + 0.7899(M_S^d) - 0.6629, \\ a_3 &= 0.1875(M_S^d)^2 - 0.7735(M_S^d) + 1.0740, \\ b_3 &= 0.3870(M_S^d)^2 - 4.7810(M_S^d) + 4.9470, \\ c_3 &= 0.1331(M_S^d)^2 - 0.4733(M_S^d) + 0.4032. \end{aligned} \tag{4.37}$$

In short, parameters (4.36), jointly with (4.37), determine the PID controller for any arbitrary value M_S^d in the range [1.4, 2.0]. It is important to note that the cited tuning just depends on the system model information and the design parameter M_S^d.

4.4 Robust PID Tuning Rules

Despite the high number of existing tuning rules [42], one can say that it is from the formulation of the IMC-PID [44] that they started to appear as tuning rules parameterized to give the PID parameters according to a desired degree of robustness (obviously with a corresponding implication for the performance). The formulation of tuning rules within the IMC framework introduces two ideas that have been widely exploited in recent years and have given rise to a multitude of formulations and tuning rules: first, the analytical design based on a process model, and the other, incorporating robustness considerations from existing discrepancies between this model and the actual process. These would be two of the features that made the

IMC approach to become so popular, and, of course, these ideas have been inherited by PI/PID control.

As a distinguishing factor from the tuning rules that have been seen so far in this section, we present here tuning rules that arise in order to provide a certain degree of robustness, but without any reference to a particular index or robustness metric. In other words, we can speak of unstructured uncertainty or robustness in the broad sense. These approaches attempt to formulate the tuning of the controller in terms of one or two parameters, so the choice of the user is to set the tradeoff between robustness and performance of the control system in terms of these direct parameters.

Thus, within the analytical methods, the IMC framework has won major acceptance. In the original presentation of its application to PI/PID controller design, in [44] a set of rules for different types of process models was proposed in which the three resulting controller parameters (PI or PID, depending on the model structure) are parameterized in terms of the constants of the process model and a single tuning parameter that has a direct connection with the control system robustness (the higher the value, the better the robustness), but inversely to the performance. Thus, in the case of a first-order model with delay (K, L, T), the tuning takes the following form [44]:

$$\begin{aligned} K_p &= \frac{2T + L}{K(2\lambda + L)}, \\ T_i &= T + L/2, \\ T_d &= \frac{TL}{2T + L}, \end{aligned} \tag{4.38}$$

where λ is the free-tuning parameter.

As shown, the equations are of low complexity and provide a systematic and simple way of dealing with robustness by means of the parameter λ although, as mentioned, without any robustness indicators related to the tuning parameter. This approach has led to a multitude of strategies. For a fairly comprehensive treatment of design based on IMC and its analysis of the effects on the robustness and performance, the following extensive work is available [40]. It must be said that strategies based on the designs by IMC are characterized primarily by the specification of the desired input–output relationship. Designs are therefore aimed at tracking more than regulation. Thus, it is useful to highlight the work [21], which is a completely analytical formulation, along the lines proposed by the original IMC, but specifying the desired relationship for the regulatory control. As a result, a set of tuning rules, depending on the model of the process and desired speed of response for disturbance rejection, is proposed. In contrast to the classical IMC rules in [44], the design relations appear quite complex but parameterized, as in the previous case, in terms of the variables of the process and the tuning parameter λ:

$$\begin{aligned} K_p &= \frac{1}{K}\frac{(2TL + L^2/2)(2\lambda + L/2) - \lambda^3 - 3\lambda^2 L}{2\lambda^3 + 3\lambda^2 L + (L/2)(3\lambda + L/2)}, \\ K_i &= \frac{(2TL + L^2/2)(2\lambda + L/2) - \lambda^3 - 3\lambda^2 L}{(2T + L)L}, \\ K_d &= \frac{3\lambda^2 TL + (TL^2/2)(3\lambda + L/2) - 2(T + L)\lambda^3}{(2TL + L^2/2)(2\lambda + L/2) - \lambda^3 - 3\lambda^2 L}. \end{aligned} \tag{4.39}$$

However, it is important to notice that, in both cases, the indication of robustness is given, indirectly, by the tuning parameter λ. As it is usual in IMC designs, a large value of λ means high robustness. This situation is the same for all variants of IMC designs that have appeared in the literature. Along these lines, there has recently been a growing interest in the derivation of simple rules that provide an acceptable level of robustness. In this regard, special attention is to be paid to the SIMC rule proposed in [47], based on a method of approximation of higher-order models ("the half rule") and a design aimed at improving control performance. Generally, the tuning methods developed with the IMC technique use pole/zero cancellation and, therefore, produce very slow responses to changes in the disturbance, especially when canceling the slow poles of the plant. Considering this, in this procedure the classical IMC tunings [44] are modified in order to provide fast response to changes in the set point and the disturbances. To tune a PI controller, a first-order model with delay is used, whereas for a PID, a second-order with delay is needed. In developing these rules an ideal PID controller, without derivative filter, is used. The SIMC tuning for PI and PID controllers in series form turns out to be:

$$\begin{aligned} K_p K &= \frac{1}{2\tau_o}, \\ \frac{T_i}{T} &= \min\{1, 8\tau_o\}, \\ \frac{T_d}{T} &= a. \end{aligned} \tag{4.40}$$

This tuning provides an average robustness level of $M_S \approx 1.6$ with 1DoF PI and PID controllers.

With a similar goal but from a completely different formulation, [55] provides a PID controller design for first-order plants with delay, based on a model matching specification in the $\mathcal{H}_\infty$ sense. Parameterized rules in terms of two variables, z and T_M, related to the robustness and speed of response, are provided (it should be noted that this tuning also provides the value of N for the derivative filter):

$$\begin{aligned} K_p &= \frac{T_i}{K(\rho + T_M)}, \\ T_i &= T + \chi_1 - T_M \frac{(\rho + z)}{(\rho + T_M)}, \\ \frac{T_d}{N} &= T_M \frac{(\rho + z)}{(\rho + T_M)} N, \\ N + 1 &= \frac{T}{T_i} \frac{\rho}{L} \frac{(\rho + T_M)}{(\rho + z)}, \end{aligned} \tag{4.41}$$

with $\chi_1 = z + L - \rho$ and $\rho = L(L + z)/(L + T_M)$. In the same work, specific values for z and T_M are proposed, generating a highly simple tuning:

$$\begin{aligned} K_p &= \frac{T_i}{2.65KL}, \\ T_i &= T + 0.03L, \\ \frac{T_d}{N} &= 1.72L, \\ N + 1 &= \frac{T}{T_i}, \end{aligned} \tag{4.42}$$

to guarantee a value of $M_S \approx 1.4$, which means a considerable robustness, while maintaining acceptable performance.

Based on this approach, similar formulations have appeared by introducing considerations about the improvement in the regulatory response, as a compromise between regulation and tracking performance, while ensuring a certain robustness. Thus, in [1] two model matching problems are proposed: one for the input–output relationship, specifying a desired relationship $T_d(s) = ((T_M - \gamma)s + 1)/(T_M s + 1)$, and one for the sensitivity function, such as $S_d(s) = \gamma s/(T_M s + 1)$. Both problems are solved analytically, and then, the impact of the choice of the parameters that specify the problem is analyzed. The T_M is chosen according to [55], while the value of γ is the one that determines the balance between regulation and tracking behavior. For example, in Fig. 4.3 it is possible to see the suggested range for the parameter λ (which plays the role of the filter time constant in an IMC design).

That way, by evaluating a compromise between performance loss and increase of robustness, a PID tuning rule is proposed. A variant of this extension can also be found in [2], while in [3] this methodology is presented within the design framework of $\mathcal{H}_\infty$ and the application extended to the case of unstable systems.

Building on these ideas, it is worth highlighting the Analytical Robust Tuning approach, known as ART$_2$, presented in [6] for PI controllers based on first-order models with delay and in [8] for PID controllers based on second-order models with delay. The design is based on the analytical formulation for the controller in regulation mode, while linked to a robust specification in terms of the value obtained

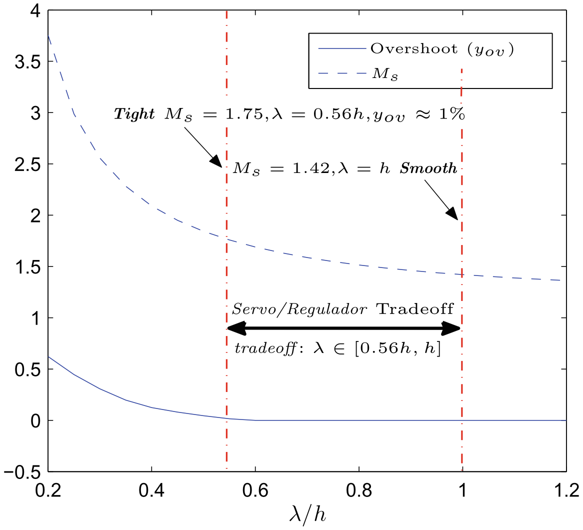


Fig. 4.3 Robustness and smooth/tight control in terms of the tuning parameter λ/h

for M_S. This approach, therefore, could be described both analytically and directed by M_S. We believe that it is important to describe this method in this section, as it results from the evolution analytical approaches that have experienced as a result of adding robustness considerations. As an example, for a PI controller, the design obtained takes the following form:

$$\begin{aligned} \kappa_p &\doteq K_p K = \frac{2\tau_c - \tau_c^2 + \tau_o}{(\tau_c + \tau_o)^2}, \\ \tau_i &\doteq \frac{T_i}{T} = \frac{2\tau_c - \tau_c^2 + \tau_o}{1 + \tau_o}, \end{aligned} \tag{4.43}$$

where $\tau_o = L/T$, and τ_c is the dimensionless design parameter (similar to the IMC time constant λ) that determines the relationship between the dominant time constant of the controlled process and the desired closed-loop response. The approach establishes a lower limit, $\tau_{c\,\min} \le \tau_c$, if it is desired to guarantee a certain level of robustness (in terms of M_S). The value $\tau_{c\,\min}$ is given in a parameterized form in terms of the desired robustness as:

$$\begin{aligned} \tau_{c\,\min} &= k_{11} + \left(\frac{k_{21}}{k_{22}}\right)\tau_o, \\ k_{11} &= 1.384 - 1.063M_S + 0.262M_S^2, \\ k_{21} &= -1.915 + 1.415M_S - 0.077M_S^2, \\ k_{22} &= 4.382 - 7.396M_S + 3.0M_S^2, \end{aligned} \tag{4.44}$$

therefore exposing, in a very compact way, a compromise between robustness and performance. The interesting point of this proposal is to link the effect of the design

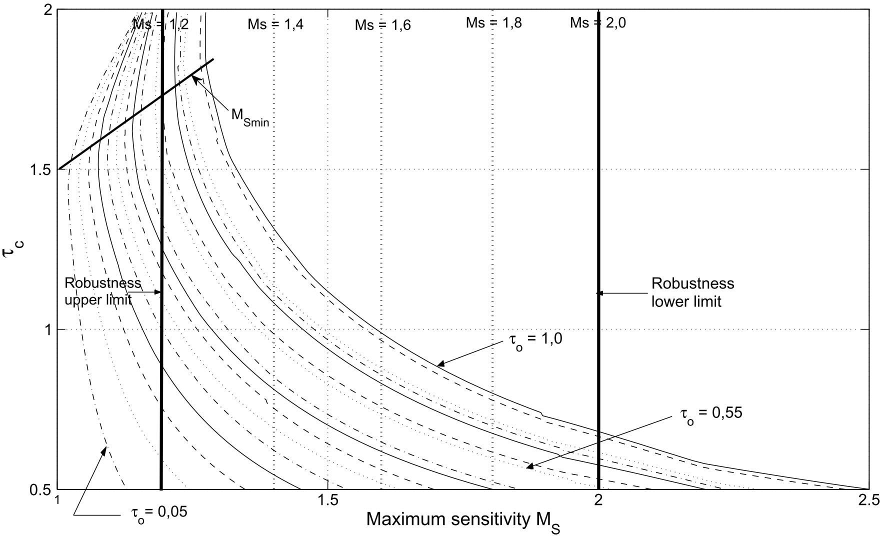


Fig. 4.4 Relationship between the robustness M_S and the design parameter τ_c

parameter (τ_c in this case) with the robustness measure M_S. In this way it is the M_S value itself which becomes a design parameter. Figure 4.4 shows the relationship for the case of the aforementioned PI controller and a first-order system with delay.

As it can be seen, if you increase too much the value of τ_c, you get the opposite effect of starting to lose robustness. Hence, the range of values for τ_c has also to be limited from above: $\tau_{c\,\min} \leq \tau_c \leq 1.5 + 0.3\tau_o$. The essentials of this method are shared with the Direct Synthesis design (DS-d) approach mentioned above in [21]. However, in the case of a PID controller both approaches do not lead to the same tuning. An additional distinction is made by the fact that in [8] the ART$_2$ method is formulated for a controller with two degrees of freedom (2DoF) and the ART$_2$ provides a design with overall better performance.

For comparison, we present the application of both methods to the following test system:

$$P(s) = \frac{1}{(s+1)(0.4s+1)(0.16s+1)(0.64s+1)}, \tag{4.45}$$

modeled by using the following second-order approximation [4]:

$$P_2(s) = \frac{e^{-0.147s}}{(0.856s+1)(0.603s+1)}. \tag{4.46}$$

Based on this model, the design relations of [8] and [21] are applied. The results are shown in Fig. 4.5 for two different robustness levels. Without getting into comparison details with numerical indicators (for this, [8] can be accessed), we can see the improvement made by the ART$_2$.

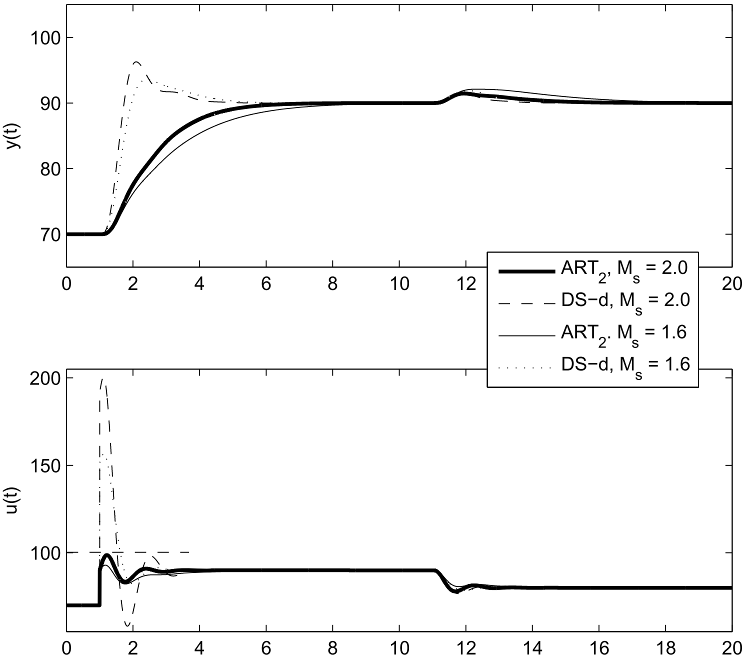


Fig. 4.5 Comparison between the ART$_2$ and DS-d methods

The fact of establishing a link between the design parameter and the robustness measure leads to a whole range of possibilities that may provide extensions of analytical design methods. This will allow the integration, in a much more explicit way, of the compromise between performance and robustness.

4.5 The Robustness/Performance *Tradeoff*

In the preceding sections we have presented a whole set of different options and alternatives that have been formulated along the years, in order to address the problem of the attainment of a robust PID. The common idea to all of them, despite the use of robustness measures and different uncertainty descriptions, is that they take into account the robustness characteristics of the resulting closed-loop. However, it is more and more obvious that there is a need to maintain high performance levels despite having to ensure a certain robustness, in other words, to *minimize performance loss due to the introduction of robustness considerations*. Therefore, the design of closed-loop control systems with PI/PID controllers should consider the constraint of two conflicting views: on one hand, the *performance* to changes in the set-point and disturbances and, on the other, the *robustness* to changes or uncertainty in the dynamics of the controlled process.

If only the system performance is taken into account, using, for example, an integrated error criteria (IAE, ITAE, or ISE) or a time response characteristic (overshoot, rise-time, or settling-time for example), as in [35, 50], the resulting closed-loop control system may have a very poor robustness. On the other hand, if the system is designed to have good robustness, as in [27], and if the performance of the resulting system is not evaluated, the designer will not have any indication of the *cost* of having such highly robust system. System performance and robustness were taken into account in [45, 51], optimizing its IAE or ITAE performance but guaranteeing only the usually accepted minimum level of robustness ($M_S = 2$).

On the other hand, among the extensive literature on PI/PID controller design, especially if we concentrate on contributions that provide PI/PID controller tuning rules, it is found that the approaches for PI/PID controller design heavily rely on the available process model. In fact, the PI/PID controller tuning literature can be broadly divided as providing tuning for FOPDT or SOPDT processes. In addition, it is also possible to find the corresponding FOPDT/PI and SOPDT/PID pairs. Even quite a few approaches exist that provide the PID tuning for a FOPDT controller. In any case, what is a true fact is that the number of tuning rules that can be found for FOPDT processes is really large compared with the very few proposals that exist for SOPDT processes. SOPDT models have richer dynamics and therefore are able to better represent the process to be controlled. It would therefore be better to use this extra information for controller design purposes, either for PI or PID. However, no unified approach has been already presented that considers at the same time PI and PID controllers as well as FOPDT and SOPDT process models.

This section is intended to provide one step in filling the previously identified gaps: to tackle the performance/robustness tradeoff with a unified FOPDT/SOPDT treatment. The presented approach provides a measure of the performance losses that the designer has to accept when imposing certain robustness constraints. Both specifications are usually thought of in an independent manner. However, they influence each other. It is in this sense that it is important to have an indication of the incurred performance losses when the control system robustness is increased. This analysis is conducted for 1DoF and 2DoF PI and PID control systems considering First- (FOPDT) and Second-Order-Plus-Dead-Time (SOPDT) process models. While the robustness of the resulting closed-loop is measured by using its Maximum Sensitivity (M_S) value, performance measure used is the Integrated-Absolute-Error (IAE) criteria. It can be found in the recent literature that IAE is the most useful and suitable index to quantify the performance of the system [16, 21, 38, 47, 49].

The analysis results allow us to obtain tuning relations for five robustness levels in the range $1.2 \leq M_S \leq 2.0$, to design robust closed-loop control systems that at the same time, have the best performance allowed under the IAE criteria. Therefore the method provides, in an unified way, a treatment of PI and PID controller design for either FOPDT and SOPDT process models.

The *performance* of the closed-loop control system will be evaluated with the Integrated-Absolute-Error (IAE) cost functional

$$J_e \doteq \int_0^\infty \left|e(t)\right| dt = \int_0^\infty \left|y(t) - r(t)\right| dt \tag{4.47}$$

for a step change in the set-point (J_{er}) and in the disturbance (J_{ed}).

As an indication of the system *robustness* (relative stability), the Sensitivity Function $S(j\omega)$ magnitude peak value will be used. If the system robustness is not taken into account at the design stage, the controller's parameters may be optimized to maximize the system performance, namely a minimum of (4.47); for both set-point (J_{er}^o) and load disturbance (J_{ed}^o) changes independently. However, due to the control system performance/robustness tradeoff, if a robustness requirement is included into the design, then it is expected that the actual system performance will be reduced ($J_e \geq J_e^o$). Then a *degraded performance factor*, F_p, defined as

$$F_p \doteq \frac{J_e^o}{J_e}, \quad F_p \leq 1, \tag{4.48}$$

will be used to evaluate the performance/robustness tradeoff.

4.5.1 Robustness/Performance **Tradeoff** *Analysis Framework*

In order to evaluate the performance/robustness tradeoff, several optimizations have to be performed. The next steps present, in an incremental way, the different sets of controller parameters that have to be obtained. In addition, for each controller, the cost functional values are also recorded in order to evaluate the performance degradation. The controller's parameters $\{K_p, T_i, T_d\}$ obtained are the ones required to have a system with a target robustness, M_S^t, stated according to the expected variation of the controlled process characteristics and, at the same time, with the best regulatory control performance allowed under the integrated absolute error criteria, J_{ed}. Using these parameters, the free proportional set-point weight factor, β, of the 2DoF PI and PID controllers were found optimizing the servo-control performance, J_{er}. Note that now the control system robustness does not change. The steps presented follow the procedure outlined in [5].

Optimum Performance is assessed by using the cost functional

$$J_e(\overline{\theta}) \doteq \int_0^\infty \left|y\left(t,\overline{\theta}\right) - r(t)\right| dt, \tag{4.49}$$

and the optimized performance servo and regulatory 1DoF PI and PID controllers' parameters $\overline{\theta}_o = \{K_{co}, T_{io}, T_{do}\}$ such that

$$J_e^o = J_e\left(\overline{\theta}_o\right) = \min_{\overline{\theta}} J_e\left(\overline{\theta}\right) \tag{4.50}$$

were obtained for (4.1) with $a \in \{0, 0.25, 0.5, 0.75, 1\}$ and ten normalized dead times τ_o in the range 0.05 to 2.0 for set-point and load disturbance step changes and target robustness levels $M_S^t \in [1.2, 2.0]$.

The robustness of the obtained optimal performance control systems was evaluated by using M_S. It has been found that only the servo-control PI (for all models) and servo-control PID (for SOPDT models with high normalized dead time) reach the standard minimum robustness allowed of $M_S = 2.0$. The results also show that, for the same model, the performance-optimized regulatory control systems are less robust than the corresponding optimized servo-control systems and that, for the same model and input (set-point or disturbance), the performance-optimized PI controllers are more robust than the corresponding PID controllers.

I Degraded Performance To increase the control-loop robustness to a target M_S^t, a degraded performance factor F_p^t was included into the cost functional

$$J_{F_p}\left(\overline{\theta}, F_p^t\right) \doteq \left|\frac{J_e^o}{J_e(\overline{\theta})} - F_d^t\right| \tag{4.51}$$

to obtain the PI and PID (servo and regulatory control) parameters $\overline{\theta}_{o1}$ such that

$$J_{F_p}^o = J_{F_p}\left(\overline{\theta}_{o1}, F_p^t\right) = \min_{\overline{\theta}} J_{F_p}\left(\overline{\theta}, F_p^t\right). \tag{4.52}$$

Therefore, by decreasing F_p^t, the control system is allowed to lose some performance and to increase the robustness. This fact can be used to increase the robustness to a desired target level M_S^t.

II Robust Performance At this point and starting from the parameters $\overline{\theta}_{o1}$ obtained before), a second optimization was conducted using the cost functional

$$J_{M_S}\left(\overline{\theta}, M_S^t\right) \doteq \left|M_S\left(\overline{\theta}\right) - M_S^t\right| \tag{4.53}$$

in order to reach the target robustness. The robust controller parameters $\overline{\theta}_{o2}$ are such that

$$J_{M_S}^o = J_{M_S}\left(\overline{\theta}_{o2}, M_S^t\right) = \min_{\overline{\theta}} J_{M_S}\left(\overline{\theta}, M_S^t\right). \tag{4.54}$$

For analysis, five robustness target levels were considered, $M_S^t \in \{2.0, 1.8, 1.6, 1.4, 1.2\}$.

III Performance Degradation Requirements Finally, the degraded performance factor required to obtain the target robustness $M_S^t \in \{2.0, \ldots, 1.2\}$ was evaluated as

$$F_p\left(M_S^t\right) \doteq \frac{J_e^o}{J_e(\overline{\theta}_{o2})}. \tag{4.55}$$

The controller's parameters $\overline{\theta}_{o2}$ are those required to have a system with a target robustness M_S^t, stated according to the expected variation of the controlled process characteristics and, at the same time, with the best performance allowed under the integrated absolute error criteria (4.47).

IV 2DoF PI/PID Proportional Set-Point Weight Using the 1DoF PI and PID regulatory control parameters for robust performance $\overline{\theta}_{o2}$, the free proportional set-point weight factors β of the 2DoF PI and PID controllers were found optimizing (4.47) for a set-point step change (J_{er}). Note that now the control system robustness does not change.

The final result of these optimization steps is a set of controller parameters (not a tuning rule) that in what follows will be referred as the *RoPe* (*Ro*bust *Pe*rformance) tuning as it emerges from considering the necessary performance reduction in order to attain a certain robustness level. As an example, Figs. 4.6 and 4.7 illustrate the evolution of the performance degradation factor (F_p). Not all the possibilities are shown.

4.5.2 Robust Performance Controllers Analysis

As indicated, fifty models were used in the servo and regulatory PI and PID control performance/robustness tradeoff analysis with five robustness levels, obtaining a total of 1500 controller's parameters sets. A short analysis of the existing tradeoff for different combinations of controller/operation according to PI/PID and servo/regulation now follows.

PI Servo-Control The required performance degradation increases with the model normalized dead-time, but the model time constants ratio a has practically no influence on it, i.e., PI control systems for FOPDT and DPPDT models with same normalized dead-time require similar performance degradation to increase its robustness to the same level. For example, increasing the robustness of a FOPDT PI servo-control to $M_S = 1.8$ has a marginal cost, but increasing it to $M_S = 1.2$ reduces the control system performance 60% ($F_p \approx 0.4$) as shown in Fig. 4.6. PI servo-control systems losses range from 10 to 20% of their optimum performance if $M_S^t = 1.6$ is specified.

PI Regulatory-Control For the regulatory control case, the required performance degradation decreases as the model normalized dead-time increases. In the regulatory control case a higher performance degradation is required, compared with the servo-control case, to reach the same robustness level.

While, for example, the SOPDT ($a = 0.5$, $\tau_o = 1$) PI regulatory control requires $F_p = 0.64$ (36% reduction in performance) for $M_S^t = 1.4$, its servo-control counterpart requires $F_p = 0.51$ (49% performance reduction). Therefore, we have a measure for the price to pay for robustness.

PID Servo-Control Comparing the PI and PID servo-control degradation requirements, it was noted that the PI control systems are more robust than the PID. In addition, the model normalized dead-time influence is reversed. To have a robustness level corresponding to $M_S^t = 1.6$, the DPPDT model PID servo-control requires $F_p = 0.37$ ($\tau_o = 0.10$) and $F_p = 0.82$ ($\tau_o = 1.5$) (Fig. 4.7), while the corresponding PI only $F_p = 0.94$ ($\tau_o = 0.10$) and $F_p = 0.86$ ($\tau_o = 1.5$).

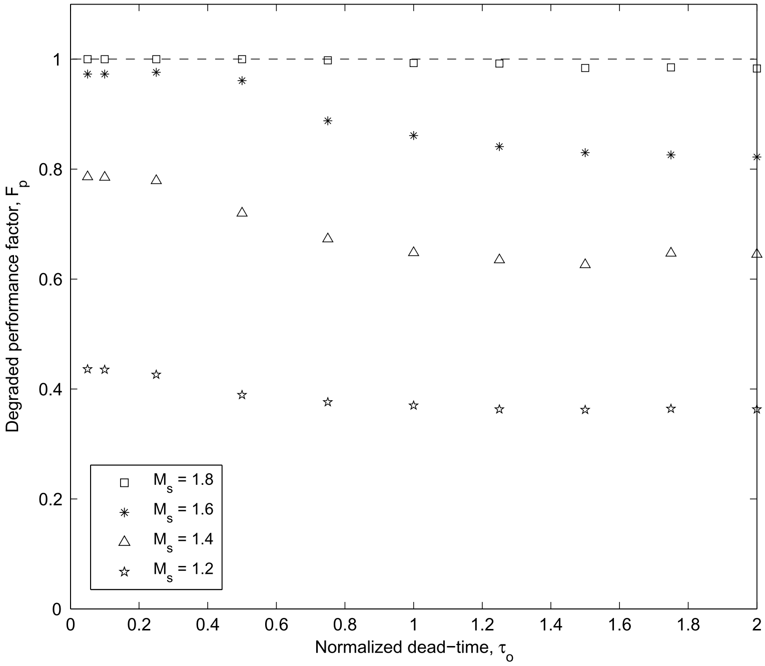


Fig. 4.6 PI Servo-Control, FOPDT model ($a = 0.0$)

PID Regulatory-Control The results in this case show that the regulatory PID control systems are less robust than the regulatory PI controllers and also less robust than the servo-control PIDs.

In general, performance-optimized PI control systems are more robust than the PID ones, and the PI and PID servo-control systems are more robust than their regulatory control counterparts.

4.5.3 Illustrative Example

This section provides a simulation study to show that the presented rules provide robustness and performance comparable (if not better) to other specific tunings existing in the literature. The main purpose is not to compare time responses but to exemplify the robustness/performance tradeoff.

Consider the controlled process proposed as benchmark in [14] and given by the transfer function

$$P(s) = \frac{1}{(s+1)(\alpha s+1)(\alpha^2 s+1)(\alpha^3 s+1)} \tag{4.56}$$

with $\alpha = 0.5$.

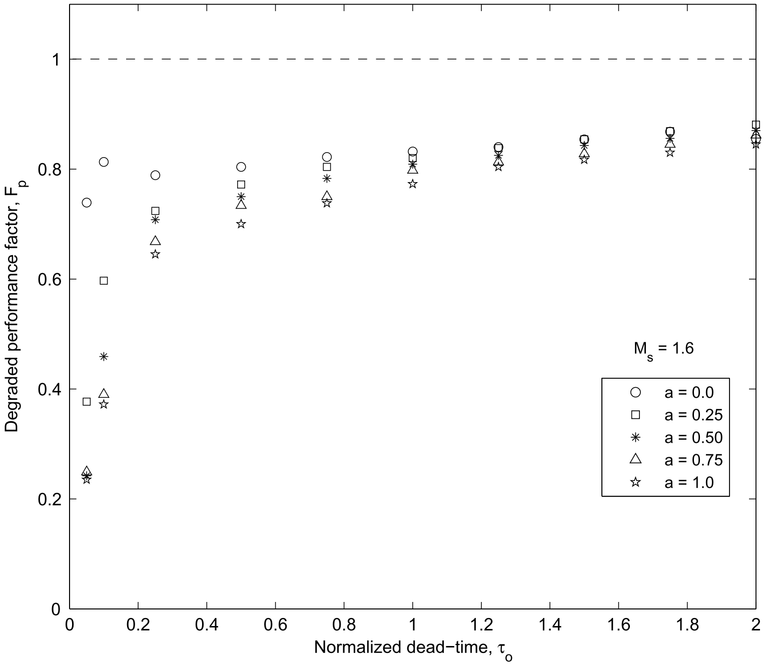


Fig. 4.7 PID Servo-Control, $M_s^t = 1.6$

Table 4.3 Example—models

Model	K	T	a	L	τ_o
FOPDT	1	1.247	0	0.691	0.554
SOPDT	1	0.876	0.821	0.277	0.316

Using the three-point identification method *123c* [4], FOPDT and SOPDT models were obtained with parameters shown in Table 4.3.

From these models and using the tuning provided by the previously presented optimization procedure, the parameters for 2DoF PI and PID controllers, Tables 4.4 and 4.5, were obtained for four levels of robustness. For the second-order model, linear interpolations for the time constants ratio included in the analysis were made to consider the particular ratio of this example.

For comparison purposes, the SIMC ("Simple Control") [47], the PO ("Percent Overshoot") [10], and the MOO ("Multi-Objective Optimization") [52] tuning methods were considered. The SIMC is an IMC-based tuning method for 1DoF PI and PID controllers to obtain a good tradeoff between speed of response, disturbance rejection, robustness, and control effort requirements. The PO method using a relation between the set-point step response overshoot (OS) and the closed-loop robustness provides tuning equations for 1DoF PI and PID controllers for a smooth control POs (0% OS, $M_S = 1.38$) and tight control POt (10% OS, $M_S = 1.71$). Fi-

Table 4.4 Example—RoPe PI_2 parameters

	M_S^t							
	2.0	1.6	1.4	1.2	2.0	1.6	1.4	1.2
	From FOPDT Model				From SOPDT Model			
K_p	1.396	0.943	0.750	0.362	1.914	1.326	0.976	0.525
T_i	1.394	1.382	1.392	1.418	1.966	1.966	1.954	1.973
β	0.859	1.154	1.456	2.372	0.934	1.037	1.165	1.456

Table 4.5 Example—RoPe PID_2 parameters

	M_S^t							
	2.0	1.6	1.4	1.2	2.0	1.6	1.4	1.2
	From FOPDT Model				From SOPDT Model			
K_p	1.705	1.176	0.935	0.516	3.753	2.155	1.681	1.015
T_i	0.939	0.962	0.978	0.999	0.817	0.804	0.937	1.020
T_d	0.292	0.303	0.293	0.301	0.461	0.395	0.395	0.404
β	0.620	0.842	1.059	1.733	0.573	0.598	0.715	1.148

Table 4.6 PI/PID parameters

Method	K_p	T_i	T_d	β
SIMC PI	0.902	1.247	0	1
SIMC PID	2.879	1.595	0.395	1
POs PI	0.722	1.247	0	1
POt PI	1.029	1.247	0	1
POs PID	2.334	1.595	0.395	1
POt PID	3.314	1.595	0.395	1
MOO	0.987	1.178	0	0.746

nally, the MOO considers the tradeoff between servo and regulatory fast response to step changes (IAE), a smooth control (TV_u), and robustness (M_S) for 2DoF PI controllers. The various controller parameters are shown in Table 4.6.

The performance and robustness shown in Table 4.7 for the different tuning rules above are to be compared with those for the proposed PIs and PIDs on the basis of the achieved robustness (M_S^m) and the model the tuning is based on.

It is worth mentioning that the regulatory control operation is the one that really matters. In fact, the normal operation of the control loop is on maintaining the operation point, therefore ensuring good disturbance rejection. It is in this sense that the PI/PID parameters were optimized for such an operational mode. The second degree of freedom (β) was optimized in a second stage, with the other parameters constant, to minimize the servo-control integrated absolute error performance.

The results of the *RoPe* tuning are shown in Tables 4.8, 4.9, 4.10 and 4.11. For each situation, the achieved performance is provided for the established robustness

Table 4.7 Performance/robustness

Method	J_{ed}	J_{er}	TV_{ud}	TV_{ur}	M_S^m
SIMC PI	1.590	1.382	1.815	1.176	1.530
SIMC PID	1.183	0.554	5.446	1.186	1.600
POs PI	1.727	1.746	1.036	1.337	1.438
POt PI	1.220	1.557	1.295	2.207	1.715
POs PID	0.683	1.233	1.058	4.164	1.555
POt PID	0.481	1.217	1.301	5.770	1.955
MOO	1.234	1.658	1.316	1.794	1.703

Table 4.8 RoPe PI$_2$ (FOPDT) performance/robustness

M_S^t	J_{ed}	J_{er}	TV_{ud}	TV_{ur}	M_S^m
2.0	1.007	1.483	1.555	2.920	2.101
1.6	1.466	1.509	1.126	2.039	1.596
1.4	1.856	1.550	1.002	1.807	1.427
1.2	3.917	1.972	1.000	1.088	1.178

Table 4.9 RoPe PI$_2$ (SOPDT) performance/robustness

M_S^t	J_{ed}	J_{er}	TV_{ud}	TV_{ur}	M_S^m
2.0	1.027	1.437	1.777	4.638	2.016
1.6	1.483	1.475	1.226	2.723	1.605
1.4	2.002	1.680	1.016	1.914	1.414
1.2	3.758	2.859	1.000	1.099	1.203

levels. The comparison with the other tunings should start from the results in Table 4.7.

Take, for example, the SIMC PID (based on a SOPDT process model). The comparison should be with the proposed PID$_2$ (SOPDT), Table 4.11, choosing the tuning corresponding to $M_S^t = 1.6$. It may be seen that the robustness is slightly higher while achieving at the same time a dramatic improvement in the regulatory control performance.

Note that, in addition, the tunings SIMC, POs, POt, and MOO provide a robustness that is completely determined by the tuning. It cannot be modulated according to the user demand. In contrast, *RoPe* can achieve an optimized performance for all the range of robustness specifications. This fact adds flexibility to the tuning, giving, at the same time, more generality to this tuning.

For this particular process (4.56), it can be seen that for same robustness level, the best regulatory performance is obtained with the *RoPe* PID$_2$ tuned with the SOPDT model.

Table 4.10 RoPe PID$_2$ (FOPDT) performance/robustness

M_S^t	J_{ed}	J_{er}	TV_{ud}	TV_{ur}	M_S^m
2.0	0.760	1.630	1.573	2.968	2.030
1.6	1.041	1.758	1.354	2.265	1.554
1.4	1.256	1.817	1.253	2.008	1.390
1.2	2.041	1.960	1.061	1.442	1.199

Table 4.11 RoPe PID$_2$ (SOPDT) performance/robustness

M_S^t	J_{ed}	J_{er}	TV_{ud}	TV_{ur}	M_S^m
2.0	0.313	1.310	1.613	5.430	2.394
1.6	0.611	1.682	1.686	3.976	1.501
1.4	0.757	1.652	1.399	2.811	1.364
1.2	1.198	1.806	1.200	2.088	1.220

4.6 Conclusions and Perspectives

This chapter has outlined different perspectives and formulations for the design of a robust PID controller. The spectrum of approaches that can be found in the literature is really broad, and, in fact, the inclusion of robustness considerations has motivated the appearance of a large number of papers in recent years. This shows once more the continued interest on the subject.

In recent publications the situation has turned attention to the use of specific robustness measures as design specifications that are incorporated as integral parts of the PID controller design. In turn, this enables us to analyze the tradeoff between robustness and other control-loop properties, such as those related to the control system performance. To obtain a robust design is not just the target; you must have a precise idea of the price paid by demanding certain levels of robustness. In this sense, as it has been noted in the last section, an analysis should be pursued in order to find ways allowing a clear statement of the existing performance/robustness balance.

We can also foresee that another aspect that will deserve attention in the future will be that of guaranteeing the targeted robustness characteristics [56]. Advanced PID controller designs should not only incorporate a robustness design specification as an integral part of the design but to somehow guarantee this specification to be met. This is needed in order to allow the previously mentioned robustness/performance tradeoff analysis. Otherwise, it does not make sense to perform any comparison.

Another aspect that also deserves attention for future developments is the controller's fragility. This aspect has been introduced recently into the literature as an analysis tool. However, in order to avoid possible robustness and performance losses, it should be also introduced at the design stage. This is a new challenge that should distinguish new PID controller tunings.

Acknowledgements This work has received financial support from the Spanish CICYT program under grant DPI2010-15230. Also, the financial support from the University of Costa Rica and from the MICIT and CONICIT of the Government of the Republic of Costa Rica is greatly appreciated.

References

1. Alcántara, S., Pedret, C., Vilanova, R.: On the model matching approach to PID design: analytical perspective for robust servo/regulator tradeoff tuning. J. Process Control **20**, 596–608 (2010)
2. Alcántara, S., Pedret, C., Vilanova, R., Zhang, W.D.: Simple analytical min-max model matching approach to robust proportional-integrative-derivative tuning with smooth set-point response. Ind. Eng. Chem. Res. **49**, 690–700 (2010)
3. Alcántara, S., Zhang, W.D., Pedret, C., Vilanova, R., Skogestad, S.: IMC-like analytical $\mathcal{H}_\infty$ design with S/SP mixed sensitivity consideration: utility in PID tuning guidance. J. Process Control **21**, 554–563 (2011)
4. Alfaro, V.M.: Low-order models identification from process reaction curve. Cienc. Tecnol. **24**(2), 197–216 (2006) (in Spanish)
5. Alfaro, V.M., Méndez, V., Vilanova, R., Lafuente, J.: Performance/Robustness tradeoff analysis of PI/PID servo and regulatory control systems. In: IEEE International Conference on Industrial Technology (ICIT2010), Viña del Mar, Valparaiso, Chile, 14–17 March 2010
6. Alfaro, V.M., Vilanova, R., Arrieta, O.: Analytical robust tuning of PI controllers for first-order-plus-dead-time processes. In: 13th IEEE International Conference on Emerging Technologies and Factory Automation, Hamburg, Germany, 15–18 September 2008
7. Alfaro, V.M., Vilanova, R., Arrieta, O.: NORT: a non-oscillatory robust tuning approach for 2-DoF PI controllers. In: 18th International Conference on Control Applications part of the IEEE Multi-Conference on Systems and Control (CCA2009), Saint Petersburg, Russia, 8–9 July 2009
8. Alfaro, V.M., Vilanova, R., Arrieta, O.: A single-parameter robust tuning approach for two-degree-of-freedom PID controllers. In: European Control Conference (ECC09), Budapest, Hungary, pp. 1788–1793, 23–26 August 2009
9. Alfaro, V.M., Vilanova, R., Arrieta, O.: Maximum sensivity based robust tuning for two-degree-of-freedom proportional-integral controllers. Ind. Eng. Chem. Res. **49**, 5415–5423 (2010)
10. Ali, A., Mahi, S.: PI/PID controller design based on IMC and percentage overshoot specification to controller setpoint change. ISA Trans. **48**, 10–15 (2009)
11. Arrieta, O., Vilanova, R.: Simple PID tuning rules with guaranteed M_s robustness achievement. In: 18th IFAC World Congress, Milano, Italy, 28 August–2 September 2011
12. Åström, K.J., Hägglund, T.: Automatic tuning of simple regulators with specifications on phase and amplitude margin. Automatica **20**, 645–651 (1984)
13. Åström, K.J., Hägglund, T.: PID Controllers: Theory, Design, and Tuning. Instrument of Society of America (1995)
14. Åström, K.J., Hägglund, T.: Benchmark systems for PID control. In: IFAC Digital Control: Past, Present and Future of PID Control (PID'00), Terrassa, Spain (2000)
15. Åström, K.J., Hägglund, T.: Revisiting the Ziegler–Nichols step response method for PID control. J. Process Control **14**, 635–650 (2004)
16. Åström, K.J., Hägglund, T.: Advanced PID Control. ISA—The Instrumentation, Systems, and Automation Society (2006)
17. Åström, K.J., Panagopoulos, H., Hägglund, T.: Design of PI controllers based on non-convex optimization. Automatica **34**, 585–601 (1998)
18. Babb, M.: Pneumatic instruments gave birth to automatic control. Control Eng. **37**(12), 20–22 (1990)

19. Bar-on, J.R., Jonckheere, E.A.: Phase margins for multivariable control systems. Int. J. Control **52**(2), 485–498 (1998)
20. Bennett, S.: The past of PID controllers. In: IFAC Digital Control: Past, Present and Future of PID Control, Terrassa, Spain (2000)
21. Chen, D., Seborg, D.E.: PI/PID controller design based on direct synthesis and disturbance rejection. Ind. Eng. Chem. Res. **41**(19), 4807–4822 (2002)
22. Chien, I.L., Fruehauf, P.S.: Consider IMC tuning to improve performance. Chem. Eng. Prog. **86**(10), 33–41 (1990)
23. Chien, I.L., Hrones, J.A., Reswick, J.B.: On the automatic control of generalized passive systems. Trans. ASME **74**, 175–185 (1952)
24. Cohen, G.H., Coon, G.A.: Theoretical considerations of retarded control. Trans. Am. Soc. Mech. Eng. **75**, 827–834 (1953)
25. Dahlin, E.G.: Designing and tuning digital controllers. Instrum. Control Syst. **41**(6), 77–81 (1968)
26. Grimble, M.J.: Robust Industrial Control. Optimal Design Approach for Polynomial Systems. Prentice-Hall International, Englewood Cliffs (1994)
27. Hägglund, T., Åström, K.J.: Revisiting the Ziegler–Nichols tuning rules for PI control. Asian J. Control **4**, 354–380 (2002)
28. Hägglund, T., Åström, K.J.: Revisiting the Ziegler–Nichols tuning rules for PI control—Part II, the frequency response method. Asian J. Control **6**, 469–482 (2004)
29. Herreros, A., Baeyens, E., Peran, J.R.: Design of PID-type controllers using multiobjective genetic algorithms. ISA Trans. **41**(4), 457–472 (2002)
30. Ho, W.K., Gan, O.P., Tay, E.B., Ang, E.L.: Performance and gain and phase margins of well-known PID tuning formulas. IEEE Trans. Control Syst. Technol. **4**(11), 473–477 (1996)
31. Ho, W.K., Hang, C.C., Cao, L.S.: Tuning PID controllers based on gain and phase margin specifications. Automatica **31**(3), 497–502 (1995)
32. Ho, W.K., Hang, C.C., Zhou, J.H.: Performance and gain and phase margins of well-known PI tuning formulas. IEEE Trans. Control Syst. Technol. **3**(2), 245–248 (1995)
33. Ho, W.K., Lee, T.H., Gan, O.P.: Tuning of multiloop PID controllers based on gain and phase margin specifications. Ind. Eng. Chem. Res. **36**(6), 2231–2238 (1997)
34. Ho, W.K., Lim, K.L., Hang, C.C., Ni, L.Y.: Getting more phase margin and performance out of PID controllers. Automatica **35**, 1579–1585 (1999)
35. Huang, H.-P., Jeng, J.-C.: Monitoring and assessment of control performance for single loop systems. Ind. Eng. Chem. Res. **41**, 1297–1309 (2002)
36. Johnson, M., Moradi, M.H.: PID Control: New Identification and Design Methods. Springer, London (2005)
37. Kristiansson, B.: PID controllers, design and evaluation. PhD thesis, Control and Automation Laboratory, Department of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden (2003)
38. Kristiansson, B., Lennartson, B.: Evaluation and simple tuning of PID controllers with high-frequency robustness. J. Process Control **16**, 91–102 (2006)
39. Kristiansson, B., Lennartson, B.: Robust tuning of PI and PID controllers. IEEE Control Syst. Mag. **26**(1), 69 (2006)
40. Leva, A., Colombo, A.M.: On the IMC-based synthesis of the feedback block of ISA PID regulators. Trans. Inst. Meas. Control **26**(5), 417–440 (2004)
41. Morari, M., Zafiriou, E., Economou, C.G.: Robust Process Control. Springer, Berlin (1988)
42. O'Dwyer, A.: Handbook of PI and PID Controller Tuning Rules, 3rd edn. Imperial College Press, London (2009)
43. Persson, P.: Towards autonomous PID control. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden (1992)
44. Rivera, D.E., Morari, M., Skogestad, S.: Internal model control 4. PID controller design. Ind. Eng. Chem. Res. **25**, 252–265 (1986)
45. Shen, J.-C.: New tuning method for PID controller. ISA Trans. **41**, 473–484 (2002)
46. Shinskey, F.G.: Process Control Systems: Application, Design, and Tuning. 3rd edn. McGraw-Hill, New York (1988)

47. Skogestad, S.: Simple analytic rules for model reduction and PID controller tuning. J. Process Control **13**, 291–309 (2003)
48. Smith, A.C., Corripio, B.A.: Principles and Practice of Automatic Process Control. Wiley, New York (1985)
49. Tan, W., Liu, J., Chen, T., Marquez, H.J.: Comparison of some well-known PID tuning formulas. Comput. Chem. Eng. **30**, 1416–1423 (2006)
50. Tavakoli, S., Tavakoli, M.: Optimal tuning of PID controllers for first order plus time delay models using dimensional analysis. In: The Fourth International Conference on Control and Automation (ICCA'03), Montreal, Canada, June 2003
51. Tavakoli, S., Griffin, I., Fleming, P.J.: Robust PI controller for load disturbance rejection and setpoint regulation. In: IEEE Conference on Control Applications, Toronto, Canada, 28–31 August 2005
52. Tavakoli, S., Griffin, I., Fleming, P.J.: Multi-objetive optimization approach to the PI tuning problem. In: IEEE Congress on Evolutionary Computing (CEC2007), pp. 3165–3171 (2007)
53. Toivonen, H.T., Totterman, S.: Design of fixed-structure controllers with frequency-domain criteria: a multiobjective optimisation approach. IEE Proc. D, Control Theory Appl. **153**(1), 46–52 (2006)
54. Vidyasagar, M.: Control System Synthesis. A Factorization Approach. MIT Press, Cambridge (1985)
55. Vilanova, R.: IMC based robust PID design: tuning guidelines and automatic tuning. J. Process Control **18**, 61–70 (2008)
56. Vilanova, R., Alfaro, V.M., Arrieta, O., Pedret, C.: Analysis of the claimed robustness for PI/PID robust tuning rules. In: 18th IEEE Mediterranean Conference on Control and Automation (MED10), Marrakech, Morocco, 23–25 June 2010
57. Visioli, A.: Optimal tuning of PID controllers for integral and unstable processes. IEE Proc., Control Theory Appl. **148**(2), 180–184 (2001)
58. Visioli, A.: Practical PID Control. Advances in Industrial Control Series. Springer, Berlin (2006)
59. Wang, Q.-G.: Decoupling Control. Springer, New York (2003)
60. Wang, Q.-G., He, Y., Ye, Z., Lin, C., Hang, C.-C.: On loop phase margins of multivariable control systems. J. Process Control **18**(2), 202–211 (2008)
61. Wang, Q.-G., Ye, Z., Cai, W.-J., Hang, C.-C.: PID Control for Multivariable Processes. Lecture Notes in Control and Information Sciences. Springer, Berlin (2008)
62. Zames, G., Francis, B.A.: Feedback, minmax sensitivity and optimal robustness. IEEE Trans. Autom. Control **28**, 585–601 (1983)
63. Zhuang, M., Atherton, D.P.: Automatic tuning of optimum PID controllers. IEE Proc. Part D. Control Theory Appl. **140**(3), 216–224 (1993)
64. Ziegler, J.G., Nichols, N.B.: Optimum settings for automatic controllers. Trans. Am. Soc. Mech. Eng. **64**, 759–768 (1942)

Chapter 5
The SIMC Method for Smooth PID Controller Tuning

Sigurd Skogestad and Chriss Grimholt

5.1 Introduction

Although the proportional-integral-derivative (PID) controller has only three parameters, it is not easy, without a systematic procedure, to find good values (settings) for them. In fact, a visit to a process plant will usually show that a large number of the PID controllers are poorly tuned. The tuning rules presented in this chapter have developed mainly as a result of teaching this material, where there are several objectives:

1. The tuning rules should be well motivated, and preferably model-based and analytically derived.
2. They should be simple and easy to memorize.
3. They should work well on a wide range of processes.

In this paper the simple two-step SIMC procedure [11] that satisfies these objectives is summarized:

Step 1. Obtain a first- or second-order plus delay model.
Step 2. Derive model-based controller settings. PI-settings result if we start from a first-order model, whereas PID-settings result from a second-order model.

The SIMC method is based on classical ideas presented earlier by Ziegler and Nichols [17], the IMC PID-tuning paper by Rivera et al. [8], and the closely related direct synthesis tuning rules in the book by Smith and Corripio [13]. The Ziegler–Nichols settings result in a very good disturbance response for integrating processes but are otherwise known to result in rather aggressive settings [2, 15] and also to give poor performance for processes with a dominant delay. On the other hand, the analytically derived IMC-settings of Rivera et al. [8] are known to result in poor

S. Skogestad (✉) · C. Grimholt
Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway
e-mail: skoge@ntnu.no

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_5, © Springer-Verlag London Limited 2012

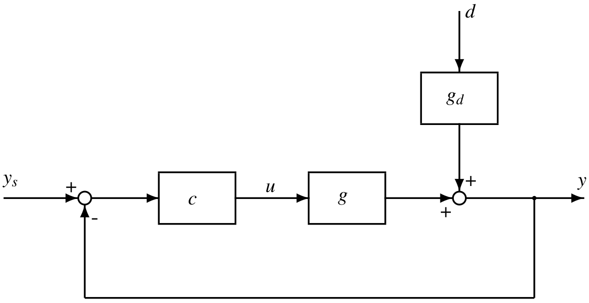


Fig. 5.1 Block diagram of feedback control system. In this chapter we consider an input ("load") disturbance ($g_d = g$)

disturbance response for integrating processes [3, 7] but are robust and generally give very good responses for setpoint changes. The SIMC tuning rule presented in this chapter works well for both integrating and pure time delay processes and for both setpoints and load disturbances.

This chapter provides a summary of the original SIMC method and provides some new results on obtaining the model from closed-loop data and on the Pareto-optimality of the SIMC method. There is some room for improvement for delay-dominant processes, and at the end of the chapter "improved" SIMC rules are presented.

The notation is summarized in Fig. 5.1. Here u is the manipulated input (controller output), d the disturbance, y the controlled output, and y_s the setpoint (reference) for the controlled output. $g(s) = \frac{\Delta y}{\Delta u}$ denotes the process transfer function, and $c(s)$ is the feedback part of the controller. Note that all the variables u, d, and y are deviations from the initial steady state, but the Δ used to indicate deviation variables is usually omitted. Similarly, the Laplace variable s is often omitted to simplify notation. The settings given in this chapter are for the series (cascade, "interacting", classical) form PID controller:

$$\text{Series PID:} \quad c(s) = K_c \cdot \left(\frac{\tau_I s + 1}{\tau_I s} \right) \cdot (\tau_D s + 1) = \frac{K_c}{\tau_I s} \big(\tau_I \tau_D s^2 + (\tau_I + \tau_D) s + 1 \big) \tag{5.1}$$

where K_c is the controller gain, τ_I the integral time, and τ_D the derivative time. The reason for using the series form is that the PID rules with derivative action are then much simpler. The corresponding settings for the ideal (parallel form) PID controller are easily obtained using (5.30).

The following practical PID controller (series form) is used in the simulations:

$$u(s) = K_c \left(\frac{\tau_I s + 1}{\tau_I s} \right) \left(y_s(s) - \frac{\tau_D s + 1}{(\tau_D / N) s + 1} y(s) \right) \tag{5.2}$$

with $N = 10$. Note that we in order to avoid "derivative kick," do not differentiate the setpoint in (5.2). In most cases we use PI-control, i.e., $\tau_D = 0$, and the above implementation issues and differences between series and ideal form do not apply.

5.2 Model Approximation (Step 1)

The first step in the SIMC design procedure is to obtain an approximate first- or second-order time delay model on the form

$$g_1(s) = \frac{k}{\tau_1 s + 1} e^{-\theta s} = \frac{k'}{s + 1/\tau_1} e^{-\theta s}, \tag{5.3}$$

$$g_2(s) = \frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)} e^{-\theta s}. \tag{5.4}$$

Thus, we need to estimate the following model information:

- Plant gain, k
- Dominant lag time constant, τ_1
- (Effective) time delay (dead time), θ
- Optional: Second-order lag time constant, τ_2 (for dominant second-order process for which $\tau_2 > \theta$, approximately)

Such data may be obtained in many ways, three of which are discussed below.

1. From open-loop step response
2. From closed-loop setpoint response with P-controller
3. From detailed model: Approximation of effective delay using the half rule

5.2.1 Model from Open-Loop Step Response

In practice, the model parameters for a first-order model are commonly obtained from a step response experiment as shown in Fig. 5.2. From a theoretical point of view this may not be the most effective method, but it has the advantage of being very simple to use and interpret.

For plants with a large time constant τ_1, one has to wait a long time for the process to settle. Fortunately, it is generally not necessary to run the experiment for longer than about 10 times the effective delay (θ). At this time, one may simply stop the experiment and either extend the response "by hand" toward settling or approximate it as an integrating process (see Fig. 5.3),

$$\frac{ke^{-\theta s}}{\tau_1 s + 1} \approx \frac{k' e^{-\theta s}}{s} \tag{5.5}$$

where

- Slope, $k' \overset{\text{def}}{=} k/\tau_1$

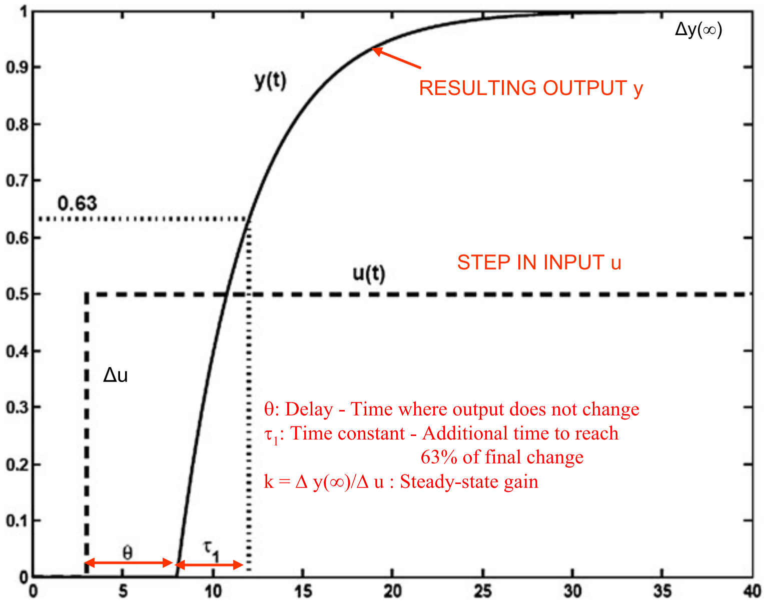
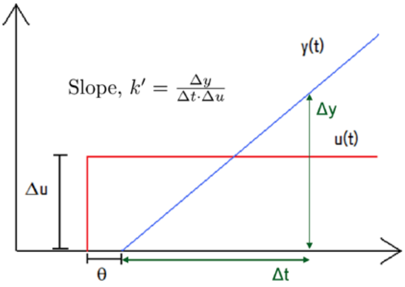


Fig. 5.2 Open-loop step response experiment to obtain parameters k, τ_1, and θ in first-order model (5.3)

Fig. 5.3 Open-loop step response experiment to obtain parameters k' and θ in integrating model (5.5)



is the slope of the integrating response. The reason is that for lag-dominant processes, i.e., for $\tau_1 > 8\theta$ approximately, the individual values of the time constant τ_1 and the gain k are not very important for controller design. Rather, their ratio k' determines the PI-settings, as is clear from the SIMC tuning rules presented below.

5.2.2 Model from Closed-Loop Setpoint Response

In some cases, open-loop responses may be difficult to obtain, and using closed-loop data may be more effective. The most famous closed-loop experiment is the Ziegler–Nichols where the system is brought to sustained oscillations by use of a P-only controller. One disadvantage with the method is that the system is brought to its

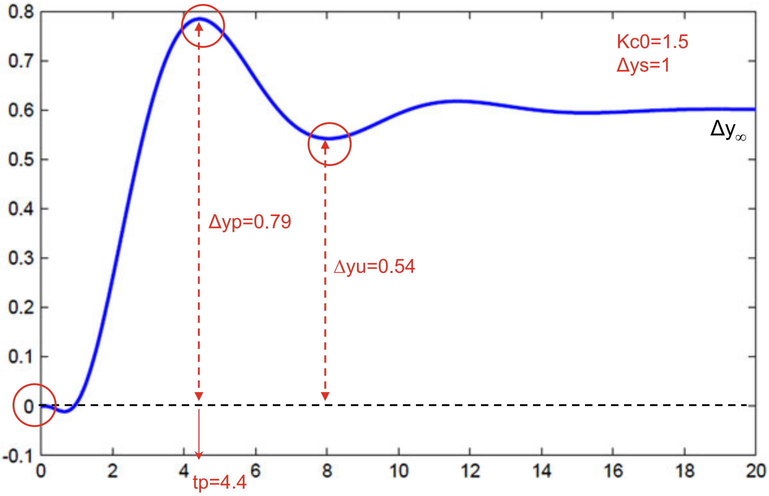


Fig. 5.4 Extracting information from closed-loop setpoint response with P-only controller

instability limit. Another disadvantage is that it does not work for a simple second-order process. Finally, only two pieces of information are used (the controller gain K_u and the ultimate period P_u), so the method cannot possibly work on a wide range of first-order plus delay processes, which we know are described by three parameters (k, τ_1, θ).

Yuwana and Seborg [16], and more recently Shamsuzzoha and Skogestad [10], proposed a modification to the Ziegler–Nichols closed-loop experiment, which does not suffer from these three disadvantages. Instead of bringing the system to its limit of stability, one uses a P-controller with a gain that is about half this value, such that the resulting overshoot (D) to a step change in the setpoint is about 30% (that is, D is about 0.3).

We here describe the procedure proposed by Shamsuzzoha and Skogestad [10], which seems to use the most easily available parameters from the closed-loop response. The system should be at steady state initially, that is, before the setpoint change is applied. Then, from the closed-loop setpoint response one obtains the following parameters (see Fig. 5.4):

- Controller gain used in experiment, K_{c0}.
- Setpoint change, Δy_s.
- Time from setpoint change to reach first (maximum) peak, t_p.
- Corresponding maximum output change, Δy_p.
- Output change at first undershoot, Δy_u.

This seems to be the information that is most easy (and robust) to observe directly, without having to record and analyze all the data before finding the parameters. Also note that one may stop the experiment already at the first undershoot.

The undershoot Δy_u is used to estimate the steady-state output change (at infinite time) [10],

$$\Delta y_\infty = 0.45(\Delta y_p + \Delta y_u). \tag{5.6}$$

Alternatively, if one has time to wait for the experiment to settle, one may record Δy_∞ instead of Δy_u.

From this information one computes the relative overshoot and the absolute value of the relative steady-state offset, defined by:

- Overshoot, $D = \frac{\Delta y_p - \Delta y_\infty}{\Delta y_\infty}$.
- Steady-state offset, $B = |\frac{\Delta y_s - \Delta y_\infty}{\Delta y_\infty}|$.

Shamsuzzoha and Skogestad [10] use this information to obtain directly the PI settings. Alternatively, we may use a two-step procedure, where from K_{c0}, D, B, and t_p we first obtain estimates for the parameters in a first-order plus delay model (see the Appendix for details). We compute the parameters

$$A = 1.152D^2 - 1.607D + 1,$$
$$r = 2A/B$$

and we obtain the following first-order plus delay model parameters from the closed-loop setpoint response (Fig. 5.4):

$$k = 1/(K_{c0}B), \tag{5.7}$$
$$\theta = t_p \cdot \left(0.309 + 0.209e^{-0.61r}\right), \tag{5.8}$$
$$\tau_1 = r\theta. \tag{5.9}$$

These values may subsequently be used with any tuning method, for example, the SIMC PI rules. The closed-loop method may also be used for an unstable process, provided that it can be approximated reasonably well by a stable first-order process. The extension to unstable processes is the reason for taking the absolute value when obtaining the steady-state offset B.

Example E2 ([11]) For the process

$$g_0(s) = \frac{(-0.3s+1)(0.08s+1)}{(2s+1)(1s+1)(0.4s+1)(0.2s+1)(0.05s+1)^3}$$

the closed-loop setpoint response with P-only controller with gain $K_{c0} = 1.5$ is shown in Fig. 5.4. The following data is obtained from the closed-loop response

$$K_{c0} = 1.5, \qquad \Delta y_s = 1, \qquad \Delta y_p = 0.79, \qquad t_p = 4.4, \qquad \Delta y_u = 0.54$$

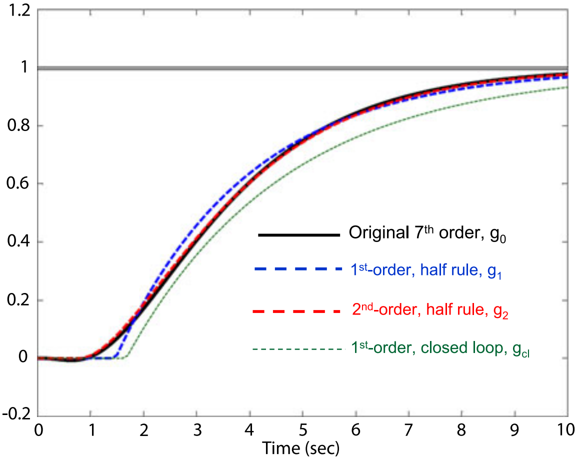


Fig. 5.5 Open-loop response to step change in input u for process E2, $g_0(s) = \frac{(-0.3s+1)(0.08s+1)}{(2s+1)(1s+1)(0.4s+1)(0.2s+1)(0.05s+1)^3}$ (*solid line*), and comparison with various approximations

and we compute

$$\Delta y_\infty = 0.5985, \qquad D = 0.32, \qquad B = 0.67, \qquad A = 0.6038, \qquad r = 1.80$$

which using (5.7)–(5.9) gives the following first-order with delay model approximation:

$$g_{cl}: \quad k = 0.994, \qquad \theta = 1.67, \qquad \tau_1 = 3.00. \tag{5.10}$$

This gives a good approximation of the open-loop step response, as can seen by comparing the curves for g_0 and g_{cl} in Fig. 5.5. The approximation is certainly not the best possible, but it should be noted that the objective is to use the model for tuning, and the resulting difference in the tuning, and thus closed-loop response, may be smaller than it appears by comparing the open-loop responses.

5.2.3 Approximation of Detailed Model Using Half Rule

Assume that we have a given detailed transfer function model in the form

$$g_0(s) = \frac{\prod_j(-T_{j0}^{\text{inv}} s + 1)}{\prod_i(\tau_{i0} s + 1)} e^{-\theta_0 s} \tag{5.11}$$

where all the given parameters are positive, and the time constants are ordered according to their magnitudes. To approximate this with a first- or second-order time delay model, (5.3) or (5.4), Skogestad [11] recommends that the "effective delay" θ

is taken as the "true" delay θ_0, plus the inverse response (negative numerator) time constant(s) T^{inv}, plus half of the largest neglected time constant (half rule), plus all smaller time constant τ_{i0}. The "other half" of the largest neglected time constant is added to get at larger time constant τ_1 (or τ_2 for a second-order model).

Half rule The largest neglected (denominator) time constant (lag) is distributed evenly to the effective delay (θ) and the smallest retained time constant (τ_1 or τ_2).

In summary, for a model in the form (5.11), to obtain a first-order model (5.3), we use

$$\tau_1 = \tau_{10} + \frac{\tau_{20}}{2}; \qquad \theta = \theta_0 + \frac{\tau_{20}}{2} + \sum_{i\geq 3} \tau_{i0} + \sum_j T_{j0}^{\mathrm{inv}} + \frac{h}{2} \tag{5.12}$$

and, to obtain a second-order model (5.4), we use

$$\tau_1 = \tau_{10}; \qquad \tau_2 = \tau_{20} + \frac{\tau_{30}}{2}; \qquad \theta = \theta_0 + \frac{\tau_{30}}{2} + \sum_{i\geq 4} \tau_{i0} + \sum_j T_{j0}^{\mathrm{inv}} + \frac{h}{2} \tag{5.13}$$

where h is the sampling period (for cases with digital implementation).

Example E1 Using the half rule, the process

$$g_0(s) = \frac{1}{(s+1)(0.2s+1)}$$

is approximated as a first-order time delay process, $g(s) = ke^{-\theta s+1}/(\tau_1 s + 1)$, with $k = 1,\ \theta = 0.2/2 = 0.1$, and $\tau_1 = 1 + 0.2/2 = 1.1$.

Example E2 (Continued) Using the half rule, the process

$$g_0(s) = \frac{(-0.3s+1)(0.08s+1)}{(2s+1)(1s+1)(0.4s+1)(0.2s+1)(0.05s+1)^3}$$

is approximated as a first-order time delay process (5.3) with (g_1)

$$\tau_1 = 2 + 1/2 = 2.5,$$

$$\theta = 1/2 + 0.4 + 0.2 + 3 \cdot 0.05 + 0.3 - 0.08 = 1.47$$

or a second-order time delay process (5.4) with (g_2)

$$\tau_1 = 2,$$

$$\tau_2 = 1 + 0.4/2 = 1.2,$$

$$\theta = 0.4/2 + 0.2 + 3 \cdot 0.05 + 0.3 - 0.08 = 0.77.$$

The small positive numerator time constant $T_0 = 0.08$ was subtracted from the effective time delay according to rule T3 (see below). Both approximations, and in particular the second-order model, are very good as can be seen by from the open-loop step responses in Fig. 5.5. Note that with the SIMC tuning rules, a first-order model yields a PI-controller, whereas a second-order model yields a PID controller.

Comment: In this case, we have $\tau_2 > \theta$ $(1.2 > 0.77)$ for the second-order model, and the use of PID control is expected to yield a significant performance improvement compared to PI control (see below for details). However, adding derivative action has disadvantages, such as increased input usage and increased noise sensitivity.

5.2.4 *Approximation of Positive Numerator Time Constants*

A process model can also contain positive numerator time constants T_0 as the following process:

$$g(s) = g_0(s)\frac{T_0 s + 1}{\tau_0 s + 1}. \tag{5.14}$$

Skogestad [11] proposes to cancel out the numerator time constant T_0 against a "neighboring" lag time constant τ_0 by the following rules:[1]

$$\frac{T_0 s + 1}{\tau_0 s + 1} \approx \begin{cases} T_0/\tau_0 & \text{for } T_0 \geq \tau_0 \geq \tau_c & \text{(Rule T1)},\\ T_0/\tau_c & \text{for } T_0 \geq \tau_c \geq \tau_0 & \text{(Rule T1a)},\\ 1 & \text{for } \tau_c \geq T_0 \geq \tau_0 & \text{(Rule T1b)},\\ T_0/\tau_0 & \text{for } \tau_0 \geq T_0 \geq 5\tau_c & \text{(Rule T2)},\\ \frac{(\tilde{\tau}_0/\tau_0)}{(\tilde{\tau}_0 - T_0)s+1} & \text{for } \tilde{\tau}_0 \overset{\text{def}}{=} \min(\tau_0, 5\tau_c) \geq T_0 & \text{(Rule T3)}. \end{cases} \tag{5.15}$$

Here τ_c is the desired closed-loop time constant, which appears as the tuning parameter in the SIMC PID rules. Because the tuning parameter is normally chosen after obtaining the effective time delay (the recommended value for "tight control" is $\tau_c = \theta$), one may not know this value before the model is approximated. Therefore, one may initially have to guess the value τ_c and iterate.

We normally select τ_0 as the closest *larger* denominator time constant $(\tau_0 > T_0)$ and use Rules T2 or T3. Note that an integrating process corresponds to a process with an infinitely large time constant, $\tau_0 = \infty$. For example, for an integrating-pole-zero (IPZ) process of the form $k'\frac{e^{-\theta s}}{s}\frac{Ts+1}{\tau_2 s+1}$, we get $\frac{Ts+1}{s} \approx T$ (Rule T2 with $\tau_0 = \infty > T$). However, if T is smaller than τ_2, then we may use the approximation $\frac{Ts+1}{\tau_2 s+1} \approx \frac{T}{\tau_2}$ (Rule T2 with $\tau_2 > T > 5\theta$). Rule T3 would apply if T was even smaller.

[1] The rules are slightly generalized compared to [11] by replacing θ (effective time delay in final model) by τ_c (desired closed-loop time constant). This makes the rules applicable also to cases where τ_c is selected to be different from θ.

However, if there exists no larger τ_0, or if there is smaller denominator time constant "close to" T_0, then we select τ_0 as the closest *smaller* denominator time constant ($\tau_0 < T_0$) and use rules T1, T1a, or T1b. To define "close to" more precisely, let τ_{0a} (large) and τ_{0b} (small) denote the two neighboring denominator constants to T_0. Then, we select $\tau_0 = \tau_{0b}$ (small) if $T_0/\tau_{0b} < \tau_{0a}/T_0$ and $T_0/\tau_{0b} < 1.6$ (both conditions must be satisfied).

Derivations of the above rules and additional examples are given in [11].

5.3 SIMC PI and PID Tuning Rules (Step 2)

In step 2, we use the model parameters ($k, \theta, \tau_1, \tau_2$) to tune the PID controller. We here derive the SIMC rules and apply them to some typical processes.

5.3.1 Derivation of SIMC Rules

The SIMC rules may be derived using the method of direct synthesis for setpoints [13] or equivalently the Internal Model Control approach for setpoints [8]. For the system in Fig. 5.1, the closed-loop setpoint response is

$$\frac{y}{y_s} = \frac{g(s)c(s)}{g(s)c(s)+1} \tag{5.16}$$

where we have assumed that the measurement of the output y is perfect. The idea of direct synthesis is to specify the desired closed-loop response and solve for the corresponding controller. From (5.16) we get

$$c(s) = \frac{1}{g(s)} \frac{1}{\frac{1}{(y/y_s)_{\text{desired}}} - 1}. \tag{5.17}$$

We here consider the second-order time delay model $g(s)$ in (5.4) and specify that we, following the delay, desire a "smooth" first-order response with time constant τ_c,

$$\left(\frac{y}{y_s}\right)_{\text{desired}} = \frac{1}{\tau_c s + 1} e^{-\theta s}. \tag{5.18}$$

The delay θ is kept in the "desired" response because it is unavoidable. Substituting (5.18) and (5.4) into (5.17) gives a "Smith Predictor" controller [14]:

$$c(s) = \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{k} \frac{1}{(\tau_c s + 1 - e^{-\theta s})} \tag{5.19}$$

τ_c is the desired closed-loop time constant and is the sole tuning parameter for the controller. To derive PID settings, we introduce in (5.19) a first-order Taylor series approximation of the delay, $e^{-\theta s} \approx 1 - \theta s$. This gives

$$c(s) = \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{k} \frac{1}{(\tau_c + \theta)s} \tag{5.20}$$

which is a series form PID-controller (5.1) with [8, 13]

$$K_c = \frac{1}{k} \frac{\tau_1}{\tau_c + \theta} = \frac{1}{k'} \frac{1}{\tau_c + \theta}; \qquad \tau_I = \tau_1; \qquad \tau_D = \tau_2. \tag{5.21}$$

These settings are derived by considering the setpoint response. However, it is well known that for lag dominant processes with $\tau_1 \gg \theta$ (e.g., integrating processes), the choice $\tau_I = \tau_1$ results in a long settling time for *input* ("*load*") *disturbances* [3]. To improve the load disturbance response, one may reduce the integral time, but not by too much, because otherwise we get slow oscillations and robustness problems. Skogestad [11] suggests that a good trade-off between disturbance response and robustness is obtained by selecting the integral time such that we just avoid the slow oscillations, which with the controller gain given in (5.21) corresponds to

$$\tau_I = 4(\tau_c + \theta). \tag{5.22}$$

5.3.2 *Summary of SIMC Rules (Original)*

For a **first-order model**

$$g_1(s) = \frac{k}{(\tau_1 s + 1)} e^{-\theta s} \tag{5.23}$$

the SIMC method results in a PI controller with settings

$$K_c = \frac{1}{k} \frac{\tau_1}{\tau_c + \theta} = \frac{1}{k'} \frac{1}{\tau_c + \theta}, \tag{5.24}$$

$$\tau_I = \min\{\tau_1, 4(\tau_c + \theta)\}. \tag{5.25}$$

The desired first-order *closed-loop* time constant τ_c is the only tuning parameter.

For a **second-order model**

$$g_2(s) = \frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)} e^{-\theta s} \tag{5.26}$$

the SIMC method results in a PID controller with settings (series form)

$$K_c = \frac{1}{k}\frac{\tau_1}{\tau_c + \theta} = \frac{1}{k'}\frac{1}{\tau_c + \theta}, \tag{5.27}$$

$$\tau_I = \min\{\tau_1, 4(\tau_c + \theta)\}, \tag{5.28}$$

$$\tau_D = \tau_2. \tag{5.29}$$

Again, the desired first-order *closed-loop* time constant τ_c is the only tuning parameter. These PID settings are for the cascade (series) form in (5.1). The corresponding settings for the ideal (parallel form) PID controller are easily obtained using (5.30).

PID-control (with derivative action) is primarily recommended for processes with dominant second order-dynamics, defined as having $\tau_2 > \theta$, approximately. We note that the derivative time is then selected so as to cancel the second-largest process time constant.

In Table 5.1 we summarize the resulting tunings for a few special cases, including the pure time delay process, integrating process, and double integrating process. The double integrating process corresponds to a second-order process with $\tau_2 = \infty$, and direct application of the rules actually yield a PD controller, so in Table 5.1 integral action has been added to eliminate the offset for input disturbances.

Table 5.1 SIMC PID-settings (5.27)–(5.29) for some special cases of (5.4) (with τ_c as a tuning parameter)

Process	$g(s)$	K_c	τ_I	$\tau_D^{(5)}$
First-order, (5.3)	$k\frac{e^{-\theta s}}{(\tau_1 s+1)}$	$\frac{1}{k}\frac{\tau_1}{\tau_c+\theta}$	$\min\{\tau_1, 4(\tau_c+\theta)\}$	–
Second-order, (5.4)	$k\frac{e^{-\theta s}}{(\tau_1 s+1)(\tau_2 s+1)}$	$\frac{1}{k}\frac{\tau_1}{\tau_c+\theta}$	$\min\{\tau_1, 4(\tau_c+\theta)\}$	τ_2
Pure time delay$^{(1)}$	$ke^{-\theta s}$	0	$0^{(*)}$	–
Integrating$^{(2)}$	$k'\frac{e^{-\theta s}}{s}$	$\frac{1}{k'}\cdot\frac{1}{(\tau_c+\theta)}$	$4(\tau_c+\theta)$	–
Integrating with lag	$k'\frac{e^{-\theta s}}{s(\tau_2 s+1)}$	$\frac{1}{k'}\cdot\frac{1}{(\tau_c+\theta)}$	$4(\tau_c+\theta)$	τ_2
Double integrating$^{(3)}$	$k''\frac{e^{-\theta s}}{s^2}$	$\frac{1}{k''}\cdot\frac{1}{4(\tau_c+\theta)^2}$	$4(\tau_c+\theta)$	$4(\tau_c+\theta)$
IPZ process$^{(4)}$	$k'\frac{e^{-\theta s}}{s}\frac{Ts+1}{\tau_2 s+1}$	$\frac{1}{k'T}\cdot\frac{\tau_2}{\tau_c+\theta}$	$\min\{\tau_2, 4(\tau_c+\theta)\}$	–

(1) The pure time delay process is a special case of a first-order process with $\tau_1 = 0$

(2) The integrating process is a special case of a first-order process with $\tau_1 \to \infty$

(3) For the double integrating process, integral action has been added according to (5.22)

(4) For the integrating-pole-zero (IPZ) process, we assume that $T > \tau_2$. Then $(Ts+1)/s \approx T$ (rule T2) and the PI-settings follow

(5) The derivative time is for the series form PID controller in (5.1)

$(*)$ Pure integral controller $c(s) = \frac{K_I}{s}$ with $K_I = \frac{K_c}{\tau_I} = \frac{1}{k(\tau_c+\theta)}$

The choice of the tuning parameter τ_c is discussed in more detail below. If the objective is to have "tight control" (good output performance) subject to having good robustness, then the recommendation is to choose τ_c equal to the effective time delay, $\tau_c = \theta$. The same recommendation for τ_c applies to both PI- and PID-controls, but the actual controller settings will differ, because the effective delay θ in a first-order model (PI control) will be larger than that in a second-order model (PID control) of a given process.

Example E2 (Further continued) We want to derive PI- and PID-settings for the process

$$g_0(s) = \frac{(-0.3s+1)(0.08s+1)}{(2s+1)(1s+1)(0.4s+1)(0.2s+1)(0.05s+1)^3}$$

using the SIMC tuning rules with the "default" recommendation $\tau_c = \theta$. From the closed-loop setpoint response, we obtained in a previous example a first-order model with parameters $k = 0.994, \theta = 1.67, \tau_1 = 3.00$ (5.10). The resulting SIMC PI-settings with $\tau_c = \theta = 1.67$ are

$$\text{PI}_{cl}: \quad K_c = 0.904, \qquad \tau_I = 3.$$

From the full-order model $g_0(s)$ and the half rule, we obtained in a previous example a first-order model with parameters $k = 1, \theta = 1.47, \tau_1 = 2.5$. The resulting SIMC PI-settings with $\tau_c = \theta = 1.47$ are

$$\text{PI}_{\text{half-rule}}: \quad K_c = 0.850, \qquad \tau_I = 2.5.$$

From the full-order model $g_0(s)$ and the half rule, we obtained a second-order model with parameters $k = 1, \theta = 0.77, \tau_1 = 2, \tau_2 = 1.2$. The resulting SIMC PID-settings with $\tau_c = \theta = 0.77$ are

$$\text{Series PID}: \quad K_c = 1.299, \qquad \tau_I = 2, \qquad \tau_D = 1.2.$$

The corresponding settings with the more common ideal (parallel form) PID controller are obtained by computing $f = 1 + \tau_D/\tau_I = 1.60$, and we have

$$\text{Ideal PID}: \quad K_c' = K_c f = 1.69, \qquad \tau_I' = \tau_I f = 3.2, \qquad \tau_D' = \tau_D/f = 0.75. \tag{5.30}$$

The closed-loop responses for the three controllers to a setpoint change at $t = 0$ and an input (load) disturbance at $t = 10$ is shown in Fig. 5.6. The responses for the two PI controllers are very similar, as expected. The PID controller shows better output performance (upper plot), especially for the disturbance, but it may not be sufficient to outweigh the increased input usage (lower plot) and increased sensitivity to noise (not shown in plot).

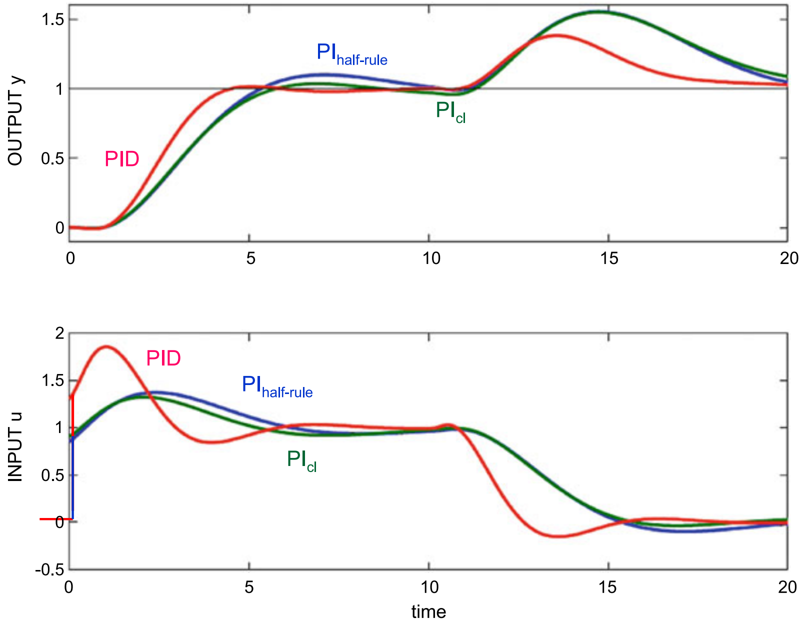


Fig. 5.6 Closed-loop responses for process E2 using SIMC PI- and PID-tunings with $\tau_c = \theta$. Setpoint change at $t = 0$ and input (load) disturbance at $t = 10$. For the PID controller, D-action is only on the feedback signal, i.e., not on the setpoint y_s

5.4 Choice of Tuning Parameter τ_c

The value of the desired closed-loop time constant τ_c can be chosen freely, but from (5.27) we must have $-\theta < \tau_c < \infty$ to get a positive and nonzero controller gain. The optimal value of τ_c is determined by a trade-off between:

1. **Output performance (tight control):** Fast speed of response and good disturbance rejection (favored by a small value of τ_c). This "tightness" can be quantified by the magnitude of the setpoint error, $|y(t) - y_s(t)|$, which should be as small as possible. Here, one may consider different "norms" of the error, for example, the maximum deviation (∞-norm), the integrated square deviation (2-norm) and the integrated absolute error (IAE) (1-norm),

$$\text{IAE} = \int_0^\infty \left|y(t) - y_s(t)\right| dt.$$

2. **Robustness (smooth control):** Good robustness, small input changes, and small noise sensitivity (favored by a large value of τ_c). The "smoothness" is here quantified by the peak value $M_s \geq 1$ of the frequency-dependent sensitivity function, $S = 1/(1 + gc)$. In terms of robustness, $1/M_s$ is the closest distance of the loop transfer function gc to the critical (-1)-point in the Nyquist diagram, so M_s

should be as small as possible. Notice that $M_s < 1.7$ guarantees gain margin (GM) > 2.43 and phase margin (PM) $> 34.2°$ [8].

In general, we have a multiobjective optimization problem, so there is no value of τ_c which is "optimal." We will consider in more detail the two limiting cases of "tight" and "smooth" control and also consider in some detail the required input usage.

5.4.1 Tight Control

With tight control, the primary objective is to keep the output close to its setpoint, but there should be some minimum requirement in terms of robustness and smoothness. A good trade-off is obtained by choosing τ_c equal to the time delay:

Tuning parameter τ_c. SIMC-recommendation for "tight control," or more precisely "tightest possible subject to maintaining smooth control":

$$\tau_c = \theta. \tag{5.31}$$

The choice $\tau_c = \theta$ gives a reasonably fast response with moderate input usage and a good robustness with M_s about 1.6 to 1.7. More specifically, the robustness margins with the SIMC PID-settings in (5.27)–(5.29) and $\tau_c = \theta$, when applied to first- or second-order time delay processes, are always between the values given by the two columns in Table 5.2. The values in the left column in Table 5.2 apply to a case with a relatively small lag time constant (so $\tau_I = \tau_1$), and the somewhat less robust values in the right column apply to an integrating process (so $\tau_I = 4(\tau_c + \theta) = 8\theta$). For the integrating process, we reduce the integral time relative to the original value of $\tau_I = \tau_1$ to get better output performance for load disturbances, and not surprisingly we have to "pay" for this in terms of less robustness.

To be more specific, for processes with a relatively small time constant where we use $\tau_I = \tau_1$ (left column), the system always has a gain margin GM $= 3.14$ and phase margin PM $= 61.4°$, which is much better than the typical minimum requirements GM > 1.7 and PM $> 30°$ [9]. The sensitivity and complementary sensitivity peaks are $M_s = 1.59$ and $M_t = 1.00$ (here small values are desired with a typical upper bound of 2). The maximum allowed time delay error is $\Delta\theta/\theta = \text{PM [rad]}/(w_c \cdot \theta)$, which in this case gives $\Delta\theta/\theta = 2.14$ (i.e., the system goes unstable if the time delay is increased from θ to $(1 + 2.14)\theta = 3.14\theta$).

For an integrating processes (right column) and $\tau_I = 8\theta$, the suggested "tight" settings give GM $= 2.96$, PM $= 46.9°$, $M_s = 1.70$, and $M_t = 1.30$, and the maximum allowed time delay error is $\Delta\theta = 1.59\theta$.

The simulated time responses to setpoint changes and disturbances with SIMC-settings are shown for five cases in Fig. 5.7 [11]. Even though these are for the

Table 5.2 "Tight" settings: Robustness margins for first-order and integrating time delay process for SIMC-rules (5.24)–(5.25) with $\tau_c = \theta$. The same margins apply to a second-order process (5.4) if we choose $\tau_D = \tau_2$ in (5.29)

Process $g(s)$	$\frac{k}{\tau_1 s+1}e^{-\theta s}$	$\frac{k'}{s}e^{-\theta s}$
Controller gain, K_c $(\tau_c = \theta)$	$\frac{0.5}{k}\frac{\tau_1}{\theta}$	$\frac{0.5}{k'}\frac{1}{\theta}$
Integral time, τ_I	τ_1	8θ
Gain margin (GM)	3.14	2.96
Phase margin (PM)	61.4°	46.9°
Allowed time delay error, $\Delta\theta/\theta$	2.14	1.59
Sensitivity peak, M_s	1.59	1.70
Complementary sensitivity peak, M_t	1.00	1.30
Phase crossover frequency, $\omega_{180} \cdot \theta$	1.57	1.49
Gain crossover frequency, $\omega_c \cdot \theta$	0.50	0.51

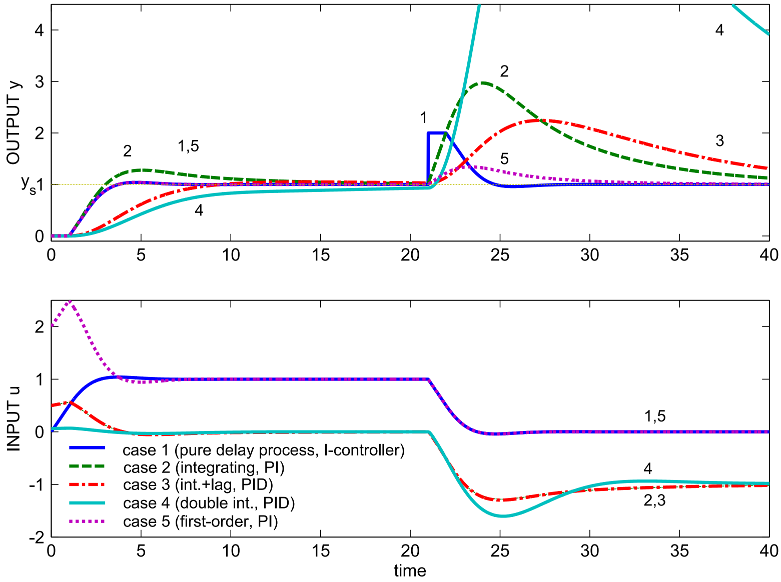


Fig. 5.7 Responses using "tight" SIMC settings $(\tau_c = \theta)$ for five time delay processes. Unit setpoint change at $t = 0$; Unit load disturbance at $t = 20$. Simulations are without derivative action on the setpoint. Parameter values: $\theta = 1, k = 1, k' = 1, k'' = 1$

"tight" settings $(\tau_c = \theta)$, the responses are all smooth. This means that it is certainly possible to get even tighter responses by choosing a smaller value, for example, $\tau_c = 0.5\theta$, but for most process control applications, this is not recommended because of less robustness, larger input usage, and more sensitivity to noise. It may seem from Fig. 5.7 that the SIMC PID-controller does not work well for the double integrating process (curve 4), but this is a difficult process to control and the response to a unit input disturbance will be large for any robust controller.

5.4.2 Smooth Control

Even though the recommended "tight" settings ($\tau_c = \theta$) give responses that are reasonably smooth, they may still be unnecessary aggressive compared to the required performance objectives, especially if the effective delay θ is small. For example, for the limiting case with $\theta = 0$ (no delay), we get with $\tau_c = \theta$ an infinite controller gain, which is clearly not realistic. Thus, in practice one often uses a "smoother" tuning, that is, $\tau_c > \theta$.

However, τ_c should not be too large, because otherwise the output y will go out of bound when there are disturbances d. The question is: *How slow (smooth) can we tune the controller and still get acceptable control?* This issue is addressed in the paper by Skogestad [12] on "tuning for smooth PID control with acceptable disturbance rejection," where the following lower bound on the controller gain is derived (for both PI- and PID-controls).

Controller gain SIMC-recommendation for "smooth control," or more precisely "smoothest possible subject to acceptable disturbance rejection":

$$|K_c| > |K_{c,\min}| = \frac{|\Delta u_0|}{|\Delta y_{\max}|}, \tag{5.32}$$

where
$\Delta y_{\max}$ = maximum allowed deviation in the output y
Δu_0 = required input change to reject the disturbance(s) d.

Substituting $K_{c,\min}$ into (5.24) or (5.27), one can obtain the corresponding value $\tau_{c,\max}$, and we end up with a region of recommended values for the tuning parameter τ_c:

$$\tau_{c,\min} \text{ ("tight")} < \tau_c < \tau_{c,\max} \text{ ("smooth")} \tag{5.33}$$

where

$$\tau_{c,\min} = \theta, \qquad \tau_{c,\max} = \frac{1}{K_{c,\min}} \cdot \frac{\tau_1}{k} - \theta. \tag{5.34}$$

The final choice of τ_c is an engineering decision. A small value for τ_c ("tight control" of y) is typically desired for control of active constraints, because tight control reduces the required backoff (safety margin to the constraint). On the other hand, tight control will require larger input changes which may disturb the rest of the process. For example, for liquid level, there is usually no reason to control the level tightly, so a large value of τ_c ("smooth control") is desired.

Details on the derivation of (5.32) and $\tau_{c,\max}$ are given in [12], but let us here give a simplified version. Consider disturbance rejection and assume that we use a

P-only controller with gain K_c. The input change (in deviation from the nominal value) is then $\Delta u = -K_c \Delta y$ or

$$|\Delta u| = |K_c| \cdot |\Delta y|.$$

Assume that the required input change to reject a disturbance is Δu_0. For example, if we have a disturbance Δd_I at the input, then $\Delta u_0 = -\Delta d_I$. The smallest controller gain that can generate the required input change Δu_0 is obtained when we have the largest output change ($|\Delta y| = |\Delta y_{\max}|$), and we get

$$|\Delta u_0| = |K_{c,\min}| \cdot |\Delta y_{\max}|$$

and (5.32) follows.

5.4.3 Input Usage

The magnitude of the dynamic input change can be an important issue when tuning the controller, that is, when selecting the value for τ_c. The transfer function from the disturbance d to the input u is given by (see Fig. 5.1):

$$u(s) = -\frac{g_d c}{1 + gc} d(s)$$

With integral action in the controller (e.g., PI or PID control), the steady-state input change to a step disturbance d is independent of the controller and is given by $u(t = \infty) = -\frac{k_d}{k} d$ where k_d is the steady-state disturbance gain and k is the steady-state process gain. We assume that we can reject the expected disturbances at steady state, that is, we assume $|u(t = \infty)| = |\frac{k_d}{k} d| \leq |u_{\max}|$ where $|d|$ is the magnitude of the disturbance change, and $|u_{\max}|$ is the maximum allowed input change, because otherwise the process is not "controllable" (with any controller). However, the dynamic input change $u(t)$ will depend on the controller tuning, and we will consider the initial change (at $t = 0^+$) just after a step disturbance d.

We consider two important disturbances, namely an input "load" disturbance d_u (corresponding to $g_d = g$) and an output disturbance d_y (corresponding to $g_d = 1$). Note that an output disturbance has an immediate effect on the output y. A physical example is a process where we add another stream (output disturbance) just before the measurement y. Mathematically, an output disturbance is equivalent to a setpoint change (with $y_s = -d_y$)

For an *input ("load") disturbances* d_u, input usage is not an important issue for SIMC-tuning, even dynamically. This is because the SIMC controller gives a closed-loop transfer function $\frac{y}{y_s} = \frac{gc}{1+gc}$ with little or no overshoot, see (5.16) and (5.18), and since $\frac{u}{d_u} = -\frac{gc}{1+gc}$, we get for d_u a corresponding input response with little overshoot. This is illustrated by the input changes for a load disturbance ($t = 20$) in Fig. 5.7.

On the other hand, for an *output disturbances* d_y ($g_d = 1$) or equivalently for a *setpoint change* $y_s = -d_y$, input usage may be an important issue for tuning. The steady-state input change to a step setpoint change y_s is $u(t = \infty) = \frac{1}{k} y_s$. However, with PI-control the input will initially jump to the value $u(t = 0^+) = K_c y_s$, as illustrated for the setpoint change in Fig. 5.7 (e.g., see the first-order process, case 5). This initial change is larger than the steady-state change if $K_c k > 1$, which is usually the case, except for delay-dominant processes. With SIMC-tunings we must require

$$\left|u\left(t = 0^+\right)\right| = |K_c y_s| = \left|\frac{\tau_1}{\tau_c + \theta} \frac{1}{k} y_s\right| \leq |u_{\max}|. \tag{5.35}$$

Note that u and y_s are deviation variables. Consider, for example, a first-order process with $\tau_1 = 8$ and $\theta = 1$. With the choice $\tau_c = \theta$, the initial input change is $\tau_1/(\tau_c + \theta) = 4$ times the steady-state input change y_s/k. If such a large dynamic input change is not feasible, then one would need to use "smoother" control with a larger value for τ_c in order to satisfy (5.35).[2]

With PID control, the derivative action will cause even larger input changes for output disturbances, and this may be one reason for reducing or even avoiding derivative action. It is also the reason why to avoid "derivative kick," we recommend that the setpoint is not differentiated, see (5.2).

5.5 Optimality of SIMC PI Rules

How good are the SIMC PI rules, that is, how much room is there for improvements? To study this, we compare the SIMC PI performance, with τ_c as a parameter, to the "Pareto-optimal" PI-controller. Pareto-optimality applies to multiobjective problems and means that no further improvement can be made in objective 1 (output performance in our case) without sacrificing objective 2 (robustness and input usage in our case).

We choose to quantify robustness and input usage in terms of the sensitivity peak M_s. We also considered other "robustness" measures, for example, the relative delay margin as suggested by Foley et al. [4], but we choose to use M_s. One reason is that we found that the M_s-value correlates well with the input usage as given by its total variation (TV), which agrees with the findings of Foley et al. [4]. Such a correlation is reasonable since a large M_s-value corresponds to an oscillatory system with large input variations.

We choose to quantify performance in terms of the integrated absolute error in response to a setpoint change (IAE_{ys}) and to an input "load" disturbance (IAE_d).

[2] It may seem from (5.35) that "slow" processes, which have a large time constant τ_1, will always require "slow" control (large τ_c) in order to avoid excessive input changes. However, this is usually not the case because such processes often have a corresponding large gain k such that the value $k' = k/\tau_1$ may be sufficiently large to satisfy (5.35) even with $\tau_c = \theta$.

Table 5.3 Optimal PI-controllers ($M_s = 1.59$) and corresponding IAE-values for four processes

Process	Setpoint			Input disturbance			Optimal combined (minimize J)					
	K_c	τ_I	IAE^o_{ys}	K_c	τ_I	IAE^o_d	K_c	τ_I	IAE_{ys}	IAE_d	J	M_s
e^{-s}	0.20	0.32	1.607	0.20	0.32	1.607	0.20	0.32	1.607	1.607	1	1.59
$\frac{e^{-s}}{s+1}$	0.55	1.15	2.083	0.50	1.04	2.036	0.54	1.10	2.084	2.037	1.00	1.59
$\frac{e^{-s}}{8s+1}$	4.0	8	2.169	3.33	3.65	1.135	3.47	4.0	3.096	1.164	1.23	1.59
$\frac{e^{-s}}{s}$	0.50	∞	2.169	0.40	5.8	15.09	0.41	6.3	4.314	15.4	1.51	1.59

IAE_{ys} is for a unit setpoint change. IAE_d is for a unit input disturbance

Table 5.4 SIMC PI-controllers ($\tau_c = \theta$) and corresponding J- and M_s-values for four processes

Process	SIMC PI ($\tau_c = \theta$)						Improved SIMC PI ($\tau_c = \theta$)					
	K_c	τ_I	IAE_{ys}	IAE_d	J	M_s	K_c	τ_I	IAE_{ys}	IAE_d	J	M_s
e^{-s}	0	$0^{(*)}$	2.17	2.17	1.35	1.59	0.17	0.33	1.95	1.95	1.21	1.45
$\frac{e^{-s}}{s+1}$	0.5	1	2.17	2.04	1.03	1.59	0.67	1.33	1.99	1.99	1.09	1.69
$\frac{e^{-s}}{8s+1}$	4	8	2.17	2.00	1.38	1.59	4.17	8	2.14	1.92	1.34	1.62
$\frac{e^{-s}}{s}$	0.5	8	3.92	16	1.43	1.70	0.5	8	3.92	16	1.43	1.70

$^{(*)}$ Pure integral controller with $K_I = K_c/\tau_I = 0.5$

The setpoint performance is often referred to as the "servo" behavior, and the disturbance (in this case the input "load" disturbance) performance is often referred to as "regulator" behavior. It may be argued that a two-degree-of-freedom controller ("feedforward action") may be used to improve the response for setpoints, but note that a setpoint change is equivalent to an output disturbance (with $g_d = 1$ in Fig. 5.1) which can only be counteracted by feedback. Thus, both setpoint changes (output disturbances) and input disturbances should be included when evaluating performance, and to get a good balance between the two, we weigh them about equally by defining the following performance cost:

$$J(c) = 0.5\left[\frac{\mathrm{IAE}_{ys}(c)}{\mathrm{IAE}^o_{ys}} + \frac{\mathrm{IAE}_d(c)}{\mathrm{IAE}^o_d}\right] \tag{5.36}$$

where the reference values, IAE^o_{ys} and IAE^o_d, are for IAE-optimal PI-controllers (with $M_s = 1.59$) for a setpoint change and input disturbance, respectively. We could have used the truly optimal IAE-value as the reference when computing J (without the restriction $M_s = 1.59$), but this would not have changed the results much because the IAE-value is anyway quite close to its minimum at $M_s = 1.59$. Table 5.3 gives the tunings and reference values obtained using IAE-optimal PI-controllers (with $M_s = 1.59$) for four different processes, and Table 5.4 gives the tunings, costs J, and M_s-values for the SIMC PI-controller (with $\tau_c = \theta$). Importantly, the weighted cost J is independent of the process gain k and the disturbance

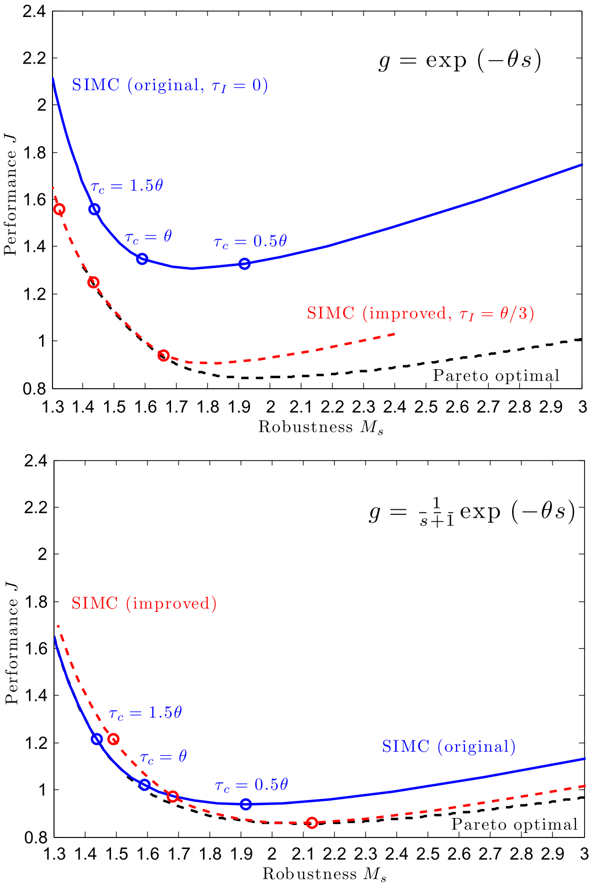
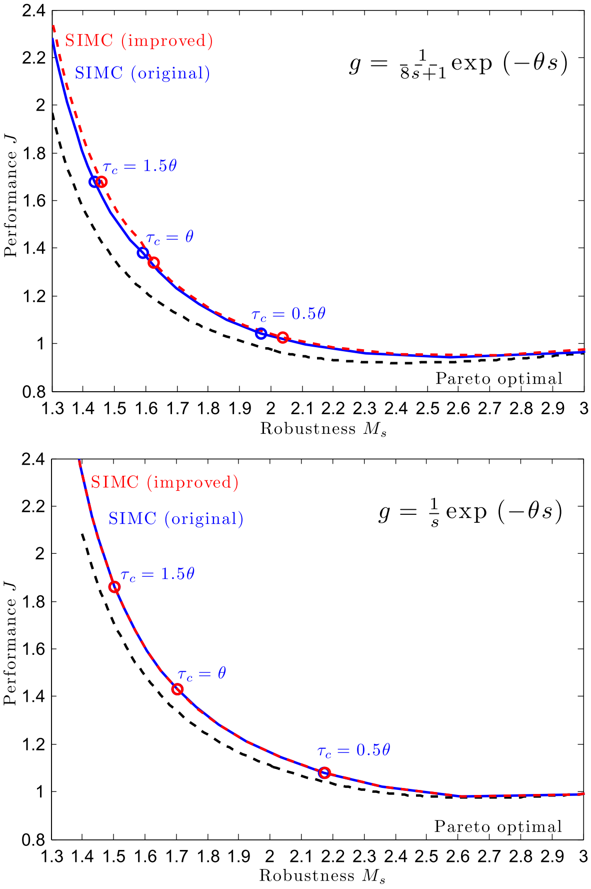


Fig. 5.8 Check of optimality of SIMC PI tuning rules for four processes

magnitude, and also of the unit used for time. Note that two different optimal PI-controllers are used to obtain the two reference values, whereas a single controller c is used to find $\mathrm{IAE}_{ys}(c)$ and $\mathrm{IAE}_d(c)$ when evaluating the weighted IAE-cost $J(c)$.

Figure 5.8 shows the trade-off between performance (J) and robustness (M_s) for the SIMC PI-controller (blue solid curve) and the Pareto-optimal controller (dashed black curve) for four different processes: pure time delay ($\tau_1/\theta = 0$), small time constant ($\tau_1/\theta = 1$), intermediate time constant ($\tau_1/\theta = 8$), and integrating process ($\tau_1/\theta = \infty$). The curve for the SIMC controller was generated by varying the tuning parameter τ_c from a large to a small value. The controllers corresponding to the choices $\tau_c = 1.5\theta$ (smoother), $\tau_c = \theta$ (recommended), and $\tau_c = 0.5\theta$ (aggressive) are shown by circles. The Pareto-optimal curve was generated by finding for each value of M_s, the optimal PI-controller c with the smallest IAE-value $J(c)$. Except for the pure time delay process, the differences between the J-values for SIMC (blue solid curve) and optimal (dashed black curve) are small (within about 10%), which shows that the SIMC PI-rules are close to optimal.

Fig. 5.8 (Continued)



Note that we have a real trade-off between performance (J) and robustness (M_s) only when there is a negative slope between these variables (in the left region in the figures in Fig. 5.8). We never want to be in the region with a zero or positive slope (to the right in the figures), because here we can improve both performance (J) and robustness (M_s) at the same time with another choice for the tuning parameter (using a larger value for τ_c). Another important observation from Fig. 5.8 is then that the SIMC-recommendation $\tau_c = \theta$ for "tight" control (as given by middle of the three circles) in all cases is located in the desired trade-off region with a negative slope, well before we reach the minimum. Also, the recommended choice gives a fairly constant M_s-value in the region 1.59 to 1.7. From this we conclude that, except for the time delay process, there is little room to improve on the SIMC PI rules, at least when performance and robustness are as defined above (J and M_s).

The IAE-cost J in (5.36) is based on equal weighting of servo (output disturbance) and regulator (input disturbance) performance. The existence of a trade-off between servo and regulator performance can be quantified by considering how much larger the (Pareto) optimal cost J_{opt} (dashed black line) is than 1 at the reference robustness, $M_s = 1.59$, see also Table 5.3. For a pure time delay-process, we have that $J_{\mathrm{opt}} = 1$ for $M_s = 1.59$, and there is no trade-off. The reason is that the setpoint and output disturbance responses are the same. On the other hand, for the other extreme of an integrating process, we have a clear trade-off since the optimal PI-controller has $J_{\mathrm{opt}} = 1.51$ (the SIMC PI-controller with $M_s = 1.59$ is close to this with J about 1.6). The existence of the servo/regulator trade-off for an integrating process implies that for a given robustness (M_s-value), one can find PI-settings with significantly better regulator (load disturbance) performance or better servo (setpoint) performance, but not both at the same time. To be able to shift the trade-off, one may introduce an extra parameter in the PID rules [1], in addition to τ_c. For the SIMC method, this extra servo/regulator trade-off parameter could be c in the following expression for the integral time:

$$\tau_I = \min\bigl(\tau_1, c(\tau_c + \theta)\bigr) \tag{5.37}$$

where $c = 4$ gives the original SIMC-rule. A larger value of c improves the setpoint performance, and a smaller value, e.g., $c = 2$, improves the input disturbance performance [6]. However, introducing an extra parameter adds complexity, and the potential benefit does not seem sufficiently large. Nevertheless, one may consider choosing another (lower) fixed value for c. There are two reasons why we recommend keeping the SIMC-value of $c = 4$. First, it is close to the Pareto-optimal PI controller (as seen from Fig. 5.8), so we cannot get a significant improvement with our performance objective J. Second, with a smaller value for c, say $c = 2.5$, the recommended choice $\tau_c = \theta$ becomes less robust (with a higher M_s), so one would need to recommend a different value for τ_c for an integrating process, say $\tau_c = 1.5\theta$, which would add complexity. In summary, we find that the value $c = 4$ in the original SIMC rule provides a well-balanced servo/regulator trade-off.

5.6 Improved SIMC Tuning Rules

For a pure time delay process, we see from Fig. 5.8 that the IAE-value (J) for the SIMC controller is about 40% higher than the minimum with the same robustness (M_s). This is further illustrated by the closed-loop simulations in Fig. 5.9, where we see that the SIMC PI-controller (denoted SIMC-original in the figure) gives a nice and smooth response. However, the response is somewhat sluggish initially, because it is actually a pure I-controller (with $K_c = 0$, $\tau_I = 0$, and $K_I = K_c/\tau_I = 0.5$). On the other hand, the IAE-optimal PI-controller (with minimum J for $M_s = 1.59$) has K_c about 0.2 and τ_I about 0.32 (and $K_I = 0.62$). In fact, the optimal PI-controller for a pure time delay process (dashed black line in Fig. 5.8) has an almost fixed integral time of approximately $\theta/3$ for all values of M_s between 1.4 and 1.7.

Based on this fact, we propose a simple change to the SIMC-rules, namely to replace τ_1 by $\tau_1 + \theta/3$ in the rules (PI control), which markedly improved the responses for a pure time delay process. It is important that the change is simple

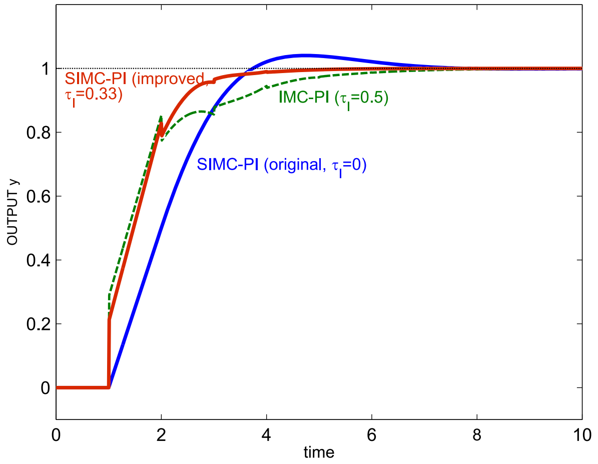


Fig. 5.9 Closed-loop setpoint responses for pure time delay process ($\theta = 1, k = 1, \tau_1 = 0$) with PI-control. All three controllers have the same robustness ($M_s = 1.59$). For a pure time delay process, the setpoint and disturbance responses are identical, and the input and output are identical. IMC PI: $K_c = 0.29$ and $\tau_I = 0.5$ ($K_I = K_c/\tau_I = 0.58$). SIMC PI original ($\tau_c = \theta$): $K_c = 0$ and $\tau_I = 0$ ($K_I = 0.5$). SIMC PI improved ($\tau_c = 0.61\theta$): $K_c = 0.207$ and $\tau_I = 0.333$ ($K_I = 0.62$)

because "simplicity" was one of the main objectives when originally deriving the SIMC rules.

A similar change, but with $\theta/2$ rather than $\theta/3$, was originally proposed by Rivera et al. [8] for their "improved PI" tuning rule, and the effectiveness of this modification is also clear from the paper of Foley et al. [4]. However, as seen in Fig. 5.9, the response with this IMC PI controller also settles rather slowly toward the setpoint, indicating that the integral time $\theta/2$ is too large. The proposed value $\theta/3$ gives a faster settling and is also closer to the original SIMC-rule (which is zero for a time delay process). The conclusion is that we recommend to replace τ_1 by $\tau_1 + \theta/3$ in the SIMC rules to get the improved SIMC rules:

Improved SIMC PI-rule for first-order with delay process

$$K_c = \frac{1}{k}\frac{\tau_1 + \frac{\theta}{3}}{\tau_c + \theta}, \tag{5.38}$$

$$\tau_I = \min\left\{\tau_1 + \frac{\theta}{3}, 4(\tau_c + \theta)\right\}. \tag{5.39}$$

The improvement of this rule for a pure time delay processes is clear from the red curves in Figs. 5.9 and 5.8 (upper left); for small M_s-values, the improved SIMC-controller is almost identical to the Pareto-optimal, which confirms that $\tau_I = \theta/3$

is close to optimal for a pure time delay process. For the process with a small time constant ($\tau_1 = \theta$), the improved SIMC rule (red curve in lower left plot in Fig. 5.8) is slightly better than the "original" SIMC rule (blue curve) for higher M_s-values (where we get better performance) but slightly worse for lower M_s-values. For the two processes with a large time constant ($\tau_1 = 8\theta$ and $\tau_1 = \infty$), there is, as expected, almost no difference between the original and improved SIMC rules.

5.7 Discussion

5.7.1 Measurement Noise

Measurement noise has not been considered in this chapter, but it is an important consideration in many cases, especially if the proportional gain K_c is large, or, for cases with derivative action, if the derivative gain $K_c\tau_D$ is large. However, since the magnitude of the measurement noise varies a lot in applications, it is difficult to give general rules about when measurement noise may be a problem. In general, robust designs (with small M_s) are insensitive to measurement noise. Therefore, the SIMC rules with the recommended choice $\tau_c = \theta$ are less sensitive to measurement noise than most other published settings method, including the Ziegler–Nichols settings. If actual implementation shows that the sensitivity to measurement noise is too large, then the following modifications may be attempted:

1. Filter the measurement signal, for example, by sending it through a first-order filter $1/(\tau_F s + 1)$; see also (5.2). With the proposed SIMC-settings, one can typically increase the filter time constant τ_F up to almost 0.5θ, without a large affect on performance and robustness.
2. If derivative action is used, one may try to remove it, and obtain a first-order model before deriving the SIMC PI-settings.
3. If derivative action has been removed and filtering the measurement signal is not sufficient, then the controller needs to be detuned by selecting a larger value for τ_c.

5.7.2 Retuning for Integrating Processes

Integrating processes

$$g(s) = k' \frac{e^{-\theta s}}{s}$$

are common in industry, but control performance is often poor because of incorrect controller settings. When encountering oscillations, the intuition of the operators is to reduce the controller gain. If the oscillations are relatively slow, then this is the exactly opposite of what one should do for an integrating process. The product of the controller gain K_c and the integral time τ_I must be *larger* than $4/k'$ to avoid slow oscillations [11]. One solution is to simply use proportional control (with $\tau_I = \infty$), but this is often not desirable. Here we show how to easily retune the controller to just avoid the oscillations without actually having to derive a model. This approach has been applied with success to industrial examples.

Consider a PI controller with (initial) settings K_{c0} and τ_{I0} which results in "slow" oscillations with period P_0 (larger than $3 \cdot \tau_{I0}$, approximately). Then we likely have a close-to integrating process for which the product of the controller gain and integral time ($K_{c0}\tau_{I0}$) is too low. To avoid oscillations with the new settings K_c and τ_I, we must require [11]:

$$\frac{K_c \tau_I}{K_{c0}\tau_{I0}} \geq \frac{1}{\pi^2} \cdot \left(\frac{P_0}{\tau_{i0}}\right)^2. \tag{5.40}$$

Here $1/\pi^2 \approx 0.10$, so we have the **rule**:

- To avoid "slow" oscillations, the product of the controller gain and integral time should be increased by a factor $F \approx 0.1(P_0/\tau_{I0})^2$.

5.7.3 Controllability

The effective delay θ is easily obtained using the proposed half rule. Since the effective delay is the main limiting factor in terms of control performance, its value gives invaluable insight about the inherent controllability of the process.

From the settings in (5.27)–(5.29), a PI-controller results from a first-order model, and a PID-controller results from a second-order model. With the effective delay computed using the half rule in (5.12)–(5.13), it then follows that PI-control performance is limited by (half of) the magnitude of the second-largest time constant τ_2, whereas PID-control performance is limited by (half of) the magnitude of the third-largest time constant, τ_3.

5.8 Conclusions and Future Perspectives

This chapter has summarized the SIMC two-step procedure for deriving PID settings for typical process control applications.

Step 1 The real process is approximated by a first-order with delay model (for PI control) or a second-order model (for PID control). To obtain the model, the simplest approach is probably to use an open-loop step experiment (Fig. 5.3), but if this is difficult for some reasons, then one may alternatively use a closed-loop setpoint response with P-controller (Fig. 5.4). If the starting point is a detailed model, then the half rule may be used to obtain the effective delay θ, see (5.12)–(5.13).

Step 2 For a first-order model (with parameters k, τ_1, and θ), the following SIMC PI-settings are suggested (original SIMC rule):

$$K_c = \frac{1}{k}\frac{\tau_1}{\tau_c + \theta}; \qquad \tau_I = \min\{\tau_1, 4(\tau_c + \theta)\}$$

where the closed-loop response time τ_c is the tuning parameter. For a dominant second-order process (for which $\tau_2 > \theta$, approximately), one needs to add derivative action with

$$\text{Series-form PID:} \quad \tau_D = \tau_2.$$

To improve the performance for delay-dominant processes, one may replace τ_1 by $\tau_1 + \frac{\theta}{3}$ and use the "improved" SIMC PI-rules in (5.38)–(5.39). A more careful analysis needs to be done to check if a similar improvement can be used with a PID controller.

Note that although the same formulas are used to obtain K_c and τ_I for both PI- and PID-control, the actual values will differ since the effective delay θ is smaller for a second-order model. The tuning parameter τ_c should be chosen to get the desired trade-off between fast response (small IAE) on the one side, and smooth input usage and robustness (small M_s) on the other side. The recommended choice $\tau_c = \theta$ gives robust (M_s about 1.6 to 1.7) and somewhat conservative settings when compared with most other tuning rules, and if it is desirable to get faster control, one may consider reducing τ_c to about $\theta/2$ (see Fig. 5.8). More commonly, one may want to have "smoother" control with $\tau_c > \theta$ and a smaller controller gain K_c. However, the controller gain must be larger than the value given in (5.32) to achieve a minimum level of disturbance rejection.

Comparing the performance of the SIMC-rules with the optimal for a given robustness (M_s value) shows that the SIMC-rules are close to the Pareto-optimal settings (Fig. 5.8). This means that the room for improving the SIMC PI-rules is limited, at least for the first-order plus delay processes considered in this chapter, and with a good trade-off between rejecting input and output (setpoint) disturbances.

However, it should be noticed that the SIMC rules apply to processes that can be reasonably well approximated by first- or second-order plus delay models. This applies to most process control applications, including some unstable plants, but it obviously does not apply in general, for example, for some of the unstable or oscillating processes found in mechanical systems. For such processes, it would be interesting to study the validity and extension of the SIMC rules or similar analytic model-based PID tuning rules. It is also interesting to establish for which processes the PID controller is a suitable controller and for which processes it is not.

Appendix: Estimation of Parameters τ_1 and θ from Closed-Loop Step Response

Shamsuzzoha and Skogestad [10] discuss at the end of their paper a two-step closed-loop procedure, where the first step is to use closed-loop data and some expressions to obtain the parameters k, τ_1, and θ. We use this approach but have modified the expressions. Our expression for k in (5.7) is given by their equation (35) by noting that $B = |(1-b)/b|$ where $b = \Delta y_\infty / \Delta y_s$. However, our expressions for θ and τ_1 in (5.8)–(5.9) differ somewhat from their equations (36) and (37). The reason is that their equations (36) and (37) are not consistent in terms of the time delay estimate, because the expression for τ_1 in (36) is based on $\theta = 0.43t_p$, whereas (37) uses $\theta = 0.305t_p$. To correct for this, we first note from (19) in their paper (noting that $\tau_1 = \tau_I$ for the delay-dominant case) that τ_1 and θ are related by

$$\tau_1 = r\theta$$

where $r = 2A/B$, which is our expression in (5.9). Here, Shamsuzzoha and Skogestad [10] recommend to use $\theta = 0.44t_p$ for $\tau_1 < 8\theta$ and $\theta = 0.305t_p$ for $\tau_1 > 8\theta$. However, to get better accuracy and a smooth transition, we fitted simulation data for θ/t_p as a function of τ_1/θ for a wide range of processes with an overshoot of 0.3 and obtained the correlation [5]

$$\theta = t_p \cdot \left(0.309 + 0.209e^{-0.61(\tau_1/\theta)}\right)$$

as given in (5.8). Note here that $(0.309+0.209e^{-0.61(\tau_1/\theta)})$ is 0.518 for $r = \tau_1/\theta = 0$ and 0.309 for $r = \infty$.

References

1. Alcantara, S., Pedret, C., Vilanova, R.: On the model matching approach to PID design: analytical perspective for robust servo/regulator tradeoff tuning. J. Process Control **20**, 596–608 (2010)
2. Astrom, K.J., Hagglund, T.: PID Controllers: Theory, Design and Tuning, 2nd edn. ISA—Instrument Society of America (1995)
3. Chien, I.L., Fruehauf, P.S.: Consider IMC tuning to improve controller performance. Chem. Eng. Prog. **86**, 33–41 (1990)
4. Foley, M.W., Ramharack, N.R., Copeland, B.R.: Comparison of PI controller tuning methods. Ind. Eng. Chem. Res. **44**(17), 6741–6750 (2005)
5. Grimholt, C.: (2010). Verification and improvements of the SIMC method for PI control. Technical report. 5th year project. Department of Chemical Engineering. Norwegian University of Science and Technology, Trondheim. http://www.nt.ntnu.no/users/skoge/diplom/prosjekt10/grimholt
6. Haugen, F.: Comparing PI tuning methods in a real benchmark temperature control system. Model. Identif. Control **31**, 79–91 (2010)
7. Horn, I.G., Arulandu, J.R., Gombas, J., VanAntwerp, J.G., Braatz, R.D.: Improved filter design in internal model control. Ind. Eng. Chem. Res. **35**(10), 3437–3441 (1996)
8. Rivera, D.E., Morari, M., Skogestad, S.: Internal model control. 4. PID controller design. Ind. Eng. Chem. Res. **25**(1), 252–265 (1986)

9. Seborg, D.E., Edgar, T.F., Mellichamp, D.A.: Process Dynamics and Control. Wiley, New York (1989)
10. Shamsuzzoha, M., Skogestad, S.: The setpoint overshoot method: a simple and fast method for closed-loop PID tuning. J. Process Control **20**, 1220–1234 (2010)
11. Skogestad, S.: Simple analytic rules for model reduction and PID controller tuning. J. Process Control **13**, 291–309 (2003)
12. Skogestad, S.: Tuning for smooth PID control with acceptable disturbance rejection. Ind. Eng. Chem. Res. **45**, 7817–7822 (2006)
13. Smith, C.A., Corripio, A.B.: Principles and Practice of Automatic Process Control. Wiley, New York (1985)
14. Smith, O.J.: Closer control of loops with dead time. Chem. Eng. Prog. **53**, 217 (1957)
15. Tyreus, B.D., Luyben, W.L.: Tuning PI controllers for integrator/dead time processes. Ind. Eng. Chem. Res. **31**, 2628–2631 (1992)
16. Yuwana, M., Seborg, D.E.: A new method for on-line controller tuning. AIChE J. **28**, 434–440 (1982)
17. Ziegler, J.G., Nichols, N.B.: Optimum settings for automatic controllers. Trans. Am. Soc. Mech. Eng. **64**, 759–768 (1942)

Chapter 6
PID Control for MIMO Processes

Qing-Guo Wang and Zhuo-Yun Nie

6.1 MIMO Feedback Systems

A multivariable system is a system with multiple inputs and multiple outputs. A MIMO system may be described by a transfer function matrix

$$Y(s) = G(s)U(s), \quad G(s) = \begin{bmatrix} g_{11} & \cdots & g_{1m} \\ \vdots & \ddots & \vdots \\ g_{l1} & \cdots & g_{lm} \end{bmatrix},$$

in which $g_{ij}(s)$ is the transfer function relating the ith output to the jth input. If $m = l$, the system is called square, otherwise the system is nonsquare.

Zeros and poles play an important role in SISO feedback system analysis and design. We need to generalize them to the MIMO case. For a SISO system, the poles and zeros of a scalar, strictly proper, coprime rational function

$$g(s) = \frac{n(s)}{d(s)}$$

are given by the roots of the equations $d(s) = 0$ and $n(s) = 0$, respectively. It can be shown that a rational transfer matrix $G(s)$ can be expressed in terms of with its Smith–McMillan form $M(s)$ via

$$G(s) = U_1(s)M(s)U_2(s),$$

Q.-G. Wang (✉)
Department of Electrical and Computer Engineering, National University of Singapore, Singapore, Singapore
e-mail: elewqg@nus.edu.sg

Z.-Y. Nie
School of Information Science and Engineering, Central South University, Changsha, China
e-mail: yezhuyun2004@sina.com

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_6, © Springer-Verlag London Limited 2012

where $U_1(s)$ and $U_2(s)$ are unimodular, and

$$M(s) = \begin{bmatrix} \frac{\varepsilon_1(s)}{\varphi_1(s)} & & & & 0 \\ & \frac{\varepsilon_1(s)}{\varphi_1(s)} & & & \\ & & \ddots & & \\ & & & \frac{\varepsilon_r(s)}{\varphi_r(s)} & \\ 0 & & & & 0 \end{bmatrix}.$$

The monic polynomials $\{\varepsilon_i(s), \varphi_i(s)\}$ are coprime for each i (they have no common factors) and satisfy the divisibility property: $\varepsilon_i | \varepsilon_{i+1}$ and $\varphi_{i+1} | \varphi_i$, $i = 1, 2, \ldots, r-1$. We define pole polynomial and zero polynomial, respectively, as

$$p(s) = \varphi_1(s)\varphi_2(s) \cdots \varphi_r(s),$$
$$z(s) = \varepsilon_1(s)\varepsilon_2(s) \cdots \varepsilon_r(s).$$

The roots of the equations $p(s) = 0$ and $z(s) = 0$ are called the poles and zeros of $G(s)$, respectively [1]. The degree of a pole polynomial $p(s)$ is called the McMillan degree of $G(s)$. A MIMO $G(s)$ is called stable if all its poles have strictly negative real parts, the same as for the SISO case.

If $G(s)$ is square and nonsingular, then

$$\det G(s) = \det(U_1 M U_2) = c \frac{z(s)}{p(s)} \quad \text{for a constant } c,$$

which tells us that zeros of $\det G(s)$ are zeros of $G(s)$ and poles of $\det G(s)$ are poles of $G(s)$. But the converse is NOT true because there might be common factors between $z(s)$ and $p(s)$. A multivariable system can have a pole and a zero at the same place, but they do not cancel each other out within the transfer function matrix [2].

Example 6.1 Let

$$G(s) = \begin{bmatrix} \frac{1}{s^2+3s+2} & \frac{-1}{s^2+3s+2} \\ \frac{s^2+s-4}{s^2+3s+2} & \frac{2s^2-s-8}{s^2+3s+2} \\ \frac{s-2}{s+1} & \frac{2s-4}{s+1} \end{bmatrix}.$$

We can obtain the Smith–McMillan form of $G(s)$ as follows:

$$M(s) = \begin{bmatrix} \frac{1}{(s+1)(s+2)} & 0 \\ 0 & \frac{s-2}{s+1} \\ 0 & 0 \end{bmatrix}.$$

The pole and zero polynomials of $G(s)$ are

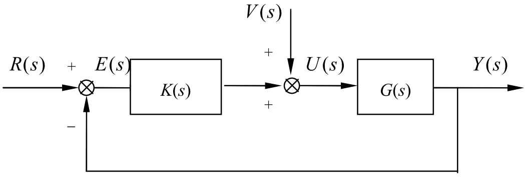


Fig. 6.1 The unity feedback system

$$p(s) = (s+1)^2(s+2),$$
$$z(s) = (s-2).$$

Hence, $G(s)$ has poles $\{-1, -1, -2\}$ and a zero $\{2\}$.

Consider now a typical feedback control system which has a plant, a sensor, a controller $K(s)$, and an actuator as its principal components. One of the control objectives is to make the plant output track the reference (or set point) $R(s)$ as closely as possible in the presence of possible disturbances $V(s)$. For system analysis and design, the plant, sensor, and actuator are usually combined and viewed as a single object (still called the plant $G(s)$), and this gives rise to the most popular control configuration, the unity (output) feedback system, as depicted in Fig. 6.1.

We assume through the rest of this chapter that $G(s)$ and $K(s)$ are proper and that $\det(I + G(\infty)K(\infty)) \neq 0$. The bounded input, bounded output stability (BIBO stability) is usually defined with regard to dynamic behavior relating the set point R to the output Y,

$$Y = GK(1+GK)^{-1}R,$$

and is determined by stability of its transfer function matrix $GK(1+GK)^{-1}$. But the BIBO stability does not guarantee internal stability of overall systems which involves several input-output pairs. For the system in Fig. 6.1 where there are two dynamic subsystems, it suffices to consider the stability of the mapping from two vector signals outside the feedback loop and two vector signals within the loop. For instance, one may consider

$$\begin{bmatrix} E(s) \\ U(s) \end{bmatrix} = H(s) \begin{bmatrix} R(s) \\ V(s) \end{bmatrix},$$

where

$$H = \begin{bmatrix} (I+GK)^{-1} & -(I+GK)^{-1}G(s) \\ (I+KG)^{-1}K(s) & (I+KG)^{-1} \end{bmatrix}^{-1}. \tag{6.1}$$

Define the characteristic polynomial of the closed-loop system to be

$$p_c := p_G p_K \det[I+GK] = p_G p_K \det[I+KG],$$

where p_G and p_K are the pole polynomials of $G(s)$ and $K(s)$, respectively. Suppose that both $G(s)$ and $K(s)$ are proper and $\det[I + G(\infty)K(\infty)] \neq 0$. The interconnected system described in Fig. 6.1 is called *internally stable* if and only if $H(s)$ in (6.1) is stable, that is, all its poles are in the open left half of the complex plane [3].

Theorem 6.1 *The following are equivalent*:

1. *The system in Fig.* 6.1 *is internally stable*, *or* $H(s)$ *is stable.*
2. *The characteristic polynomial of the closed-loop system*, p_c, *is a stable polynomial.*

The important message conveyed by Theorem 6.1 is that internal stability of the system is equivalent to (input–output, or BIBO) stability of the augmented $H(s)$. But note that all four blocks H_{ij}, $i, j = 1, 2$, in H must be stable to ensure internal stability of the feedback system. Otherwise, the system cannot be internally stable while some (but not all) of H_{ij} is stable. This is precisely the problem associated with input–output stability, but it can be revealed by internal stability analysis.

Example 6.2 Let

$$G(s) = \begin{bmatrix} \frac{1}{s+1} & 0 \\ \frac{1}{s-1} & \frac{1}{s+1} \end{bmatrix} \quad \text{and} \quad K(s) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

GK has no pole-zero cancellation as K has neither zeros nor poles. Further, one sees that

$$\det[I + GK] = \left(1 + \frac{1}{s+1}\right)^2 = \left(\frac{s+2}{s+1}\right)^2,$$

whose numerator has stable roots only, so $(I + GK)^{-1}$ is stable. However, simple calculations give

$$p_c(s) = p_G p_K \det[I + GK] = (s-1)(s+1)^2 \cdot 1 \cdot \left(\frac{s+2}{s+1}\right)^2 = (s-1)(s+2)^2,$$

an unstable polynomial. Thus, the system is not internally stable. In fact, the unstable off-diagonal element gets no feedback for possible stabilization. The cause for instability here is not because that there is any unstable pole-zero cancellation between $G(s)$ and $K(s)$, but because that some unstable pole is lost when performing $\det[I + GK]$, which can never happen for SISO case!

Let the controller be in the form of $K(s) = kI$ for scalar k. It follows from Theorem 6.1 that

$$p_c(s) = p_G(s) \det\big[I + kG(s)\big],$$

where $p_c(s)$ is the pole or the characteristic polynomial of the closed-loop system, $p_G(s)$ is the pole polynomial of $G(s)$. Let $\lambda_i(s)$ be an eigenvalue of $G(s)$. The graphs of $\lambda_i(s)$ for the Nyquist Contour ($s = j\omega$) are called the characteristic loci.

Theorem 6.2 (Generalized Nyquist Theorem [4]) *If* $G(s)$ *has* P_o *unstable poles*, *then the negative feedback system consisting of* $G(s)$ *and* kI *is stable if and only if the characteristic loci of* $kG(s)$, *taken together*, *encircle the point* $-1 + j0$ P_o *times anticlockwise*, *assuming that* $G(s)$ *has no hidden unstable modes.*

6.2 MIMO PID Controller Tuning Based on Gain and Phase Margins

Gain and phase margins are often used to tune PID controllers for industry processes, which is one of the most common control schemes for single-input and single-output (SISO) systems [5–7] . The main reason for this is that the gain and phase margins have served as important measures of stability robustness in control loop unlike the H_∞, H_2, l^1, and μ methods, which often lead to fragile controllers [8]. A well-known design procedure for SISO systems is Astrom's [9] method. The fundamental step is estimating the critical gain and the critical frequency by a relay test. A controller can be found straightforward by using the critical point. Ho et al. [10] presented a tuning method with simplifications on the structure of the process to be a first-order plus dead-time model based on some approximate analysis. Unlike such simple model-based methods, Fung et al. [11] proposed a graphical method for PI controller tuning in which exact margins can be accomplished regardless of the process order, time delay, or damping nature.

In a multiinput and multioutput (MIMO) system, the tuning problem becomes complicated because of the interactions between control loops [12]. For easy understanding and implementation, decentralized control is the most common control scheme for MIMO systems as it has simpler structure and fewer tuning parameters to handle [13–15]. Many processes in industry are naturally MIMO systems, but a limited number of works have addressed the tuning controller for MIMO systems in terms of gain and phase margins. Ho et al. [16] proposed a tuning method based on the Gershgorin bands of MIMO systems. The tuning of decentralized PID controllers is used to meet the user-specified gain and phase margins in each loop. However, Kookos [17] pointed out that this method can be applied only in the case where the open-loop system is column dominant. Huang et al. [18] decomposed a MIMO system into a number of equivalent single loops together with some effective open-loop processes (EOPs). Based on the EOPs, a model-based method aimed at having desired gain and phase margins was presented to derive multiloop PI/PID controllers. In this method, model reductions and approximations are necessary to obtain the EOPs, which will inevitably bring design error.

In this section, we focus on TITO processes, which is the most common multivariable system in industry processes [19, 20], and a new method for tuning of decentralized PI controller is developed. For a TITO process controlled by a decentralized PI controller, two pairs of gain and phase margins can be specified. The main issue with MIMO controller tuning is that the couplings between loops make the equivalent process in one loop depend on the unknown controller in the other loop in a nonlinear way. We propose a simple yet powerful graphical method with least iteration to design controllers loop by loop based on the equivalent processes. The design procedure is given and a simulation example is provided to show the effectiveness of the proposed method.

6.2.1 Problem Formulation and Preliminaries

Let a stable TITO process be described by its transfer function matrix,

$$G(s) = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}.$$

The process is to be controlled in a unity output feedback configuration depicted in Fig. 6.2 by the diagonal conventional PI controller,

$$C(s) = \begin{bmatrix} c_1(s) & 0 \\ 0 & c_2(s) \end{bmatrix},$$

where each controller is parameterized as

$$c_l(s) = k_{pl} + \frac{k_{il}}{s}, \quad l = 1, 2.$$

It follows from Fig. 6.2 that

$$y_1 = g_{11}(s)u_1 + g_{12}(s)u_2,$$
$$y_2 = g_{21}(s)u_1 + g_{22}(s)u_2.$$

When the second loop is closed, that is, $u_2 = -c_2(s)y_2$, the equivalent open-loop transfer function between y_1 and u_1 is given by

$$g_1(s) = g_{11}(s) - \frac{g_{12}(s)g_{21}(s)c_2(s)}{1 + g_{22}(s)c_2(s)}. \tag{6.2}$$

Similarly, the equivalent open-loop transfer function between y_2 and u_2 is given by

$$g_2(s) = g_{22}(s) - \frac{g_{12}(s)g_{21}(s)c_1(s)}{1 + g_{11}(s)c_1(s)}. \tag{6.3}$$

Assume that the specifications of gain and phase margin for each loop are (A_{m1}, ϕ_{m1}) and (A_{m2}, ϕ_{m2}), respectively. Then, the problem becomes how to design PI controllers $c_1(s)$ and $c_2(s)$ for $g_1(s)$ and $g_2(s)$, respectively, to achieve the desired gain and phase margins in each loop.

Because of interactions between control loops, it will encounter more difficulties to design of such controller than that for a SISO case. For instance, the equivalent process (6.2) becomes uncertain since the unknown controller $c_2(s)$ is included. Also, the unknown controller $c_1(s)$ is included in the equivalent process (6.3). To deal with such a problem, Huang et al. [18] proposed a method to estimate the effective open-loop processes for each loop by model reductions and approximations, which leads to design error in both loops inevitably. In this section, we propose a one-step iteration of a graphical method to design controllers loop by loop. Firstly, we design an initial controller in one loop based on some uncertainty estimation

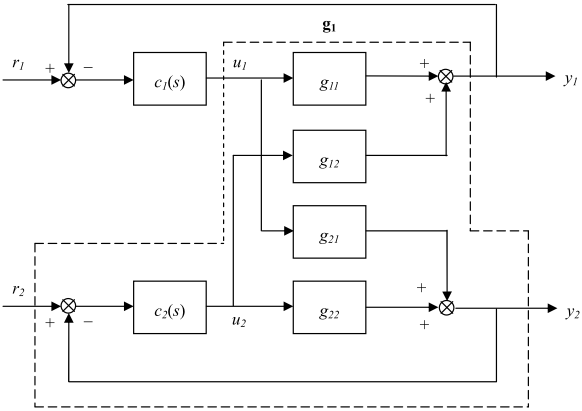


Fig. 6.2 The equivalent open-loop transfer function between y_1 and u_1

due to unknown 2nd loop controller; then design the second-loop controller with the known first-loop controller; Finally, retune the first-loop controller with the known 2nd-loop controller. In such a strategy, the key problem is how to design an initial controller for one loop with an unknown controller in other loop. The details will be given in the next section.

6.2.2 The Proposed Method

Without loss of generality, we design an initial controller for loop 1. To deal with the equivalent uncertain system (6.2), we define the uncertainty to be

$$\Delta_1(s) = \frac{g_{22}(s)c_2(s)}{1+g_{22}(s)c_2(s)}, \quad \left|\Delta_1(s)\right| \le \Delta_1^*,$$

where Δ_1^* is the maximum magnitudes of $\Delta_1(s)$, and we will estimate it later. Then, the open-loop transfer function of loop 1 is given by

$$g_1(s)c_1(s) = g_{11}(s)c_1(s) - \frac{g_{12}(s)g_{21}(s)c_1(s)}{g_{22}(s)}\Delta_1(s). \tag{6.4}$$

With the uncertainty $\Delta_1(s)$, the Nyquist curve of (6.4) will be embodied by a Nyquist band, which is formed by the circle with center $g_{11}(j\omega)c_1(j\omega)$ and radius $|\frac{g_{12}(j\omega)g_{21}(j\omega)c_1(j\omega)}{g_{22}(j\omega)}\Delta_1^*|$, respectively.

To design an initial controller in loop 1, we consider such Nyquist band and set the gain and phase margins for it to be $A_{m1}^o(A_{m1}^o < A_{m1})$ and $\phi_{m1}^o(\phi_{m1}^o < \phi_{m1})$, respectively. Note that the Nyquist curve of $g_{11}(s)c_1(s)$ is the central line of the Nyquist band (6.4). Figures 6.3 and 6.4 show a typical Nyquist curve of diagonal element $g_{11}(s)c_1(s)$ and its uncertainty disk. To start, we assume that the gain and phase crossover frequencies on the Nyquist band of (6.4) and its center curve

$g_{11}(s)c_1(s)$ are identical with each other, which is the same assumption as Ho's case [16], but it will be used only for the initial controller design. This implies that

1. $\angle g_{11}(j\omega_{p1})c_1(j\omega_{p1}) = -\pi$, where ω_{p1} is the phase crossover frequency on Nyquist bands of (6.4).
2. $|g_{11}(j\omega_{g1})c_1(j\omega_{g1})| = 1$, where ω_{g1} is the gain crossover frequency on Nyquist bands of (6.4).

Under the above assumptions, the gain and phase margins of $g_{11}(s)c_1(s)$ can be expressed by

$$A'_{m1} = A^o_{m1}\left(1 + \frac{|g_{12}(j\omega_{p1})g_{21}(j\omega_{p1})|}{|g_{11}(j\omega_{p1})g_{22}(j\omega_{p1})|}\Delta^*_1\right), \tag{6.5}$$

$$\phi'_{m1} = \phi^o_{m1} + 2\arcsin\left(\frac{\Delta^*_1|g_{12}(j\omega_{g1})g_{21}(j\omega_{g1})|}{2|g_{11}(j\omega_{g1})g_{22}(j\omega_{g1})|}\right), \tag{6.6}$$

respectively. They imply that

$$g_{11}(j\omega_{p1})\left(k_{p1} - j\frac{k_{i1}}{\omega_{p1}}\right) = -\frac{1}{A'_{m1}}, \tag{6.7}$$

$$g_{11}(j\omega_{g1})\left(k_{p1} - j\frac{k_{i1}}{\omega_{g1}}\right) = -\exp\bigl(j\phi'_{m1}\bigr). \tag{6.8}$$

Note that (6.7) and (6.8) are two nonlinear and complex equations with four unknowns, that is, two controller parameters and two crossover frequencies. To solve (6.7) and (6.8), an effective method is proposed by Fung et al. [11]. The equations are solved by a graphical method, and the two parameters of the controller are determined by intersection points. According to Fung et al.'s method [11], we define following two complex functions:

$$\begin{aligned} f_{p1}(\omega_{p1}) &= \mathrm{Re}\left[\frac{-1}{A'_{m1}g_{11}(j\omega_{p1})}\right] - j\omega_{p1}\,\mathrm{Im}\left[\frac{-1}{A'_{m1}g_{11}(j\omega_{p1})}\right], \\ &-\frac{\pi}{2} < \angle g_{11}(j\omega_{p1}) < -\pi, \end{aligned} \tag{6.9}$$

$$\begin{aligned} f_{g1}(\omega_{g1}) &= \mathrm{Re}\left[\frac{-\exp(j\phi'_{m1})}{g_{11}(j\omega_{g1})}\right] - j\omega_{g1}\,\mathrm{Im}\left[\frac{-\exp(j\phi'_{m1})}{g_{11}(j\omega_{g1})}\right], \\ &-\frac{\pi}{2} + \phi'_{m1} < \angle g_{11}(j\omega_{g1}) < -\pi + \phi'_{m1}, \end{aligned} \tag{6.10}$$

to draw graphics. It is necessary to point out that, in Fung et al.'s [11] method, A'_{m1} and ϕ'_{m1} in (6.9) and (6.10) are constant values, while in our case, they are determined by (6.5) and (6.6) for each frequency.

Then, we give a method to estimate the maximum magnitude Δ^*_1 of the uncertainty $\Delta_1(s)$. Since $c_2(s)$ is still unknown, it is impossible to get its value exactly.

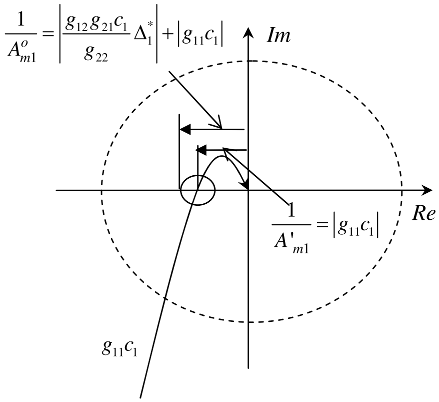

$\frac{1}{A^o_{m1}} = \left|\frac{g_{12}g_{21}c_1}{g_{22}}\Delta_1^*\right| + |g_{11}c_1|$
Im
Re
$\frac{1}{A'_{m1}} = |g_{11}c_1|$
$g_{11}c_1$

Fig. 6.3 Gain margin of Nyquist band of $g_1(s)c_1(s)$

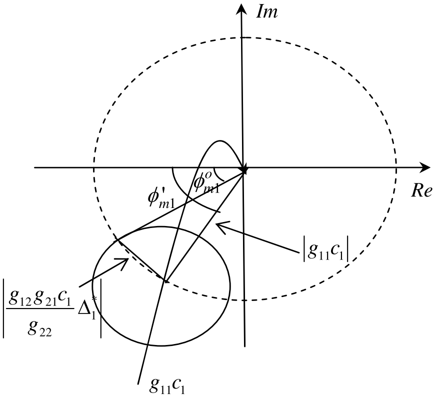


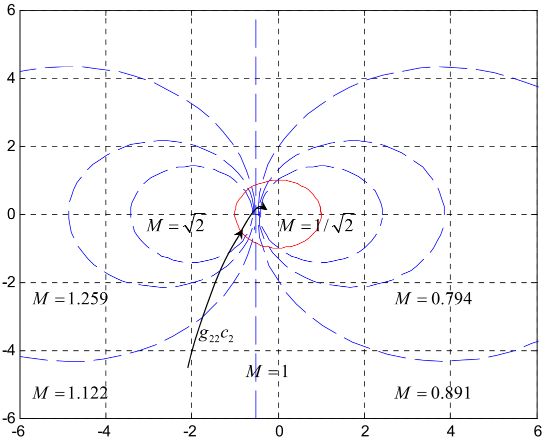
Fig. 6.4 Phase margin of Nyquist band of $g_1(s)c_1(s)$

Due to the fact of $\Delta_1(s) = \frac{g_{22}(s)c_2(s)}{1+g_{22}(s)c_2(s)}$, we firstly discuss the properties of the equation

$$\left|\frac{z}{1+z}\right| = M. \tag{6.11}$$

As shown in Fig. 6.5, the trajectories of z for different values of M are plotted in the complex plane, which are often called M-circles. We can find that all those M-circles for which $M > 1$ are located at the plane of $\mathrm{Re}(z) < -\frac{1}{2}$, and the circle becomes smaller with M increasing; and that all those M-circles for which $0 < M < 1$ are located at the plane of $\mathrm{Re}(z) > -\frac{1}{2}$, and the circle becomes smaller with M decreasing. Typically, the Nyquist curve of $g_{22}(s)c_2(s)$ (or the trajectory of z) starts from positive real axis and moves to the unit circle with $|\Delta_1(s)|$ (or M) increasing; then, the Nyquist curve (or the trajectory of z) falls into the unit circle and tends to origin quickly with $|\Delta_1(s)|$ (or M) decreasing. So, the maximum $|\Delta_1(s)|$ is achieved approximately when $|g_{22}(s)c_2(s)| = |e^{(-\pi+\theta)j}| = 1$, where θ is the phase margin of $g_{22}(s)c_2(s)$. Because the Nyquist curve of $g_{22}(s)c_2(s)$ locates

Fig. 6.5 M-circles



in the Nyquist band of $g_2(s)c_2(s)$, we take $\theta = \lambda\phi_{m2}$, where $\lambda \in (0.5, 1.5)$. So Δ_1^* can be taken approximately as

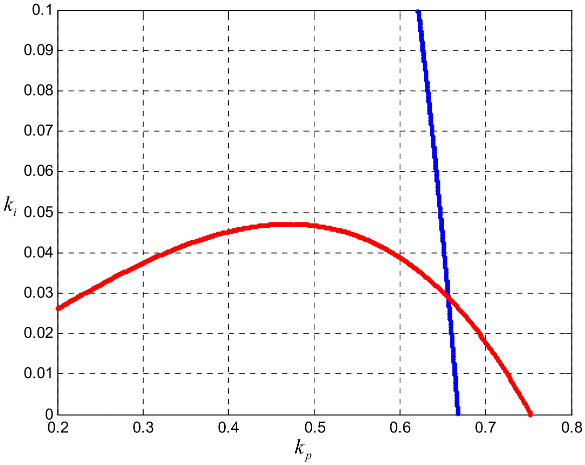
$$\Delta_1^* = \left|\frac{g_{22}(s)c_2(s)}{1 + g_{22}(s)c_2(s)}\right|_{\max} \approx \left|\frac{e^{(-\pi+\theta)j}}{1 + e^{(-\pi+\theta)j}}\right|. \tag{6.12}$$

In this way, we can get the initial controller $c_1(s)$ in loop 1 to make the Nyquist band of $g_1(s)c_1(s)$ approximately achieve the gain margin A_{m1}^o and the phase margin ϕ_{m1}^o. Substitute this controller into (6.3), and the effective open-loop process is available with knowledge of $c_1(s)$. Then Fung et al.'s [11] method can be used directly to design $c_2(s)$ for loop 2 to achieve the desired gain margin A_{m2} and phase margin ϕ_{m2}. Finally, substitute $c_2(s)$ in to (6.2) and re-tune $c_1(s)$ in loop 1 with knowledge of $c_2(s)$, based on the desired gain margin A_{m1} and the phase margin ϕ_{m1}.

In our design procedure, $c_1(s)$ has to be designed twice. The first time is for the initial one, and the second time is for the final one. To design the initial one, we use the enveloping curve of the Nyquist band of $g_1(s)c_1(s)$. Though this step of design is conservative and approximate in nature, it helps get $c_1(s)$ without knowing the controller in other loop; and further, this initial controller $c_1(s)$ not only stabilizes the equivalent process (6.2) but also enable the resulting loop to have the gain and phase margins close to the desired ones. It is obvious that both the initial controller and the finial controller of $c_1(s)$ have similar dynamic behavior, and the effect of retuning $c_1(s)$ to the loop 2 equivalent frequency response would be very limited and will bring in only a little design error. Thus, retuning of $c_2(s)$ is not required, and our design is able to stop merely after one iteration [21].

It should be pointed out that since most of PID controllers used in the process industry are actually of PI type with the derivative action turned off [22–24], we only discussed the case of PI controller in our design procedure. In the case where D-action is really needed, our design method is still applicable with trivial changes to PD controller and PID controller if some common relation between derivative and integral gains is adopted.

Fig. 6.6 The graphics for $c_1(s)$



6.2.3 An Example

Consider the well-known Wood and Berry binary distillation column plant [25]:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} \frac{3.8e^{-8s}}{14.9s+1} \\ \frac{4.9e^{-3s}}{13.2s+1} \end{bmatrix} d(s).$$

The gain and phase margins are set to 2.5, 60° and 2.5, 50° for loop 1 and loop 2, respectively. Firstly, to obtain an initial controller in loop 1, we set the gain and phase margins for the Nyquist band of $g_1(s)c_1(s)$ to be 2, 30°, respectively. Choose $\lambda = 1.1$, and then we get $\Delta_1^* = 1.0828$. The graphs of (6.9) and (6.10) are plotted with respect to ω_{p1} and ω_{g1} by Fung et al.'s [11] method in Fig. 6.6, and the intersection point is found to be at (0.6564, 0.0289), which gives the resultant PI controller for loop 1 as

$$c_1(s) = 0.6564 + \frac{0.0289}{s}.$$

Now, we substitute $c_1(s)$ into the equivalent process (6.3) and use Fung et al.'s [11] method again to design $c_2(s)$ for the process $g_2(s)$. As shown in Fig. 6.7, an intersection point (−0.0484, −0.01857) is found, and the resultant PI controller for loop 2 is given as

$$c_2(s) = -0.0484 - \frac{0.01857}{s}.$$

Substitute $c_2(s)$ into the equivalent process (6.2) and retune the controller $c_1(s)$. The graphics and the intersection point are given in Fig. 6.8. The controller is given as

$$c_1(s) = 0.802 + \frac{0.046}{s}.$$

Fig. 6.7 The graphics for $c_2(s)$

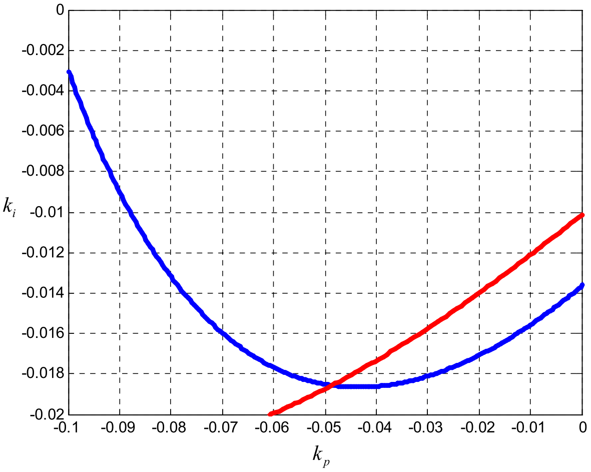
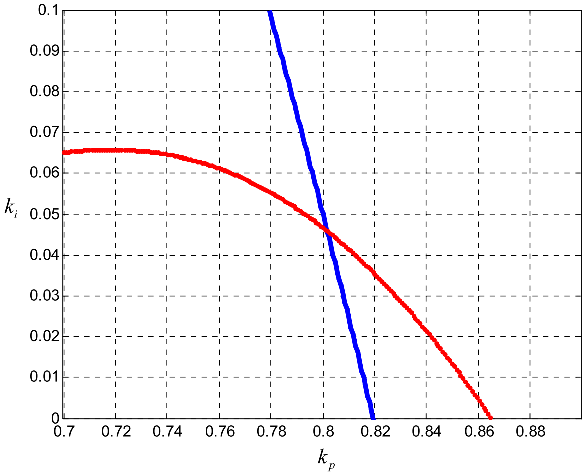


Fig. 6.8 The graphics for $c_1(s)$ retuning



Nyquist curves of $g_1(s)c_1(s)$ and $g_2(s)c_2(s)$ are plotted in Figs. 6.9 and 6.10, which show that the achieved gain and phase margins for loop 1 are 2.5 and 60°, respectively, and the achieved gain and phase margins for loop 2 are 2.551 and 51.7553° respectively. One sees that the gain and phase margins specifications are achieve exactly in loop 1 and approximately achieved in loop 2.

The resultant closed-loop output response to unit step set-point change and step disturbance changes of magnitude of 0.2 are shown in Fig. 6.11, and the control signals are shown in Fig. 6.12. The performance of the proposed method is compared with those given by the BLT [26] method and Ho's [16] method. It is apparent from the figures that the proposed controller provides superior performance for both set-point tracking and disturbance rejection with reasonable control signals.

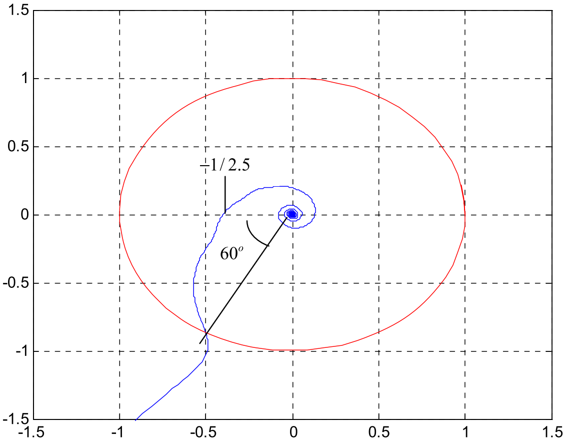


Fig. 6.9 Nyquist curve of $g_1(s)c_1(s)$

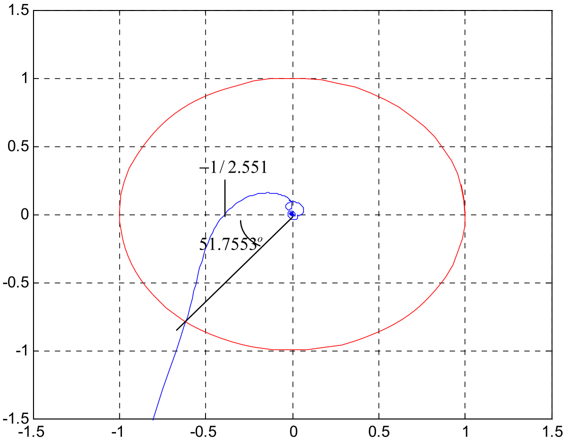


Fig. 6.10 Nyquist curve of $g_2(s)c_2(s)$

6.3 MIMO Loop Gain Margins

The SISO gain margin is one of most popular tools for control system analysis and design in the process control industry due to its well-posedness, easy computation, and clear measure of stability and performance [7, 9, 27–29]. A MIMO gain margin would have enjoyed the same degree of popularity if it could also have been developed similarly and successfully. But unfortunately, historically, MIMO control studies started with the state space approach, which was thought to be much powerful than traditional frequency domain methods which seemed to be losing favor at this early time of modern control. Only when one began to address model uncertainties and robust control, there was renewed interest in frequency domain approach for both SISO and MIMO cases. However, in the frequency domain, the gain margin cannot be extended straightforward to a MIMO system, because of the interactions between the control loops [18, 30, 31]. In this section, we introduce the MIMO loop gain margins which measure the simultaneous largest changes of all

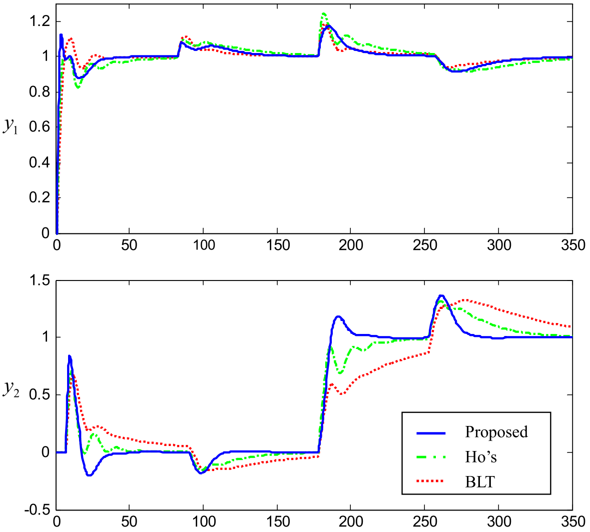


Fig. 6.11 Closed-loop time responses of the example

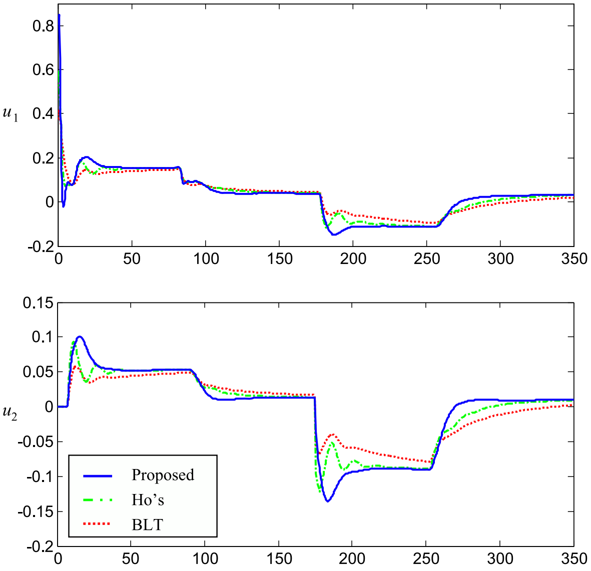


Fig. 6.12 Control signals of the example

loop gains while preserving closed-loop stability. This is different from the SISO gain margin, where there is only one loop and one gain change, and the MIMO sequential version of gain margin where only one loop gain change at a time is addressed.

Doyle [32] developed the μ-analysis as an effective tool for robust stability analysis in multivariable feedback control. As a method in the frequency domain, the μ-analysis treats system uncertainties as complex valued and inevitably brings the conservativeness for gain margins computation because the gain change in each loop is a real number. Then, Fan, Tits, and Doyle [33] proposed a computation of μ for mixed real and complex uncertainty to deal with this problem. However, Braatz et al. [34] pointed out that it was an NP-hard problem to exactly calculate μ with pure real or mixed uncertainty. So, the μ-analysis method for real uncertainty is seldom used to calculate the gain margins of a MIMO system. Safonov and coworkers [35–37] employed several different methods, such as Generalized Conic Sectors, Perron–Frobenius nonnegative matrix results and the Zadeh–Desoer mapping theorem, to calculate the stability margin. But their stability margin was defined by a single parameter k_m ($k_m := \min_{\Delta}\{k \in [0, \infty) \det(I - k\Delta G) = 0\}$, where Δ is a diagonal complex perturbation, and G is the system transfer function matrix), which represents the common gain change for all loops. Thus, this method does not allow independent and simultaneous loop gain changes. Baron and Jonckheere [38] defined the gain margins for a MIMO system as the minimal complex matrix perturbation before the system goes to instability. Such a definition allows the gain perturbation to be a full matrix, not necessarily restricted to be diagonal. However, it does not reflect the gain changes of individual loops, which practical control engineers have been used to. Li and Lee [39] showed that the H_∞ norm of a sensitivity function matrix for a stable multivariable closed-loop system is related to some common gain and phase margins for all the loops. The Gershgorin band method is also used to calculate the gain margins of MIMO systems [16, 40]. It gives conservative results because it requires the diagonal dominance of the system, which brings some limitations to its applications.

Recently, the loop gain margin for a MIMO system was introduced by Wang et al. [12]. They first computed the stabilizing parameter ranges of a multiloop proportional controller by a quasi-LMI technique and then take their rectangular subset to obtain the loop gain margins. They show better results than the μ-analysis method. The loop gain margins obtained with this approach are indeed stability margins but not the exact or the maximum margins available due to the inherent conservativeness of the LMI framework. Besides, the method was developed for delay-free systems only. If the plant has time delay, one has to make approximations for time delay to apply this method. Fundamentally, it is always desirable to find the exact or the maximum controller parameter regions for stabilizing a given process.

In view of the above observations, this section aims to remove the abovementioned conservativeness and compute exact loop gain margins of MIMO systems. We have to discard a time domain/Lyapunov/LMI framework, which is the root cause of conservativeness, and work in a frequency domain instead. We first transform the fundamental MIMO margin equation with the diagonal structure of gain perturbations into a constrained quadratic optimization problem. Next, we utilize the Lagrange multiplier method to obtain an unconstrained optimization problem, whose necessary/stationary condition is numerically solved by the Newton–Raphson iteration algorithm and whose sufficient condition is verified by the eigenvalue check.

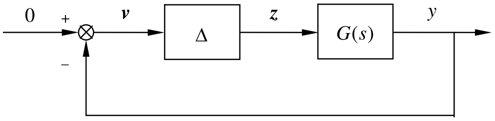


Fig. 6.13 Diagram of a MIMO control system

6.3.1 The Proposed Approach

Consider the conventional unity output feedback system depicted in Fig. 6.13. It follows from Wang et al. [12] that the loop gain margin problem of MIMO feedback systems is formulated as follows.

Problem For an $m \times m$ open-loop process model, $G(s)$, under the decentralized gain controller, $\Delta = \mathrm{diag}\{k_1, \ldots, k_m\}$, find the ranges, $(\underline{k_i}, \overline{k_i})$, $i = 1, \ldots, m$, such that the closed-loop system is stable when $k_i \in (\underline{k_i}, \overline{k_i})$ for all i, but marginally stable when $k_i = \underline{k_i}$ or $k_i = \overline{k_i}$ for some i. For the special case of the common gain controller, $K(s) = kI_m$, the resulting solution $k \in [\underline{k}, \overline{k}]$ is called the common gain margin of the system.

Let the plant $G(s)$ and the diagonal controller Δ be described by

$$G(s) = \begin{bmatrix} g_{11} & \cdots & g_{1m} \\ \vdots & \ddots & \vdots \\ g_{m1} & \cdots & g_{mm} \end{bmatrix}, \qquad \Delta = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & k_m \end{bmatrix}.$$

Note that unlike the standard robust stability analysis for which the nominal case means $\Delta = 0$, our nominal case is for no gain perturbation, that is, $\Delta = I_m$, unity gain or no gain change at all. In the frequency domain, the generalized Nyquist stability criterion [4] is widely used to examine the stability of the closed-loop system. Under the nominal stabilization, the closed-loop system can be destabilized if and only if there is a gain perturbation Δ such that

$$\det\bigl(I + G(j\omega)\Delta\bigr) = 0. \tag{6.13}$$

In this section, we try to trace all the diagonal gain perturbations Δ that satisfy (6.13). It follows from linear algebra that (6.13) holds if and only if there exists some nonzero unit vectors $z \in C^m$ such that

$$z = -\Delta G z \quad \text{or} \quad (I + \Delta G)z = 0. \tag{6.14}$$

Baron and Jonckheere [38] found the solution to (6.14) by recasting it into a constrained optimization, but for a full (nondiagonal) matrix which would not correspond to any loop gain or phase perturbations in practical situations. Their method is thus not applicable to our case of the diagonal real matrix, Δ.

Consider a diagonal real matrix Δ. Let $v = [v_1, v_2, \ldots, v_m]^{\mathrm{T}}$, and $z = [z_1, z_2, \ldots, z_m]^{\mathrm{T}}$. It follows from Fig. 6.13 that

$$v = -Gz \tag{6.15}$$

and $z = \Delta v$, which implies

$$z_i = k_i v_i \quad \text{or} \quad k_i = z_i / v_i, \quad i = 1, \ldots, m. \tag{6.16}$$

We try to find the solution z through an optimization technique and then obtain v and Δ from (6.15) and (6.16), respectively. We take the following quadratic function as the cost function for ease of optimization:

$$J = -v^* z = z^* G^* z.$$

This problem should be addressed subject to (6.14). But (6.14) involves additional Δ, which is inconvenient to deal with. We need to change (6.14) to a more tractable form based on the system configuration in Fig. 6.13. By (6.16), $v_i^* z_i = k_i v_i^* v_i = k_i \|v_i\|^2$ is a real number. Note that $v_i^* z_i = [0, \ldots, v_i^*, 0, \ldots][0, \ldots, z_i, 0, \ldots]^{\mathrm{T}} = v^* H^i z = -z^* G^* H^i z$, where $H^i = [h_{pq}]$ is given by

$$h_{pq} = \begin{cases} 1, & p = q = i, \\ 0, & \text{otherwise.} \end{cases}$$

Besides, the unit vector z meets $z^* z = 1$. Combining the above two conditions yields the constrain on z:

$$\begin{cases} z^* z = 1, \\ \mathrm{Im}(-z^* G^* H^i z) = 0, \quad i = 1, \ldots, m. \end{cases} \tag{6.17}$$

Therefore, we obtain our constrained optimization problem as

$$\begin{aligned} & \min\bigl(z^* G^* z\bigr) \quad \text{or} \quad \max\bigl(z^* G^* z\bigr), \\ & \text{s.t.} \quad \begin{cases} z^* z = 1 \\ \mathrm{Im}(-z^* G^* H^i z) = 0, \quad i = 1, \ldots, m. \end{cases} \end{aligned} \tag{6.18}$$

It should be pointed out that $z \in C^m$ will lead to the failure of solving the optimization problem (6.18) because neither the cost function nor the constraints are holomorphic functions of z [41]. Fortunately, it is convenient to convert (6.18) to an equivalent real constrained optimization problem by decomplexification, which is a standard technique described by Galin [42]. This process makes use of a canonical isomorphism between C^m and R^{2m}. Let $z_i = \mathrm{Re}(z_i) + j\,\mathrm{Im}(z_i)$; $v_i = \mathrm{Re}(v_i) + j\,\mathrm{Im}(v_i)$; $Z_i = [\mathrm{Re}(z_i), \mathrm{Im}(z_i)]^{\mathrm{T}}$; $V_i = [\mathrm{Re}(v_i), \mathrm{Im}(v_i)]^{\mathrm{T}}$; $\overline{Z} = [Z_1; \cdots; Z_i; \cdots]$; $\overline{V} = [V_1; \cdots; V_i; \cdots]$; $g_{pq} = \mathrm{Re}(g_{pq}) + j\,\mathrm{Im}(g_{pq})$; $G_{pq} = \left[\begin{smallmatrix} \mathrm{Re}(g_{pq}) & -\mathrm{Im}(g_{pq}) \\ \mathrm{Im}(g_{pq}) & \mathrm{Re}(g_{pq}) \end{smallmatrix}\right]$; $i = 1, \ldots, m$, $p, q = 1, \ldots, m$;

$$\overline{G} = \begin{bmatrix} G_{11} & \cdots & G_{1m} \\ \vdots & & \vdots \\ G_{m1} & \cdots & G_{mm} \end{bmatrix}.$$

Note that $\mathrm{Im}(v_i^* z_i) = V_i^{\mathrm{T}} M Z_i$, where $M = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$, and it follows that $z^* G^* z = \overline{Z}^{\mathrm{T}} \overline{G}^{\mathrm{T}} \overline{Z}$; $z^* z = \overline{Z}^{\mathrm{T}} \overline{Z}$; $\mathrm{Im}(-z^* G^* H^i z) = -\overline{Z}^{\mathrm{T}} \overline{G}^{\mathrm{T}} \overline{H}^i W \overline{Z}$, where $W =$

$\mathrm{diag}\{M,\ldots,M\}$, $i=1,\ldots,m$, $\overline{H}^i=[h_{pq}]$ with

$$h_{pq}=\begin{cases}1, & p=q=2i \text{ or } 2i-1,\\ 0, & \text{otherwise.}\end{cases}$$

Thus, the constrained optimization (6.18) in C^m is equivalent to the following optimization problem in R^{2m}:

$$\begin{aligned}&\min\bigl(\overline{Z}^{\mathrm{T}}\overline{G}^{\mathrm{T}}\overline{Z}\bigr) \quad \text{or} \quad \max\bigl(\overline{Z}^{\mathrm{T}}\overline{G}^{\mathrm{T}}\overline{Z}\bigr),\\ &\text{s.t.} \quad \begin{cases}\overline{Z}^{\mathrm{T}}\overline{Z}=1,\\ \overline{Z}^{\mathrm{T}}\overline{G}^{\mathrm{T}}\overline{H}^i W\overline{Z}=0, \quad i=1,2,\ldots,m.\end{cases}\end{aligned} \tag{6.19}$$

To solve the above constrained optimization problem (6.19), we use the Lagrange multiplier technique:

$$f(\theta)=\overline{Z}^{\mathrm{T}}\overline{G}^{\mathrm{T}}\overline{Z}+\lambda_1\bigl(\overline{Z}^{\mathrm{T}}\overline{Z}-1\bigr)+\sum_{i=1}^{m}\lambda_{i+1}\bigl(\overline{Z}^{\mathrm{T}}\overline{G}^{\mathrm{T}}\overline{H}^i W\overline{Z}\bigr),$$

where $\theta=[\overline{Z}^{\mathrm{T}},\lambda_1,\lambda_2,\ldots,\lambda_{m+1}]^{\mathrm{T}}$. The stationary condition for optimality is

$$\begin{aligned}&q(\theta)=:\partial f(\theta)/\partial\theta\\ &=\begin{bmatrix}(\overline{G}^{\mathrm{T}}+\overline{G})\overline{Z}+2\lambda_1\overline{Z}+\sum_{i=1}^{m}\lambda_{i+1}[(\overline{G}^{\mathrm{T}}\overline{H}^i W)^{\mathrm{T}}+\overline{G}^{\mathrm{T}}\overline{H}^i W]\overline{Z}\\ \overline{Z}^{\mathrm{T}}\overline{Z}-1\\ \overline{Z}^{\mathrm{T}}\overline{G}^{\mathrm{T}}\overline{H}^1 W\overline{Z}\\ \vdots\\ \overline{Z}^{\mathrm{T}}\overline{G}^{\mathrm{T}}\overline{H}^m W\overline{Z}\end{bmatrix}=0.\end{aligned} \tag{6.20}$$

A numerical solution to (6.20) is sought by the Newton–Raphson algorithm:

$$\theta_{n+1}=\theta_n-J^{-1}\bigl[q(\theta_n)\bigr]q(\theta_n), \tag{6.21}$$

where

$$
\begin{aligned}
J\big[q(\theta_n)\big] & \\
&= \frac{\partial q(\theta_n)}{\partial \theta_n} = \frac{\partial^2 f(\theta_n)}{\partial \theta_n^2} \\
&= \begin{bmatrix}
\begin{matrix}(\overline{G}^{\mathrm{T}}+\overline{G})+2\lambda_1 I_{2m} \\ +\sum_{i=1}^{m}\lambda_{i+1}[(\overline{G}^{\mathrm{T}}\overline{H}^{i}W)^{\mathrm{T}} \\ +\overline{G}^{\mathrm{T}}\overline{H}^{i}W]\end{matrix} & 2\overline{Z} & \begin{matrix}[(\overline{G}^{\mathrm{T}}\overline{H}^{1}W)^{\mathrm{T}} \\ +\overline{G}^{\mathrm{T}}\overline{H}^{1}W]\overline{Z}\end{matrix} & \cdots & \begin{matrix}[(\overline{G}^{\mathrm{T}}\overline{H}^{m}W)^{\mathrm{T}} \\ +\overline{G}^{\mathrm{T}}\overline{H}^{m}W]\overline{Z}\end{matrix} \\
2\overline{Z}^{\mathrm{T}} & 0 & 0 & \cdots & 0 \\
\overline{Z}^{\mathrm{T}}[(\overline{G}^{\mathrm{T}}\overline{H}^{1}W)^{\mathrm{T}}+\overline{G}^{\mathrm{T}}\overline{H}^{1}W] & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & & \vdots \\
\overline{Z}^{\mathrm{T}}[(\overline{G}^{\mathrm{T}}\overline{H}^{m}W)^{\mathrm{T}}+\overline{G}^{\mathrm{T}}\overline{H}^{m}W] & 0 & 0 & \cdots & 0
\end{bmatrix}
\end{aligned}
$$

is the Jacobian matrix of $q(\theta_n)$. If J is singular, then a Moore–Penrose inverse is used.

Since (6.19) involves quadratic functions only, the iteration routine will converge, and the convergent point gives the minimum or maximum when the constraints have a nonempty admissible set, that is, the nonzero vector $\overline{Z}$ satisfies all the constraint conditions in (6.19). To see whether the iteration routine achieves the maximum or minimum, the eigenvalues of the Hessian matrix

$$
H = \big(\overline{G}^{\mathrm{T}}+\overline{G}\big)+2\lambda_1 I_{2m}+\sum_{i=1}^{m}\lambda_{i+1}\big[\big(\overline{G}^{\mathrm{T}}\overline{H}^{i}W\big)^{\mathrm{T}}+\overline{G}^{\mathrm{T}}\overline{H}^{i}W\big] \tag{6.22}
$$

are calculated, and they should be all nonnegative for the minimum case and all nonpositive for the maximum case. To find, say, the maximum from the minimum, a new search is carried out with the initial search direction set as opposite to the eigenvector of H corresponding to the largest positive eigenvalue.

Algorithm 6.1 Finding the gain solution to (6.13), given ω, $G(j\omega)$ and initial search vector θ_0.

Step 1. Run the Newton–Raphson iteration (6.21). If the iteration is convergent, obtain z, v and the gains from (6.21), (6.15), and (6.16), respectively; otherwise, stop.

Step 2. Calculate the eigenvalues of (6.22) and decide if the solution is for the minimum (or maximum) case. Set the new initial vector θ_0 as opposite to the eigenvector of H corresponding to the largest positive (or smallest negative) eigenvalue.

Step 3. Go to Step 1 once more for the maximization (or minimization).

6.3.2 Frequency Range Estimation

In principle, the optimization in (6.19) is carried out for $G(j\omega)$ for each frequency ω. Practically, it is too costly to do so and also unnecessary to do so since (6.19) usually has a solution for a small subset of the entire frequency interval of zero to infinity, recalling that a SISO system typically involves one or few frequency points (with phase of $-\pi$ or so) in determining its gain margin. Thus, we look for the relevant frequency range Ω such that the solution to (6.19) may exist while there is no solution in its complement set for which (6.19) will not be performed. From the constraint analysis in the previous section, we know that $v_i^* z_i = k_i v_i^* v_i = k_i \|v_i\|^2$ is real, so that $v^* z = \sum_{i=1}^m k_i \|v_i\|^2$ is also real, which leads to

$$\mathrm{Im}(v^* z) = \frac{v^* z - z^* v}{2i} = z^* \left(\frac{G(j\omega) - G(j\omega)^*}{2i} \right) z = 0. \tag{6.23}$$

Let

$$P(j\omega) = \frac{G(j\omega) - G(j\omega)^*}{2i}, \tag{6.24}$$

and let $\lambda(P(j\omega))$ be an eigenvalue of $P(j\omega)$. The set $\Psi(P) = \{z^* P z : z^* z = 1, z \in C^m\}$ is commonly called the numerical range of P [43]. Since $P(j\omega)$ is a Hermitian matrix, the numerical range of $P(j\omega)$ is the segment of the real axis bounded by the smallest and largest eigenvalues of $P(j\omega)$ [44]. As a result, if the eigenvalues of $P(j\omega)$ are spread across zero, that is, there are opposite-sign eigenvalues or zero eigenvalues, then the numerical range of $P(j\omega)$ contains the origin, and there will exist z satisfying (6.23) at the underlying frequency ω. This frequency is thus relevant; otherwise, it is not. Therefore, we can employ this condition to find relevant ω. To this end, we first find all the real roots ω_l such that $\det(P(j\omega_l)) = 0$. Note that $\lambda(P(j\omega))$ is a continuous function of ω, and between two consecutive roots ω_l and ω_{l+1}, the sign distribution of all $\lambda(P(j\omega))$ does not change. This implies that $\lambda(P(j\omega))$ is always greater or less than zero for all $\omega \in (\omega_l, \omega_{l+1})$. Hence, by calculating all $\lambda(P(j\omega))$ for one $\omega \in (\omega_l, \omega_{l+1})$, we know their sign distribution and can then determine whether or not (6.23) may have a solution. If the answer is yes, we assign $(\omega_l, \omega_{l+1}) \subseteq \Omega$.

It follows from Fig. 6.13 that $v_i^* z_i = k_i v_i^* v_i = k_i \|v_i\|^2$should be a real number for all i. If this is the case, we have (6.23), which has been used to determine the frequency range Ω. Thus, the so-calculated Ω contains all ω which satisfy the constraint of real $v_i^* z_i = k_i v_i^* v_i = k_i \|v_i\|^2$ for all i. But the converse is not necessarily true, that is, (6.23) does not imply $v_i^* z_i = k_i v_i^* v_i = k_i \|v_i\|^2$ for all i. Hence, the calculation by numerical range method to meet (6.23) may not be necessary, and the calculated Ω is overestimated.

There is a complex issue caused by high order or time delay of the system. For a high order or time delay process, the characteristic loci can pass through the real axis many or infinite times as the frequency goes to infinity, which implies that there will be many or infinite frequency intervals in Ω, most of which will be shown to be irrelevant in computing the stability boundaries (recall that we overestimated Ω

before). Our next task is to remove these irrelevant ones in Ω to further reduce computational burden.

As a motivation of our idea in this development, let us first consider the SISO case. Let the phase-crossover frequencies of the time delay SISO process $G(j\omega)$ be $\omega_1, \dots, \omega_l, \dots$, then, the gain solutions to $1 + G(j\omega)K = 0$ at these frequencies are $K_l = -1/G(j\omega_l)$, $l = 1, \dots$. $[1, K_l]$ defines one possible gain margin. The gain margin is the intersection of all intervals, $[1, K_l]$. Suppose that you have done calculations up to K_l. For $l + 1$, if K_{l+1} is larger than K_l, K_{l+1} should be discarded because intersection of $[1, K_l]$ and $[1, K_{l+1}]$ is $[1, K_l]$. Physically, if a smaller gain interval destabilizes the plant, so does a larger interval. The above analysis essentially needs to calculate the smallest K_l denoted by K^* which will determine the gain margin of $G(j\omega)$. If the plant has monotonically decreasing gain of its frequency response, $|G(j\omega_a)| > |G(j\omega_b)|$ for $\omega_a < \omega_b$, then one only needs to calculate the first (or lowest) phase-crossover frequency and the corresponding gain solution to determine the gain margin of the plant. More realistically, the plant may not necessarily have a monotonically decreasing gain of its frequency response over an entire frequency range but would almost always be so after a certain high frequency. Then, one can stop calculating gain solutions to $1 + G(j\omega)K = 0$ at that frequency onwards as their solutions will not affect or reduce the gain margin.

The situation of a MIMO process would be similar to the SISO case in the sense that the intersection of all the sets enclosed by boundaries defined by gain solutions of (6.13) including $\Delta = I_m$ (with the assumed nominal stabilization) will be the stabilizing one. At the computational level, we calculate gain solutions from zero frequency range, one interval after another in Ω, till a closed region including $\Delta = I_m$ is formed. This region is then enlarged by a hypercube which encloses the region. Then, we will discard any remaining frequency intervals in Ω for which no gain solution lies inside the hypercube, which will not affect or reduce the previously determined region.

For ease of explanation, let us denote Ω by $\Omega = (\underline{\omega_1}, \overline{\omega_1}) \cup \cdots (\underline{\omega_r}, \overline{\omega_r}) \cup \cdots$ in the ascending order of the frequency, where $(\underline{\omega_r}, \overline{\omega_r})$ replaces every (ω_l, ω_{l+1}) defined and calculated before. Suppose that we have computed the gain solutions for $(\underline{\omega_1}, \overline{\omega_1}) \cup \cdots (\underline{\omega_n}, \overline{\omega_n})$ and formed a closed stability region including $\Delta = I_m$. We can readily find the maximum $|k_i|$ on the boundaries of this region and denote it by $k_{\max}$ and enlarge and enclose this region by a hypercube defined by $k_i \in [-k_{\max}, k_{\max}]$, $i = 1, \dots, m$. Note again that for any frequency $\omega^* > \overline{\omega_n}$, the gain solutions not inside the hypercube will not affect the stability boundaries and thus need not be calculated; Otherwise, the stability region may be reduced by the new boundaries in the hypercube. The key issue is then to check if the gain solutions of the remaining frequency intervals $(\underline{\omega_{n+1}}, \overline{\omega_{n+1}}), \dots$, lie inside the hypercube. Suppose that it is the case, that is, a gain solution $\Delta^* = \operatorname{diag}(k_1^*, \dots, k_m^*)$ to (6.13) is inside this hypercube, implying $k_i^* \in [-k_{\max}, k_{\max}]$, $i = 1, \dots, m$, and there is some eigenvalue $\lambda(G\Delta^*)$ such that $|\lambda(G\Delta^*)| = 1$. It is well known that the spectral radius forms a lower bound on any compatible matrix norm [45]:

$$\rho(G\Delta) = \max\bigl|\lambda_i(G\Delta)\bigr| \le \|G\Delta\|_1 \le \|G\|_1 \|\Delta\|_1. \tag{6.25}$$

It follows that

$$\begin{aligned} 1 &= \left|\lambda(G\varDelta^*)\right| \le \max\left|\lambda_i(G\varDelta^*)\right| \le \|G\|_1\|\varDelta^*\|_1 \\ &= \max\left(\left|k_1^*\right|, \dots, \left|k_m^*\right|\right)\|G\|_1 \le k_{\max}\|G\|_1. \end{aligned}$$

Hence, for any frequency $\omega^* \in (\underline{\omega_r}, \overline{\omega_r})$, where $r > n$, if $1 \le k_{\max}\|G(j\omega^*)\|_1$, the gain solution $\varDelta^*$ to (6.13) would lie inside the hypercube, and the corresponding frequency interval $(\underline{\omega_r}, \overline{\omega_r})$ will be taken into account further. Otherwise, $(\underline{\omega_r}, \overline{\omega_r})$ will be removed from $\varOmega$.

With the help of the above development, the effective frequency range $\varOmega$ can be determined. The solution $\varDelta$ to (6.13) is found by solving an equivalent constrained optimization problem (6.19) for each $\omega \in \varOmega$ if the nonzero $\overline{Z}$ exists. Collect all the gain solutions and plot them. The smallest common region including $\varDelta = I_m$ will be the stabilizing region. The loop gain margins are found by taking a rectangular set in the region.

Algorithm 6.2 Finding loop gain margins for the given plant G.

Step 1. Calculate all the real roots ω_l of $\det(P(j\omega)) = 0$, where $P(j\omega)$ is defined in (6.24). Divide the entire frequency interval of zero to infinity into $[0, \omega_1] \cup (\omega_1, \omega_2] \cup \cdots (\omega_l, \omega_{l+1}] \cup \cdots$.

Step 2. Take one $\omega \in (\omega_l, \omega_{l+1}]$ and calculate $\lambda(P(j\omega))$ for $l = 1, 2, \dots$. If $\lambda_{\max}(P(j\omega))\lambda_{\min}(P(j\omega)) \le 0$, then $(\omega_l, \omega_{l+1}] \subset \varOmega$; Arrange $\varOmega = (\underline{\omega_1}, \overline{\omega_1}] \cup \cdots (\underline{\omega_r}, \overline{\omega_r}] \cup \cdots$, which is in the ascending order of the frequency. Set $r = 1$.

Step 3. Run Algorithm 1 for $(\underline{\omega_r}, \overline{\omega_r}]$ to get gain solution of (6.13) and plot them. If a closed region including $\varDelta = I_m$ is formed, let $n = r$ and find the maximum $|k_i|$ on the boundaries of this region, and denote it as $k_{\max}$. Otherwise, redo this step with $r = r + 1$.

Step 4. Remove the frequency interval $(\underline{\omega_r}, \overline{\omega_r}]$ from $\varOmega$ if $\|G(j\omega)\|_1 < \frac{1}{k_{\max}}$ for all $\omega \in (\underline{\omega_r}, \overline{\omega_r}]$ and $r > n$.

Step 5. Run Algorithm 1 for all the remaining $(\underline{\omega_r}, \overline{\omega_r}]$ in $\varOmega$, $r > n$, if any, and plot them. Determine the refined stabilizing region and loop gain margins from it.

6.3.3 An Illustrative Example

Consider the time delay process

$$G(s) = \begin{bmatrix} \frac{s-1}{(s+1)(s+3)} & \frac{-0.5s+1}{0.5s^3+2.5s^2+s+3} \\ \frac{0.3}{s+2}e^{-0.3s} & \frac{0.5s+1}{s^2+2s+5}e^{-0.7s} \end{bmatrix}.$$

Form $P(j\omega)$ using (6.24) for this and calculate the real roots of $\det(P(j\omega)) = 0$ to get $\omega_1 = \pm 2.652$, $\omega_2 = \pm 2.876$, $\omega_3 = \pm 6.918$, $\omega_4 = \pm 11.135$, $\omega_5 = \pm 15.757$. The frequency range $[0, +\infty)$ is divided into $[0, 2.652] \cup (2.652, 2.876] \cup (2.876, 6.918] \cdots$. We check the condition of $\lambda(P(j\omega))_{\min}\lambda(P(j\omega))_{\max} \le 0$

for any one ω in each interval and find $\Omega = [0, 2.652] \cup (2.876, 6.918] \cup (11.135, 15.757] \cup \cdots$.

Note that, for the frequency $\omega = 0$, the system $G(j\omega)$ becomes a real matrix. In this situation, only one constraint condition $z^*z = 1$ is left. Using this constraint condition and substituting (6.15) into (6.16) produce the gain solutions

$$\begin{cases} k_1 = \frac{z_1}{-z_1/3 - z_2/3}, \\ k_2 = \frac{z_2}{-0.3z_1/2 - z_2/5}, \end{cases} \tag{6.26}$$

with one parameter z. The solution curve is marked by c and d in Fig. 6.14. For nonzero frequencies, Running Step 3 of Algorithm 2 for the first two frequency intervals, $[0, 2.652]$ and $(2.876, 6.918]$, yields a closed region including $\Delta = I_2$. The maximum value of $|k_i|$ on the boundaries of this region is $k_{\max} = 6.659$. Plot $\|G(j\omega)\|_1 = \max_j \sum_{i=1}^m |g_{ij}(j\omega)|$ with respect to ω and the horizontal straight line $\frac{1}{k_{\max}}$ in Fig. 6.15. One sees that $\|G(j\omega)\|_1$ marked by blue line is always below the red line $\frac{1}{k_{\max}}$ where $\omega > 8.2$. So, $(11.135, 15.757] \cup \cdots$ is removed from Ω. There is no remaining frequency interval for further consideration.

The region including $\Delta = I_2$ found above is thus the stabilizing one and is marked in yellow in Fig. 6.14. Let us, say, fix the range of k_1 as $k_1 \in [-2.0233, 1.1502]$. The maximum rectangle is then determined in the stable region, and the corresponding range of k_2 is $k_2 \in [-3.84, 3.6852]$. This leads to the following gain margin for this time delay system:

$$k_1 \in [-2.0233, 1.1502] \quad \text{and} \quad k_2 \in [-3.84, 3.6852].$$

Besides, the common gain margin is $k \in [-3.5544, 2.4115]$ marked by green dash and dot line.

For a time delay system, one cannot use the LMI method [12] directly but need to make an approximation to time delay terms firstly. Suppose that we make the Pade approximation $e^{-L} \approx (1 - Ls/2)/(1 + Ls/2)$ for time delay terms. Let the range of k_1 be the same as above. The stabilizing range for other loop is computed by Wang et al.'s [12] method as

$$k_2 \in [-3.2188, 5.0708].$$

For comparison, the loop 2 has the equivalent transfer function as follows:

$$\overline{g}_2 = \left(g_{22} - \frac{k_1 g_{12} g_{21}}{1 + k_1 g_{11}} \right) k_2. \tag{6.27}$$

Consider the lower bound of k_2 first. We draw its Nyquist curve for two cases, ($k_1 = -2.0233$, $k_2 = -3.2188$) from the LMI method and ($k_1 = -2.0233$, $k_2 = -3.84$) from the proposed method. In Fig. 6.16, one sees that the former is not at critical stability while the latter is, indicating that the former is indeed conservative while the latter is exact. Consider now the upper bound of k_2. Figure 6.17 shows the Nyquist curve of (6.27) for ($k_1 = 1.1502, k_2 = 5.0708$) from LMI method and ($k_1 = 1.1502, k_2 = 3.6852$) from the proposed method, respectively. One sees

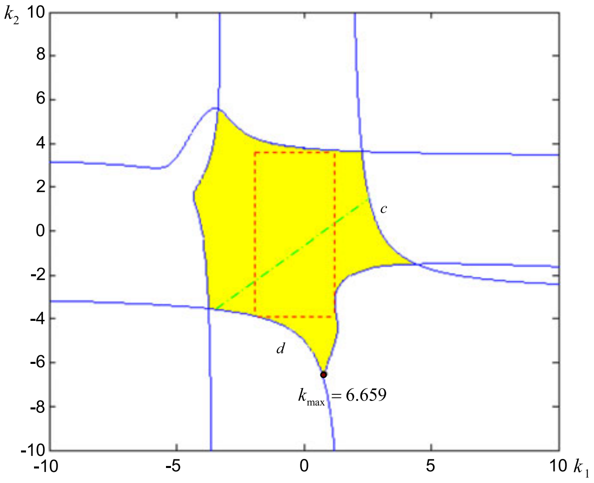


Fig. 6.14 Stabilizing region of (k_1, k_2)

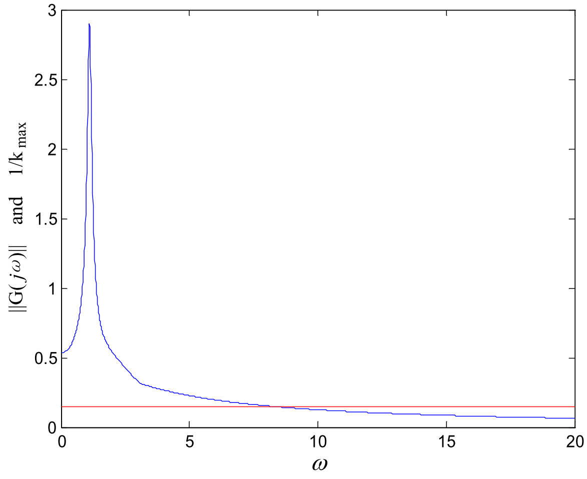


Fig. 6.15 The curves of $\|G(j\omega)\|_1$ and $\frac{1}{k_{\max}}$

that the former is unstable while the latter still produces a critically stable case. The upper bound of gain margin for loop 2 obtained by LMI method is far away from the true value, and this is because of the time delay approximation.

6.4 Conclusion

In this chapter, MIMO systems are introduced along with fundamental concepts such as transfer function matrices, poles, zeros, and feedback system stability. Next, a graphical design method for decentralized PI controller for the gain and phase margin specifications on each loop is presented for TITO processes. To deal with the interactions between the control loops, we treat the TITO process into two equiv-

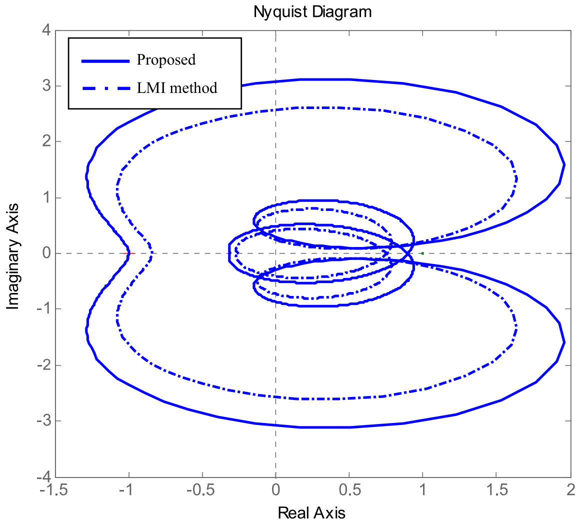


Fig. 6.16 Nyquist curve of $\overline{g}_2$

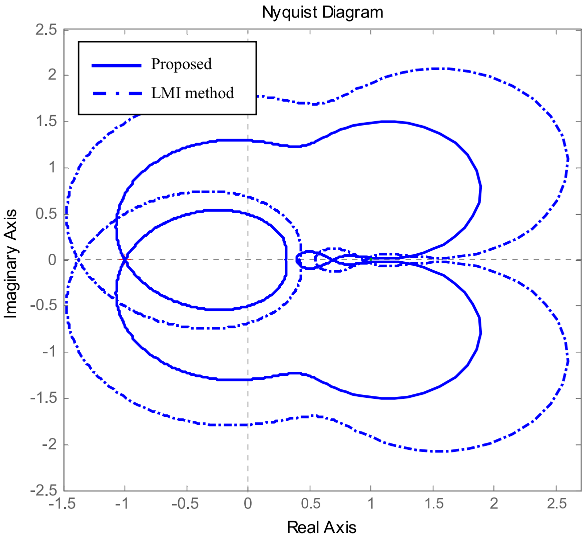


Fig. 6.17 Nyquist curve of $\overline{g}_2$

alent single loops with the unknown controller in the other loop as uncertainty at the beginning, with which an initial controller is designed. The controller in one loop is refined when the controller in the other loop is known. For SISO PI tuning, our early graphical design in Fung et al. [11] is used. A simulation example has demonstrated the effectiveness of the proposed method. Thirdly, the MIMO loop gain margins are defined with regard to simultaneous changes of multiple loop gains, and an effective algorithm is developed for their exact computation. The fundamental stability condition in the frequency domain is converted by the vector mapping method to a constrained optimization, with which the exact stability boundary is sought. The latter problem is then solved numerically by the Lagrange multiplier technique and Newton–Raphson iteration algorithm. The MIMO loop gain margins are found from exact stability boundary, and thus they are also exact, which eliminates the conservativeness of the existing method based on time domain linear matrix inequalities (LMI). The results are demonstrated for a time delay system.

Over the decades, researchers have developed rich modern control theories and designs for MIMO systems with sound mathematics. But their application in practical industrial systems seems limited except for a few methods such as MPC. The classical control theory and design for SISO systems often work in the frequency domain with a simple stability test, gain, and phases margins as performance specifications. They work well in industry because engineers find them easy to understand and convenient to apply and implement to meet practical needs. This is in contrast to the state space approach, where state variables may have no actual physical meaning, and they are often not all measurable in the practice. In line with this view, research and development of MIMO versions of SISO frequency response techniques for stability test, performance measures, and robustness would be of great theoretical and practical value. The work presented in this chapter reflects our latest efforts, and other projects are under progress.

References

1. Rosenbrock, H.H.: State-Space and Multivariable Theory. Nelson, London (1970)
2. Desoer, C.A., Shulman, J.D.: Zeros and poles of matrix transfer-functions and their dynamical interpretation. IEEE Trans. Circuits Syst. **CAS-21**, 3–8 (1974)
3. Desoer, C.A., Chan, W.S.: The feedback interconnection of linear time-invariant systems. J. Franklin Inst. **300**, 335–351 (1975)
4. Desoer, C., Wang, Y.T.: On the generalized Nyquist stability criterion. IEEE Trans. Autom. Control **25**, 187–196 (1980)
5. Jeng, J.C., Huang, H.P., Lin, F.Y.: Modified relay feedback approach for controller tuning based on assessment of gain and phase margins. Ind. Eng. Chem. Res. **45**, 4043–4051 (2006)
6. Wang, Y.G., Shao, H.H.: PID autotuner based on gain and phase margin specifications. Ind. Eng. Chem. Res. **45**, 3007–3012 (1999)
7. Wang, Q.G., Fung, H.W., Zhang, Y.: PID tuning with exact gain and phase margins. ISA Trans. **38**, 243–249 (1999)
8. Keel, L.H., Bhattacharyya, S.P.: Robust fragile, or optimal? IEEE Trans. Autom. Control **42**, 1098–1105 (1997)
9. Astrom, K.J., Hagglund, T.: Automatic tuning of simple regulators with specifications on phase and amplitude margins. Automatica **20**, 645–651 (1984)
10. Ho, W.K., Hang, C.C., Cao, L.S.: Tuning of PID controllers based on gain and phase margin specifications. Automatica **31**, 497–502 (1995)
11. Fung, H.W., Wang, Q.G., Lee, T.H.: PI tuning in terms of gain and phase margins. Automatica **34**, 1145–1149 (1998)
12. Wang, Q.G., Lin, C., Ye, Z., Wen, G.L., He, Y. Hang C.C.: A quasi-LMI approach to computing stabilizing parameter ranges of multi-loop PID controllers. J. Process Control **17**, 59–72 (2007)
13. Palmor, Z.J., Halevi, Y., Krasney, N.: Automatic tuning of decentralized PID controllers for TITO processes. Automatica **31**, 1001–1010 (1995)

14. Wang, Q.G., Zou, B., Lee, T.H., Bi, Q.: Auto-tuning of multivariable PID controllers from decentralized relay feedback. Automatica **33**, 319–330 (1997)
15. Wang, Q.G., Ye, Z., Cai, W.J., Hang, C.C.: PID Control for Multivariable Processes. Springer, Berlin (2008)
16. Ho, W.K., Lee, T.H., Gan, O.P.: Tuning of multiloop proportional-integral-derivative controllers based on gain and phase margin specifications. Ind. Eng. Chem. Res. **36**, 2231–2238 (1997)
17. Kookos, I.K.: Comments on "Tuning of multiloop proportional-integral-derivative controllers based on gain and phase margin specifications". Ind. Eng. Chem. Res. **37**, 1574 (1998)
18. Huang, H.P., Jeng, J.C., Chiang, C.H., Pan, W.: A direct method for multi-loop PI/PID controller design. J. Process Control **13**, 760–786 (2003)
19. Tavakoli, S., Griffin, I., Fleming, J.: Tuning of decentralized PI (PID) controllers for TITO processes. Control Eng. Pract. **14**, 1069–1080 (2006)
20. Zhang, Y., Wang, Q.G., Astrom, K.J.: Dominant pole placement for multi-loop control systems. Automatica **38**, 1213–1220 (2002)
21. Nie, Z.Y., Wang, Q.G., Wu, M., He, Y.: Tuning of multi-loop PI controllers based on gain and phase margin specifications. J. Process Control **21**, 1287–1295 (2011)
22. Majhi, S.: On-line PI control of stable processes. J. Process Control **15**, 859–867 (2005)
23. Astrom, K.J., Panagopoulos, H., Hagglund, T.: Design of PI controllers based on non-convex optimization. Automatica **34**, 585–601 (1998)
24. Bissell, C.: Control Engineering. Chapman and Hall, New York (1994)
25. Wood, R.K., Berry, M.W.: Terminal composition control of a binary distillation column. Chem. Eng. Sci. **28**, 1707–1717 (1973)
26. Luyben, W.L.: Simple method for tuning SISO controllers in multivariable systems. Ind. Eng. Chem. Process Des. Dev. **25**, 654–660 (1986)
27. Lee, T.T., Wang, F.Y., Newell, R.B.: Robust multivariable control of complex biological processes. J. Process Control **14**, 193–209 (2004)
28. Kristiansson, B., Lennartson, B.: Evaluation and simple tuning of PID controllers with high frequency robustness. J. Process Control **16**, 91–102 (2006)
29. Cvejn, J.: Sub-optimal PID controller settings for FOPDT systems with long dead time. J. Process Control **19**, 1486–1495 (2009)
30. Wang, Q.G.: Decoupling Control. Springer, New York (2003)
31. He, M.J., Cai, W.J., Wu, B.F., He, M.: Simple decentralized PID controller design method based on dynamic relative interaction analysis. Ind. Eng. Chem. Res. **44**, 8334–8344 (2005)
32. Doyle, J.: Analysis of feedback systems with structured uncertainty. IEE Proc. Part D. Control Theory Appl. **129**, 242–250 (1982)
33. Fan, M.K.H., Tits, A., Doyle, J.C.: Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. IEEE Trans. Autom. Control **36**, 25–38 (1991)
34. Braatz, R.P., Young, P.M., Doyle, J.C., Morari, M.: Computation complexity of calculation. IEEE Trans. Autom. Control **39**, 1000–1002 (1994)
35. Safonov, M.G.: Stability margins of diagonally perturbed multivariable feedback systems. IEE Proc. Part D. Control Theory Appl. **129**, 251–256 (1982)
36. Safonov, M.G., Athans, M.: A multiloop generalization of the circle criterion for stability margin analysis. IEEE Trans. Autom. Control **26**, 415–422 (1981)
37. De Gastion, R.R.E., Safonov, M.G.: Exact calculation of the multiloop stability margin. IEEE Trans. Autom. Control **33**, 156–171 (1988)
38. Bar-on, J.R., Jonckheere, E.A.: Multivariable gain margin. Int. J. Control **54**, 337–365 (1991)
39. Li, Y., Lee, E.B.: Stability robustness characterization and related issues for control systems design. Automatica **29**, 479–484 (1993)
40. Hong, Y., Yang, O.W.W.: Self-tuning multiloop PI rate controller for an MIMO AQM router with interval gain margin assignment. High Perform. Switch. Routing **12**, 401–405 (2005)
41. Ye, Z., Wang, Q.G., Hang, C.C.: Frequency domain approach to computing loop phase margins of multivariable systems. Ind. Eng. Chem. Res. **47**, C4418–C4424 (2008)
42. Galin, D.M.: Real matrices depending on parameters. Usp. Mat. Nauk **27**, 241–242 (1972)

43. Ballantine, C.S.: Numerical range of a matrix: some effective criteria. Linear Algebra Appl. **19**, 117–188 (1987)
44. Horn, R.A., Johnson, C.R.: Topics in Matrix Analysis. Cambridge University Press, Cambridge (1991)
45. Morari, M., Zafiriou, E.: Robust Process Control. Prentice Hall, Englewood Cliffs (1989)

Part II
Control Structures and Configurations for PID Control

Chapter 7
Feedforward Compensation for PID Control Loops

José Luis Guzmán, Tore Hägglund, and Antonio Visioli

7.1 Introduction

It is well known that the performance of a PID controller much depends, in addition to the tuning of the PID parameters [20], on the appropriate implementation of suitable additional functionalities, such as signal filtering, anti-windup, gain scheduling, and feedforward from set-point and measurable load disturbances [2, 3, 34]. Such features are nowadays easy to implement, due to the increased computational power available in Distributed Control Systems (DCS) as well as in single-station controllers. In order to preserve the ease of use of the overall controller, these functionalities should be transparent to the user and be intuitive.

This chapter focuses on the design and implementation of feedforward compensators, to be employed together with PID feedback controllers, for both set-point following and load disturbance rejection problems. The two problems are treated separately. Sect. 7.2 treats feedforward from the set-point, and Sect. 7.3 treats feedforward from load disturbances.

Regarding the set-point following, the typical approach of exploiting a feedforward control action is to implement a two-degree-of-freedom control scheme, to adopt a feedforward (linear) compensator [16]. The use of the well-known set-point weighting strategy [1] falls into this framework. The main disadvantage of this method is that the reduction of the overshoot is paid by a slower set-point response.

J.L. Guzmán (✉)
Dep. de Lenguajes y Computación, University of Almería, Almería, Spain
e-mail: joguzman@ual.es

T. Hägglund
Department of Automatic Control, Lund University, Lund, Sweden
e-mail: tore@control.lth.se

A. Visioli
Dipartimento di Ingegneria dell'Informazione, University of Brescia, Brescia, Italy
e-mail: antonio.visioli@ing.unibs.it

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_7, © Springer-Verlag London Limited 2012

To overcome this drawback, the use of an inverse model-based feedforward action can be employed [2] (see Sect. 7.2.1). Alternatively, the use of a variable set-point weight has been proposed [13, 32], but it has to be noted that in these cases, the regulation and servo control performances are no longer independent and the overall control scheme design is more complex.

In Sect. 7.2, after having presented the standard feedforward control scheme, different advanced methodologies, which exploit the fact that a process output transition is required instead of tracking a general reference signal, are presented. In particular, the design of a causal feedforward action and of a noncausal feedforward action are considered. In the first case, a (nonlinear) two-state control law is described. In the second case, to be employed when desired process output transitions are known in advance, strategies based on input–output inversion are explained both in the continuous-time and discrete-time frameworks.

Feedforward from measurable load disturbances provides a possibility to make control actions before any disturbance response has occurred in the process output. The typical design approach composes the compensator as the dynamics between the load disturbance and the process output divided by the dynamics between the control signal and the process output, with reversed sign. However, this ideal compensator is seldom realizable. The compensator may be noncausal, it may be unstable, it may have infinite high-frequency gain because of derivative action, and it may require a more complicated structure than what is available. These facts make the design problem nontrivial, and there is a need for design strategies and tuning rules.

There are a few design methods in the literature for feedforward compensator design for load disturbances. In [29], a design procedure for a lead-lag compensator was proposed, where the static gain of the compensator is calculated from pure static models, and the time constants of the lead-lag filter are then determined in order to reach $\mathrm{IE} = 0$ with minimized IAE. In [28], a design procedure where the feedforward gain is determined in the same way as in [29] is presented. A manual tuning procedure is then suggested to tune the time constants of the lead-lag filter. A similar approach to that in [28] was presented in [8], where a tuning procedure based on a training film from Foxboro, produced in 1978, was employed. Nevertheless, all these design methods are based on open-loop design, i.e., the feedback controller is not taken into account when the feedforward compensator is designed. This drawback was noticed first in [6], and later, in [14], a design method was presented where the feedback controller was taken into account when designing the feedforward compensator. The proposed design method is based on minimizing the norm of the transfer function between the disturbance and the process output, where the compensator is obtained after repeated solutions of least-square problems, but no simple and straightforward rules were presented.

Therefore, there is a lack of simple tuning rules for the design of feedforward compensators related to load disturbance rejection taking the feedback controller into account. To face this issue, a new simple feedforward control design method that takes the feedback controller into account was presented in [11], where the parameters of the feedforward compensator are calculated directly from the process models and the feedback controller parameters. In Sect. 7.3, the load rejection problem is first introduced. Afterwards, the classical open-loop design problem is summarized, and its main drawbacks are demonstrated. Then, the new design method that takes the feedback controller into account is described. The goal of the design is to obtain a load disturbance response without overshoot that has a minimum IAE value. Furthermore, restrictions on the high-frequency gain of the compensator are considered.

7.2 Feedforward Control for Set-Point Following

In the following subsections, different design methodologies for the improvement of the set-point following performance of a PID control scheme are presented. The addressed problem is to achieve a transition of the process output y from the value y_0 to the value y_1 in a predefined time interval of duration τ. Hereafter, for the sake of clarity and without loss of generality, it is assumed that $y_0 = 0$ and $y_1 > 0$.

7.2.1 Standard Approach

The standard methodology for the implementation of a feedforward action for the improvement of set-point following task (note that this control scheme can be effectively used to track any reference signal $r(t)$) is that shown in Fig. 7.1, where $M(s)$ is a reference model that gives the desired response of a set-point change, and $G(s)$ is chosen as

$$G(s) = \frac{M(s)}{\tilde{P}(s)}, \tag{7.1}$$

where $\tilde{P}(s)$ is the minimum-phase part of the process transfer function $P(s)$. Note that this is actually a general scheme and can be implemented with any feedback controller C, although here the adoption of a PID controller is assumed.

Obviously, the effectiveness of feedforward control heavily depends on the accuracy of the estimated process model (see the remarkable result presented in [9]). In any case, even if a perfect model is available, the design of $M(s)$ is a crucial issue, as it represents the desired performance. It has to contain the nonminimum-phase (i.e., the noninvertible) part of $P(s)$, and also it should take into account actuator limits (see Sect. 7.2.5).

7.2.2 Two-State Time-Optimal Feedforward Control

A feedforward technique control inspired by the bang-bang control strategy [18] can be employed effectively in order to fully exploit the capabilities of the actuator

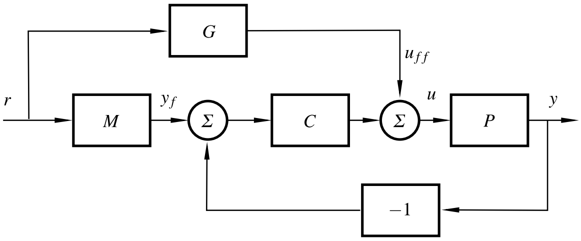


Fig. 7.1 Block diagram for the standard implementation of feedforward action for set-point following task

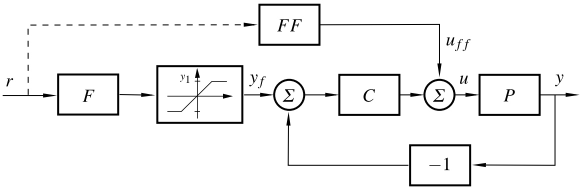


Fig. 7.2 Block diagram of the PID plus two-state feedforward action control scheme

to achieve a set-point transition [37, 38]. In particular, in [33], the control scheme shown in Fig. 7.2 has been proposed. Experimental results for a pharmaceutical plant are shown in [36]. The (self-regulating) process is described by a first-order-plus-dead-time (FOPDT) model, i.e.,

$$P(s) = \frac{K}{Ts+1}e^{-Ls}, \tag{7.2}$$

where K is the process gain, T is the time constant, and L is the dead-time. Based on this model, the output u_{ff} of the feedforward block FF is defined as follows:

$$u_{\mathrm{ff}}(t) = \begin{cases} \bar{u}_{\mathrm{ff}} & \text{if } t < \tau, \\ \frac{y_1}{K} & \text{if } t \geq \tau, \end{cases} \tag{7.3}$$

where the value of $\bar{u}_{\mathrm{ff}}$ is determined, after trivial calculations, in such a way that the process output y (which is necessarily zero until time $t = L$) is y_1 at time $t = \tau + L$. This produces the result

$$\bar{u}_{\mathrm{ff}} = \frac{y_1/K}{1 - e^{-\tau/T}}. \tag{7.4}$$

In this way, if the process is described perfectly by model (7.2), an output transition in the time interval $[L, \tau + L]$ occurs. Then, at time $t = \tau + L$, the output settles at value y_1 thanks to the constant value assumed by $u_{\mathrm{ff}}(t)$ for $t \geq \tau$. Formally, it is

$$y(t) = \begin{cases} 0 & \text{if } t < L, \\ \bar{u}_{\mathrm{ff}}(1 - \exp(-t/T)) & \text{if } L < t < L + \tau, \\ y_1 & \text{if } t \geq L. \end{cases} \tag{7.5}$$

Then, a suitable reference signal y_f has to be applied to the closed-loop system. It is desired that y_f be equal to the desired process output (7.5) that would be obtained in the case where the process is modeled perfectly by expression (7.2). Thus, the step reference signal r of amplitude y_1 has to be filtered by the system

$$F(s) = \frac{K\bar{u}_{\mathrm{ff}}}{y_1} \cdot \frac{1}{Ts+1} e^{-Ls} \tag{7.6}$$

and then saturated at the level y_1.

The overall control scheme design involves the selection of the transition time τ and of the PID parameters. The choice of a sensible value of τ can be made by the user either directly or through a (possibly) more intuitive reasoning. For example, the user might select a ratio between the bandwidth of the open-loop system and that of the closed-loop one, from which the value of τ can be determined easily. Obviously, decreasing the value of τ means that the value of $\bar{u}_{\mathrm{ff}}$ (and therefore of the overall manipulated variable) increases, and too low values of τ might imply that the determined control variable cannot be applied due to the saturation of the actuator. Thus, alternatively, the user might first select the value of $\bar{u}_{\mathrm{ff}}$ depending on the desired control effort (defined typically as a percentage of the maximum limit of the manipulated variable) and determine consequently the value of τ. In this way, the potentiality of the actuator can be fully exploited, and the problems associated with the use of the standard control scheme of Fig. 7.1 are avoided. In any case, the design parameter τ has a clear physical meaning, as it handles the trade-off between performance, robustness, and control activity [15, 19]. Indeed, it has the same role of the time constant of the reference model $M(s)$ in the classic technique. It can be therefore exploited to satisfy the specific requirements of a given application.

The tuning of the PID controller should take the robustness issue into account, since the feedforward action is based on a simple FOPDT model of the plant and the compensation of the (unavoidable) modeling errors is left to the feedback control law. In this respect, it is very useful to consider the analysis made in [37], where it is shown that the deviations due to the modeling errors between the desired and the actual output can be treated as the effect of a load disturbance. Thus, it is sensible to tune the PID controller by taking into account its load disturbance rejection performance.

As an illustrative example, consider the FOPDT process

$$P(s) = \frac{1}{10s+1} e^{-4s} \tag{7.7}$$

and the PID controller

$$C(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right) \tag{7.8}$$

whose parameters are selected according to the Ziegler–Nichols tuning rules, namely, $K_p = 3$, $T_i = 8$, and $T_d = 2$. If $\tau = 10$ is selected, it results (for $y_1 = 1$) in $\bar{u}_{\mathrm{ff}} = 1.58$. The closed-loop step response is shown as a solid line in Fig. 7.3, where it is compared with the one obtained by using the standard method described in Sect. 7.2.1 (dashed line). In this case, $M(s) = (10s+1)/(2s+1)$ has been cho-

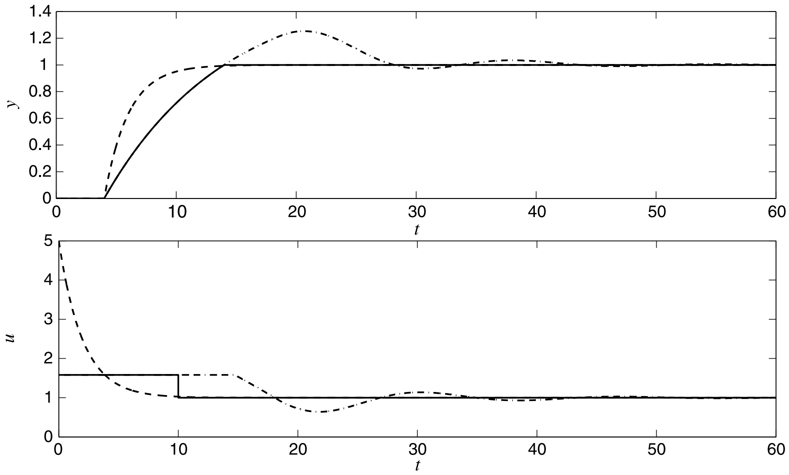


Fig. 7.3 Simulation results for the two-state and standard feedforward action control schemes. *Solid line*: two-state feedforward control. *Dashed line*: standard feedforward control with no saturation. *Dash-dotted line*: standard feedforward control with saturation

sen in order to obtain a similar rise time. It appears that a much higher control effort is required by the standard method to obtain the same performance (note that in both cases, the output of the PID controller is zero because there is no model uncertainty). Indeed, if the control variable is saturated at $\bar{u}_{\text{ff}} = 1.58$, the closed-loop step response obtained by using the control scheme of Fig. 7.1 presents a much larger overshoot and settling time (dash-dotted line in Fig. 7.3).

It is worth noting at this point that an extension of the method to multi-input-multi-output (MIMO) systems has been proposed in [27].

7.2.3 Noncausal Feedforward Action: Continuous-Time Case

When the process output transition is known in advance, a noncausal feedforward action determined can be used by applying a stable input–output inversion procedure [24]. The approach consists in selecting a desired output function that meets the control requirements and then determining, by inverting the system dynamics, the input function that causes that selected output signal. Thus, a suitable command signal is applied to the closed-loop control system, instead of the typical step signal, in order to achieve a high performance (i.e., low rise time and low overshoot at the same time) when the process output is required to assume a new value. The related scheme is that shown in Fig. 7.4.

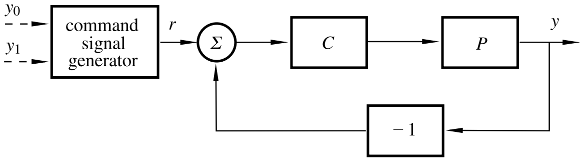


Fig. 7.4 Control scheme based on non-causal feedforward action

7.2.3.1 Design Methodology

As a first step of the design methodology, the process to be controlled (assumed to be self-regulating) is modeled as a FOPDT transfer function, where the dead-time term is approximated by means of a second-order Padè approximation, i.e.,

$$P(s) = \frac{K}{Ts+1}e^{-Ls} \cong \frac{K}{Ts+1} \cdot \frac{1 - Ls/6 + L^2s^2/12}{1 + Ls/6 + L^2s^2/12}. \tag{7.9}$$

Note that if the process is non-self-regulating, it can be modeled as an integrator-plus-dead-time (IPDT) transfer function, i.e.,

$$P(s) = \frac{K}{s} \cdot \frac{1 - Ls/6 + L^2s^2/12}{1 + Ls/6 + L^2s^2/12}. \tag{7.10}$$

Then the methodology is basically the same for the FOPDT case, and details for this case are omitted hereafter.

An output filtered PID controller in ideal form is employed as the feedback controller

$$C(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)\frac{1}{T_f s + 1}, \tag{7.11}$$

where the tuning of the parameters can be done according to any of the many methods proposed in the literature [20] or even by a trial-and-error procedure (note that, in any case, as the purpose of the overall procedure is the attainment of a high performance in the set-point following task, independently of the controller gains, it is sensible to select the PID parameters aiming only at obtaining a good load rejection performance).

At this point, a desired output function $y_d(t)$ that defines the transition from the set-point value $y_0 = 0$ to y_1 (to be performed in the time interval $[0, \tau]$) has to be selected. A sensible choice is to adopt a so-called "transition" polynomial [22], i.e., a polynomial function that is parameterized by the transition time τ and that satisfies boundary conditions at the endpoints of interval $[0, \tau]$. In particular, a third-order polynomial is selected in order to obtain a continuous command input function:

$$y_d(t) = y_1\left(-\frac{2}{\tau^3}t^3 + \frac{3}{\tau^2}t^2\right), \quad t \in [0, \tau]. \tag{7.12}$$

Outside the interval $[0, \tau]$, the function $y(t)$ is equal to 0 for $t < 0$ and to y_1 for $t > \tau$.

Once the closed-loop system is designed and the desired output function is selected, the problem of finding the command signal $r(t)$ that provides the desired output function has to be solved. This can be done by applying a suitable stable input–output inversion procedure (see [24] for details). Eventually, as the closed-loop transfer function (denoted as $H(s)$) is nonminimum-phase (and therefore a standard inversion of the transfer function would yield an unbounded command signal), a (bounded) command function defined over the interval $(-\infty, +\infty)$ results. It is therefore necessary to adopt a truncated function $r_a(t)$, resulting in an approximate generation of the desired output $y_d(t)$. In particular, a preactuation time t_s and a postactuation time t_f can be selected so that $r_a(t) = 0$ for $t < t_s$ and $r_a(t) = y_1$ for $t > t_f$. By taking into account that the preactuation and postactuation inputs (i.e., the input defined for $t < 0$ and $t > \tau$, respectively) converge exponentially to zero as time $t \to -\infty$ and to y_1 as time $t \to +\infty$, an arbitrarily precise approximation can be accomplished [23]. Alternatively, in order to simplify the computation, the method suggested in [21] can be adopted. It consists in selecting

$$t_s = -\frac{10}{D_{\text{rhp}}}, \qquad t_f = \tau + \frac{10}{D_{\text{lhp}}}, \tag{7.13}$$

where D_{rhp} and D_{lhp} are the minimum distances of the right and left half-plane zeros, respectively, from the imaginary axis of the complex plane. Note that the preactuation time depends only on the (apparent) dead-time of the process, as this determines the unstable zeros of the closed-loop systems by means of the Padè approximation. Indeed, by taking into account (7.13), it is $t_s = -10L/3$. Conversely, the postactuation time depends on the tuning of the PID parameters because the stable zeros of the closed-loop systems are those of the controller.

7.2.3.2 Discussion

The stable input–output inversion procedure can be performed by means of a symbolic computation, i.e., a closed-form expression of the command input function $r(t)$ results. Indeed, the actual command signal to be applied for a given plant and a given controller is determined by substituting the actual value of the parameters into the resulting closed-form expression, and this actually motivates its strong appeal in the context of PID control. Indeed, the overall procedure can be easily made transparent to the user. Further, being based on a general methodology [23], where $H(s)$ can be any rational transfer function of any (stable) system (provided that there are not purely imaginary zeros), the proposed approach can be straightforwardly applied also to PI, P, and PD controls and can be extended also to high-order processes. Thus, a more accurate model of the process, if available, can be fully exploited. However, in this case, the inversion procedure has to be performed.

Note also that, once the PID controller has been tuned, the only free design parameter is the transition time τ. Its role is basically the same of the transition time in the causal nonlinear feedforward method described in Sect. 7.2.2, namely, it allows one to handle the trade-off between performance, robustness, and control activity. It

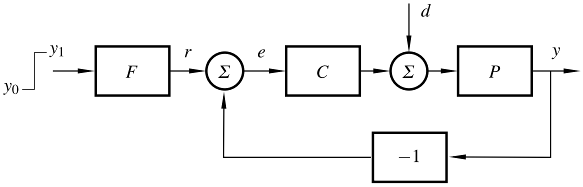


Fig. 7.5 Two-degree-of-freedom control scheme based on noncausal feedforward action

can be selected therefore by applying an analogous reasoning. However, by exploiting the closed-form expression of the control variable that can be easily derived, the transition time can be also determined by solving an optimization problem where its value has to be minimized subject to actuator constraints.

7.2.3.3 Implementation Issues

The command input in the interval $[t_s, t_f]$ is actually composed of three terms (the first one defined in the time interval $[t_s, 0]$, the second one in the time interval $[0, \tau]$, and the final one in the time interval $[\tau, t_f]$), and its expression, depending on the time variable t, can be difficult to implement with standard industrial hardware/software components [24]. For a simpler implementation of the method, it is convenient to obtain the command input as a step response, according to the two-degree-of-freedom control scheme of Fig. 7.5 [5]. Therein, the signal to be applied to the closed-loop system is obtained as a step response of a filter $F(s)$. For this purpose, it is necessary first of all to shift the time axis by substituting $t = \bar{t} - t_s$ and by taking $\bar{t}$ as the new time variable. Then, the expression of the filter $F(s)$ can be obtained by applying the Laplace transform operator to $r_a(\bar{t}; \tau)$:

$$R_a(s;\tau) = \mathcal{L}\big[r_a(\bar{t};\tau)\big] \tag{7.14}$$

and by imposing that

$$R_a(s;\tau) = \frac{1}{s}F(s;\tau). \tag{7.15}$$

Thus, we simply obtain

$$F(s;\tau) = sR_a(s;\tau). \tag{7.16}$$

By performing the required (symbolic) computations and by substituting backwards $\bar{t} = t + t_s$ we therefore have that the command signal $r_a(t;\tau)$ is obtained as the step response of the following filters, to be considered in different time intervals:

$$F(s;\tau) = \begin{cases} \frac{\beta_{2,1}s^2+\beta_{1,1}s+\beta_{0,1}}{\alpha_{2,1}s^2+\alpha_{1,1}s+\alpha_{0,1}} & \text{for } 0 \le t < -t_s, \\ \frac{\beta_{7,2}s^7+\cdots+\beta_{0,2}}{\alpha_{7,2}s^7+\cdots+\alpha_{3,2}s^3} & \text{for } -t_s \le t < -t_s+\tau, \\ \frac{\beta_{2,3}s^2+\beta_{1,3}s+\beta_{0,3}}{\alpha_{2,3}s^2+\alpha_{1,3}s+\alpha_{0,3}} & \text{for } -t_s+\tau \le t < -t_s+t_f, \\ H^{-1}(0) & \text{for } t \ge -t_s+t_f, \end{cases} \tag{7.17}$$

where $\beta_{2,1},\dots,\beta_{0,1},\beta_{7,2},\dots,\beta_{0,2},\beta_{2,3},\dots,\beta_{0,3}$ and $\alpha_{2,1},\dots,\alpha_{0,1},\alpha_{7,2},\dots,\alpha_{3,2},\alpha_{2,3},\dots,\alpha_{0,3}$ are suitable coefficients.

In other words, a step signal has to be applied at $t=0$ to the four different filters in (7.17), and then the command input to be applied to the closed-loop system is obtained by selecting the step responses of the filters according to the time intervals determined in (7.17). This strategy can be implemented easily in a Distributed Control System where a selector determines the required command input to be applied to the PID-based feedback control system by choosing between three transfer function blocks and a gain block according to the current time interval after the application of a set-point step signal.

A possible simplification for the implementation of the noncausal feedforward control strategy would be the use of a single transfer function instead of the four defined in (7.17). In general, the determined command input function can have a complex (nonmonotonic) shape, and it can be difficult to represent it as a step response of a single transfer function. However, there are cases for which this can be possible with a good accuracy. In particular, this happens when the command input function is smooth. Conditions for the occurrence of this situation can be found by considering the following proposition [22].

Proposition 7.1 *Given the closed-loop system $H(s)$ and the input function $r(\cdot;\tau)$, the following limit holds*:

$$\lim_{\tau\to+\infty}\left\| H(0)r(\cdot;\tau)-y_d(\cdot;\tau)\right\|_\infty=0, \tag{7.18}$$

where $\|f(\cdot)\| := \sup_{t\in\mathbb{R}}|f(t)|$ denotes the L_∞ norm of a real signal $f(t)$.

From a practical point of view this means that, when the transition time increases, the input function tends to be more similar to the desired output function. From another point of view, increasing the transition time τ yields a more robust system, namely, the obtained system output tends to be more similar to the desired output function. From these considerations it can be concluded that when the obtained system output has virtually no overshoot, then the corresponding system input is sufficiently smooth to be approximated as a step response of a single filter. By considering a large number of simulation results with different systems and different controllers [5], it turns out that an overshoot less than 5% is obtained in general when the transition time is selected as $\tau > 2L$. Thus, in this case the command input can be obtained as a step response of a single filter whose transfer function can be obtained by applying a standard least squares procedure [30] and by taking a step signal as input and the determined command signal as output (note that there is no noise). A fourth-order transfer function is generally sufficient to obtain a satisfactory performance.

7.2.3.4 Simulation Results

As an illustrative example, consider again process (7.7) with the PID controller tuned with the Ziegler–Nichols rules (see Sect. 7.2.2). The results obtained by se-

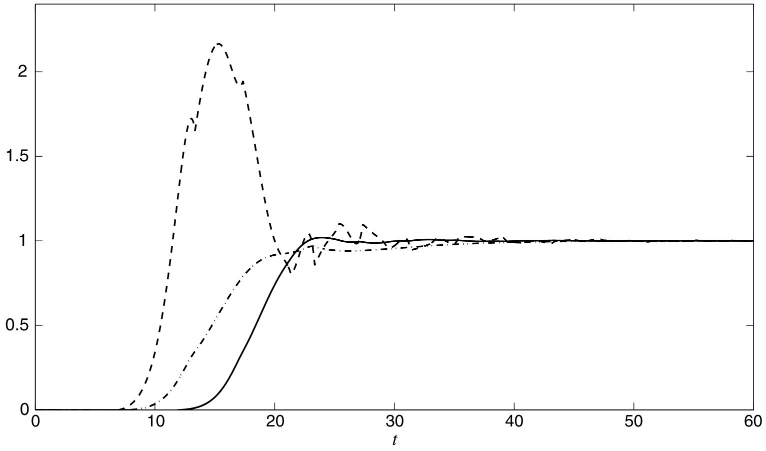


Fig. 7.6 Simulation results for the inversion-based control scheme. *Solid line*: process variable. *Dashed line*: control variable. *Dash-dotted line*: command input

lecting $\tau = 10$ are shown in Fig. 7.6, where the determined command input is plotted in addition to the process and control variables. The effect of the preactuation (which starts at time $t_s = -13.34$) can be evaluated. Note that in the plot the zero time has been conveniently shifted to t_s.

7.2.4 Noncausal Feedforward Action: Discrete-Time Case

A noncausal feedforward action can be designed also in a different context, namely, by inverting the dynamics of the closed-loop system after having identified it in the discrete-time framework by means of a step response [35].

Consider again the scheme shown in Fig. 7.4. As for the method described in Sect. 7.2.3, the aim is to find the command function $r(t)$ that produces a desired system output transition from $y_0 = 0$ to y_1, but here no a priori knowledge on the process model is assumed. For this purpose, sampled data are considered, where it is assumed that the sampling interval T has been chosen suitably by any standard technique [4].

An identification experiment can be easily performed by applying a unit step signal to the input of the closed-loop system. By denoting as $g_i := g(iT)$, $i = 1, \dots, N$, the corresponding sampled output values, a closed-loop system model can then be obtained by considering the truncated response ($t \in \{T, 2T, \dots, NT\}$):

$$y(t) = y_0 + g_{t/T} r(0) + \sum_{i=1}^{t/T-1} g_i \big[r(t - iT) - r\big(t - (i+1)T\big)\big], \tag{7.19}$$

where $r(t)$ is the system input, and the number N of parameters has to be taken sufficiently high in order to allow a sufficiently accurate description of the system, but not too high in order to minimize the computational effort of the control strategy. From a practical point of view, the sampling of the step response in order to obtain parameters g_i should stop when the process output remains close to its steady-state value for a sufficiently long time. At this point, it is convenient to write expression (7.19) in matrix form:

$$Y = GR, \tag{7.20}$$

where

$$Y = \begin{bmatrix} y(T) & y(2T) & y(3T) & \cdots & y(NT) \end{bmatrix}^{\mathrm{T}},$$

$$G = \begin{bmatrix} g_1 & 0 & 0 & \dots & 0 \\ -g_1 + g_2 & g_1 & 0 & \dots & 0 \\ -g_2 + g_3 & -g_1 + g_2 & g_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ -g_{N-1} + g_N & -g_{N-2} + g_{N-1} & -g_{N-3} + g_{N-2} & \dots & g_1 \end{bmatrix},$$

and

$$R = \begin{bmatrix} r(0) & r(T) & r(2T) & \cdots & r\big((N-1)T\big) \end{bmatrix}^{\mathrm{T}}.$$

Note that in many cases, as the model is obtained by evaluating a standard closed-loop step response, data taken from an output transition performed during routine process operations can be adopted. Obviously, it is important that the collected data be representative of a true step response (and therefore operations such as filtering and detrending might be necessary [17]) and if an unexpected load disturbance occurs during the transient response, they should not be adopted (see [12, 31] for methods to detect load disturbances).

The desired output function is chosen again as a fifth-order transition polynomial in order to obtain a convenient trade-off between the need to decrease the rise time and the need to decrease the control effort (note that the rise time decreases and the control effort increases when the order of the polynomial increases):

$$y_d(t;\tau) = \begin{cases} y_1(\frac{6}{\tau^5}t^5 - \frac{15}{\tau^4}t^4 + \frac{10}{\tau^3}t^3) & \text{if } 0 \le t \le \tau, \\ y_1 & \text{if } t > \tau. \end{cases} \tag{7.21}$$

Regarding the choice of the value of the transition time τ, the same considerations discussed in the continuous-time case can be applied also in this case.

The array Y_d can therefore be constructed based on the selected desired output function. Then, the corresponding closed-loop system input $r(t)$ that causes $y_d(t;\tau)$ can be easily determined by simply inverting the system using expression (7.20). Matrix G, in order to be invertible by a standard numeric algorithm, should be well conditioned, namely, there must not be a row (or a column) where all the elements are very small with respect to the elements of other rows (or columns). This happens when the process has a true dead-time or an apparent dead-time (i.e., when the process is of high order), which causes some of the first sampled output values g_i of the step response to be null or almost null. Thus, denote by k the number of the first rows of G in which all the elements are less than a selected threshold ε. Then, matrix $\hat{G}$ can be obtained by removing the first k rows and the last k columns from G. Subsequently, by evaluating $y_d(t;\tau)$ at the first $N-k$ sampling time intervals, the array $Y_d = [y_d(T;\tau)\ y_d(2T;\tau)\ \cdots\ y_d((N-k)T;\tau)]^{\mathrm{T}}$ can be easily constructed. The first $N-k$ values of the command reference input are then determined by applying the following expression:

$$\hat{R} = \big[r(T) \quad r(2T) \quad \cdots \quad r\big((N-k)T\big)\big]^{\mathrm{T}} = \hat{G}^{-1} Y_d. \tag{7.22}$$

In this way, the input function can be calculated by simply determining the inverse of a matrix.

Note that if the sampling interval T and the value of N have been selected appropriately, as well as the value of τ, then the last element of the array $\hat{R}$ actually corresponds to the steady-state value of the input, and therefore the value of $r((N-k)T)$ can be applied to the closed-loop system for $t > (N-k)T$. Note also that, since the first k rows and the last k columns have been removed from matrix G, the output function obtained is delayed by kT with respect to the desired one. Actually, the dead-time is removed in the model of the closed-loop system transfer function adopted in the dynamic inversion.

As an illustrative example, the same process (7.7) with the PID controller (7.8) tuned again with the Ziegler–Nichols rules has been considered. The initial set-point step response and that obtained after having applied the inversion procedure (with $\tau = 10$) are shown in Fig. 7.7. The obtained performance is, as expected, similar to that achieved in the continuous-time case.

7.2.5 Comparison

By looking at the presented examples and at many simulation and experimental results [24, 33–36], it can be deduced that the main advantages of the two-state feedforward control method is that it exploits full actuator capabilities in order to provide a minimum transition time and that there is no preactuation time interval. However, the inversion-based method provides a smoother control signal, and, most of all, it provides a similar performance in the presence of (very) different PID parameters (and also in the presence of modeling uncertainties).

7.2.6 Other Methodologies

Other feedforward control strategies to be employed with PID controllers have been proposed recently. In particular, the use of a Chebyshev technique has been devised in order to determine a feedforward control signal that provides a minimum-time rest-to-rest transition subject to constraints on both process input and output [10, 25, 26]. With the same aim, a linear programming approach is proposed in [7]. Details are not provided here for brevity. They can be found in the given references.

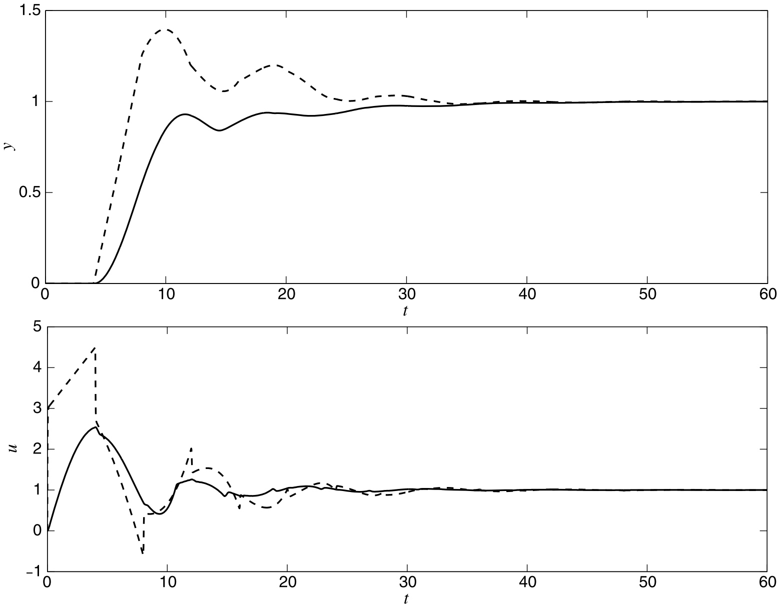


Fig. 7.7 Simulation results for the discrete-time inversion-based control scheme. *Dashed line*: initial step response. *Solid line*: response to the inversion-based command signal

7.3 Feedforward Control for Load Disturbance Rejection

In the following subsections, the problem of feedforward from load disturbances within a PID control scheme is analyzed. First, the basic feedforward design is described, and it is shown that the PID controller should be taken into account in the feedforward design process. Subsequently, new tuning rules, which have been recently presented in [11] for the design of feedforward compensators, are introduced where the closed-loop dynamics is considered during the design process. Several simulations are presented along the subsections to show the advantages of the proposed design rules.

7.3.1 The Feedforward Control Problem

The feedforward control problem for load disturbance rejection is illustrated by the block diagram in Fig. 7.8. The diagram consists of the basic feedback loop with feedback controller C, process P_1P_2, and the signals set-point r, control signal u, and process output y. A measurable load disturbance d influences the feedback loop according to the figure, with transfer function P_2P_3 between load d and process output y. The load disturbance is fed through a feedforward compensator G_{ff}, and the output from the compensator is added to the feedback control signal. The goal

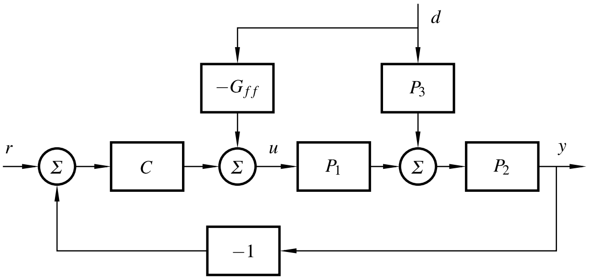


Fig. 7.8 Block diagram illustrating the feedforward control problem

is to design the feedforward compensator G_{ff} so that the effect of the disturbance d on the process output y is minimized.

In this chapter, the three process transfer functions are modeled as first-order systems with time delay, i.e.,

$$P_1(s) = \frac{K_1}{1+sT_1}e^{-sL_1}, \qquad P_2(s) = \frac{K_2}{1+sT_2}e^{-sL_2}, \qquad P_3(s) = \frac{K_3}{1+sT_3}e^{-sL_3}. \tag{7.23}$$

There are, of course, processes that are not well described by these simple transfer functions, but for process control applications, this structure is mostly good enough, and the structure has become the standard model in process control applications [3].

It is assumed that the feedback controller is a PI or PID controller with transfer function (7.8) where $T_d = 0$ in the case of PI control. Note that (7.8) is only the basic structure. As already mentioned, in the real implementation, features like filters, anti-windup, and limitations must be added.

In this chapter, we are looking for designing a feedforward compensator G_{ff} as a classical lead-lag compensator with delay:

$$G_{\text{ff}}(s) = K_{\text{ff}}\frac{1+sT_z}{1+sT_p}e^{-sL_{\text{ff}}}. \tag{7.24}$$

7.3.2 Open-Loop Design. Standard Approach

As mentioned in the introduction, most design methods for feedforward compensators neglect the feedback controller C and the effect that the feedback has on the disturbance rejection. This means that the open-loop transfer function between d and y is considered, which is given by

$$Y = P_2(P_3 - P_1G_{\text{ff}})D. \tag{7.25}$$

Perfect feedforward, which means that the effect of d is eliminated in y, is obtained when

$$G_{\text{ff}} = \frac{P_3}{P_1} = \frac{K_3}{K_1}\cdot\frac{1+sT_1}{1+sT_3}e^{-s(L_3-L_1)}, \tag{7.26}$$

which means that

$$K_{\mathrm{ff}} = \frac{K_3}{K_1}, \qquad T_z = T_1, \qquad T_p = T_3, \qquad L_{\mathrm{ff}} = L_3 - L_1. \tag{7.27}$$

When $L_3 < L_1$, the optimal parameters given by (7.27) give a noncausal feedforward compensator, since L_{ff} becomes negative. This means that perfect feedforward is not possible in this case and $L_{\mathrm{ff}} = 0$ has to be used. The negative delay can be approximated with a zero that can be added to T_z, a pole that can be added to T_p, or using a Padé approximation that alters both T_z and T_p. Such modifications will not influence the topic of this chapter and will therefore not be discussed further.

The following subsections present some examples to illustrate this tuning rule.

7.3.2.1 Open-Loop Design for $L_3 \geq L_1$

As a first example, the case where $L_3 \geq L_1$ is considered, which means that perfect feedforward is possible.

The process transfer functions are

$$P_1(s) = \frac{1}{1+2s}e^{-1.5s}, \qquad P_2(s) = \frac{1}{1+2s}e^{-1.5s}, \qquad P_3(s) = \frac{1}{1+5s}e^{-4s}. \tag{7.28}$$

The feedback controller C is a PI controller tuned using the AMIGO rule [3], which gives the parameters $K = 0.29$ and $T_i = 3.85$.

According to (7.27), the feedforward compensator becomes

$$G_{\mathrm{ff}}(s) = \frac{1+2s}{1+5s}e^{-2.5}. \tag{7.29}$$

Figure 7.9 shows the load disturbance response obtained when this feedforward design is used. The figure also shows the response obtained without feedforward compensation.

Since perfect feedforward is obtained when the lead-lag compensator with delay is used, there is no control error, and therefore the same responses are obtained when the feedback controller is manual and automatic mode.

7.3.2.2 Open-Loop Design for $L_3 < L_1$

We now consider an example where $L_3 < L_1$, which means that perfect compensation is not possible. The process transfer functions are:

$$P_1(s) = \frac{1}{1+4s}e^{-4s}, \qquad P_2(s) = \frac{1}{1+s}, \qquad P_3(s) = \frac{1}{1.5+s}e^{-3s}. \tag{7.30}$$

The controller is tuned using the AMIGO rule [3], which gives the parameters $K = 0.28$ and $T_i = 4.85$.

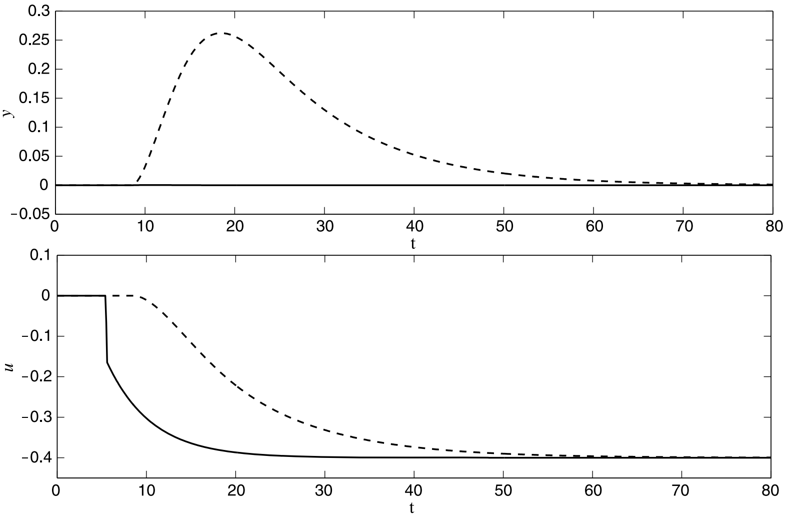


Fig. 7.9 Case $L_3 \geq L_1$, with step load disturbance of magnitude $d = 0.4$ at time $t = 5$ and $r = 0$. *Solid line*: response for a compensator using a lead-lag filter with delay. *Dashed line*: response for feedback without feedforward

Since $L_3 < L_1$, no delay should be included in the feedforward compensator, i.e., $L_{\rm ff} = 0$. This means that the resulting compensator is given by

$$G_{\rm ff}(s) = \frac{1+4s}{1+1.5s}. \tag{7.31}$$

Figure 7.10 shows the disturbance responses obtained using the feedforward compensator (7.31) with the open-loop and closed-loop responses.

Even though perfect feedforward is not possible in this case, the feedforward action gives a significant improvement of the disturbance response compared with pure feedback. However, when the loop is closed, the feedback controller deteriorates the control, and responses with overshoots are obtained. The settling times of the responses are also increased to almost twice the open-loop times. Comparing the control signals for the open-loop and closed-loop cases, it is obvious that the feedback controller adds a component to the process input that is not negligible, and therefore should be taken into account in the feedforward design.

On the other hand, Fig. 7.10 shows that the lead-lag compensator gives better performance than the pure feedback controller, but this is at the price of a significant peak in the control signal. The magnitude of this peak is

$$u_{\rm init} = K_{\rm ff}\frac{T_z}{T_p}d. \tag{7.32}$$

In this example, the load magnitude $d = 0.4$ gives the peak $u_{\rm init} = 1.06$. Peaks like this are common when lead-lag compensators are used for feedforward compensation. This problem will be discussed further in the next section.

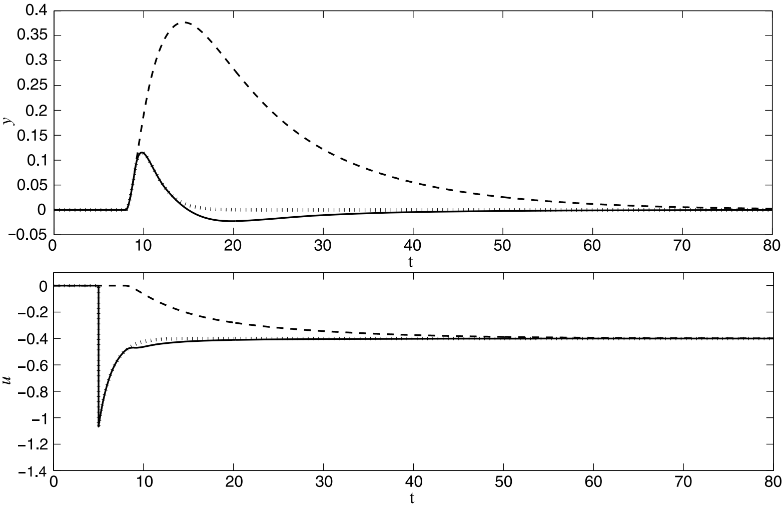


Fig. 7.10 Case $L_3 < L_1$, with step load disturbance of magnitude $d = 0.4$ at time $t = 5$ and $r = 0$. *Solid line*: response for a compensator using a lead-lag filter combined with feedback. *Dotted line*: response for a compensator using a lead-lag filter for the open-loop case. *Dashed line*: response for feedback without feedforward

7.3.2.3 Summary

This section has illustrated some drawbacks for the typical open-loop design of feedforward compensators. The open-loop design works well in cases where perfect feedforward is obtained, i.e., where the feedforward compensator manages to remove the disturbance response completely from the process output. In other cases, the feedback controller may deteriorate the response, and it is obvious that there is a need for tuning rules that take the feedback controller into account in the design.

The last example also illustrates that the feedforward action may give control signals that are considered too aggressive. The control signal response should therefore also be taken into account in the design.

These two aspects are treated in the next section.

7.3.3 Closed-Loop Design

As discussed above, the main goal when using feedforward is to eliminate the disturbance effect before affecting the process output. In the previous section, it was shown that perfect compensation is only possible for the case where $L_3 \geq L_1$. For $L_3 < L_1$, and when the compensator structure is not complex enough, no perfect compensation is possible, and there will be a disturbance response which will be fed back to the controller. This will result in a controller response that deteriorates the control when the open-loop feedforward design described in Sect. 7.3.2 is used.

In Fig. 7.10, it can be observed how a clear overshoot appears in the process output and that the settling times are increased when feedback is combined with feedforward.

For this reason, there is a need for feedforward tuning procedures. In this case, additional goals may be used in the feedforward design procedure. For instance, one goal may be to minimize the maximum deviation, and another goal may be to minimize the IAE value. Sometimes it is also desired to introduce restrictions in the design. It may, e.g., be undesirable to have an overshoot in the response, and there may be restrictions on the control signal variation.

Following these ideas, the next subsections present a tuning rule to determine the parameters of G_{ff} where the goal is to minimize the IAE value at step load disturbances without having any overshoot in the response. Furthermore, special attention will be given to reduce the initial peak of the resulting control signal.

The main goal is to propose a simple tuning rule where the parameters of the feedforward compensator are directly calculated based on the feedback controller (7.8) and the parameters of process models (7.23) [11].

7.3.3.1 Design of K_{ff} to Reduce the Overshoot

The first step in the tuning procedure is to reduce the feedforward gain so that the overshoot in the step response is removed. For that purpose, the feedback control action is taken into account to calculate the feedforward gain, K_{ff}.

Suppose that a step load disturbance of magnitude d results in a disturbance response with an integrated error equal to $\mathrm{IE} \cdot d$. Using a PI or PID controller, this disturbance causes the following static change of the control signal:

$$\Delta u = \frac{K}{T_i} \int e\, dt = \frac{K}{T_i} \mathrm{IE}\, d,$$

see [3].

The open-loop design rules for the feedforward (7.27) take the control signal to the new correct stationary level by choosing $K_{\mathrm{ff}} = K_3/K_1$. In the new rule, the goal is to take the control signal to the correct stationary level *minus* Δu in order to take the feedback control signal into account and reduce the overshoot. The gain is therefore reduced to

$$K_{\mathrm{ff}} = \frac{K_3}{K_1} - \frac{K}{T_i} \mathrm{IE}. \tag{7.33}$$

Notice that if the resulting feedforward compensator with reduced gain is used in open loop, it will give a steady-state error. However, there is no error when it is combined with feedback. This is another example to observe the feedback effect in the feedforward design.

What remains is to determine an estimate of the integrated error IE. The block diagram in Fig. 7.8 can be redrawn so that an equivalent system is obtained with an additive output residual disturbance

$$D_r = (P_2 P_3 - P_2 P_1 G_{\mathrm{ff}}) D. \tag{7.34}$$

This disturbance will be rejected by the feedback controller C. In the estimation of IE, we will assume that the feedback action can be neglected and that the IE value is what we get from the open-loop case. From Fig. 7.8 the process output in the open-loop case is given by

$$Y = P_2 P_3 D - P_2 P_1 G_{\mathrm{ff}} D. \tag{7.35}$$

The first term is the disturbance response, and the second term is the response caused by the feedforward action. The static gains in the two terms are $K_2 K_3$ and $K_1 K_2 K_{\mathrm{ff}}$, respectively. Assuming that $K_{\mathrm{ff}} = K_3/K_1$, they both have the same gain, namely

$$K_{ab} = K_2 K_3.$$

Suppose now that we can approximate the two terms in (7.35) with first-order responses. We then get

$$y(t) - r = \begin{cases} K_{ab}(1 - e^{-\frac{t}{T_a}})d, & 0 \le t \le L_b, \\ K_{ab}((1 - e^{-\frac{t}{T_a}}) - (1 - e^{-\frac{t-L_b}{T_b}}))d, & L_b < t, \end{cases} \tag{7.36}$$

where $t = 0$ corresponds to the time when the disturbance first appears in y, and where the delay and the time constants are approximated in the following way:

$$\begin{aligned} L_b &= \max(0, L_1 - L_3), \\ T_a &= T_2 + T_3, \\ T_b &= T_1 + T_2 + T_p - T_z. \end{aligned}$$

Here, the two transfer functions in (7.35) are approximated by first-order transfer functions. The time constants of these first-order transfer functions are obtained simply by adding the time constants in the denominators and subtracting time constants in the numerators of the original transfer functions ([3]).

Using (7.36), an estimate of the integral of the control error can now be determined:

$$\begin{aligned} \mathrm{IE} \cdot d &= \int_0^\infty \big(y(t) - r\big)\, dt \\ &= K_{ab} \int_0^{L_b} \big(1 - e^{-\frac{t}{T_a}}\big) d\, dt + K_{ab} \int_{L_b}^\infty \big(-e^{-\frac{t}{T_a}} + e^{-\frac{t-L_b}{T_b}}\big) d\, dt \\ &= K_{ab}\big[t + T_a e^{-\frac{t}{T_a}}\big]_0^{L_b} d + K_{ab}\big[T_a e^{-\frac{t}{T_a}} - T_b e^{-\frac{t-L_b}{T_b}}\big]_{L_b}^\infty d \\ &= K_{ab}\big(L_b + T_a e^{-\frac{L_b}{T_a}} - T_a - T_a e^{-\frac{L_b}{T_a}} + T_b\big) d \\ &= K_{ab}(L_b - T_a + T_b) d. \end{aligned}$$

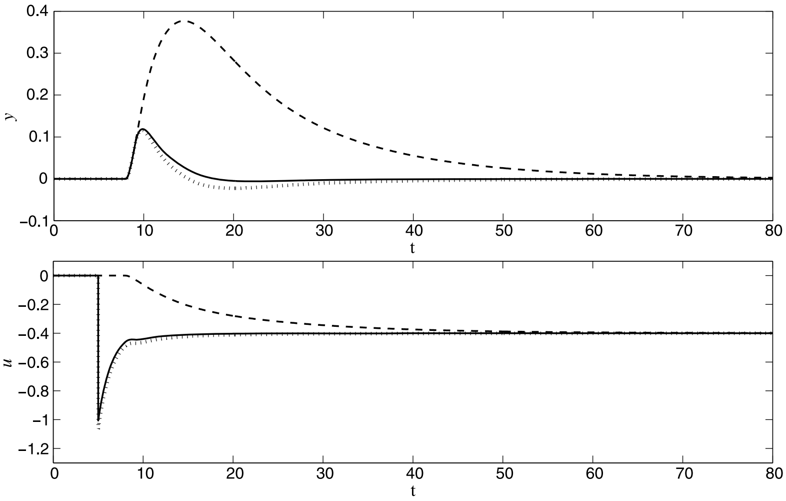


Fig. 7.11 Case $L_3 < L_1$ with load disturbances at time $t = 5$, $d = 0.4$, and $r = 0$, to analyze the gain reduction rule. *Solid line*: response for a lead-lag filter with the gain reduction rule ($K_{\mathrm{ff}} = 0.94$). *Dotted line*: response for a lead-lag filter without the gain reduction rule ($K_{\mathrm{ff}} = 1$). *Dashed line*: response for feedback without feedforward

Inserting the values of K_{ab}, L_b, T_a, and T_b gives

$$\mathrm{IE} = \begin{cases} K_2K_3(T_1 - T_3 + T_p - T_z), & L_3 \geq L_1, \\ K_2K_3(L_1 - L_3 + T_1 - T_3 + T_p - T_z), & L_3 < L_1. \end{cases} \tag{7.37}$$

Hence, knowing the parameters of the transfer functions in Fig. 7.8, IE can be easily estimated using (7.37), and the gain in the feedforward compensator can be reduced according to (7.33).

Figure 7.11 shows an example of the proposed rule. This figure compares the results of using (7.33) with those presented in Fig. 7.10, where the open-loop rule (7.27) was used. As can be observed from the results, the overshoot has been practically eliminated, where the feedforward gain K_{ff} has been reduced from 1 to 0.94. The settling times have also been significantly shortened because of the gain reduction. Notice also how the maximum deviation of the response has increased slightly because of the gain reduction. This issue will be improved in the following subsection by means of minimizing the IAE value of the load disturbance response.

7.3.3.2 Design of T_z and T_p to Reduce IAE

The second goal in the tuning rule proposed in this chapter consists in designing T_z and T_p of the lead-lag filter in order to minimize the IAE value of the disturbance response. From the open-loop design rule (7.27) it has been found advantageous to keep $T_z = T_1$ to cancel the pole in P_1. T_p can also be retained to $T_p = T_3$, but in some cases, the IAE value can be reduced significantly if a smaller value of T_p is chosen. Therefore, the proposed idea here is to keep $T_z = T_1$, canceling the pole in P_1 with the zero in G_{ff} and then modifying T_p as a filter time constant to increase the speed of the load disturbance response and reduce the associated IAE value.

The following derivations will give a rule to determine T_p for that purpose. In the derivations, it is assumed that $r = 0$ and $d = 1$. Notice that these assumptions will not influence the value of T_p.

Suppose that a step load disturbance gives a response where there may be an overshoot in the process output, i.e., where y crosses the set point. The IAE value then becomes

$$\mathrm{IAE} = \int_0^\infty \big|y(t)\big|\,dt = \int_0^{t_0} y(t)\,dt - \int_{t_0}^\infty y(t)\,dt, \tag{7.38}$$

where t_0 is the time when y crosses the set-point. From (7.36), t_0 is given by

$$\frac{t_0}{T_a} = \frac{t_0 - L_b}{T_b},$$

which gives

$$t_0 = \frac{T_a L_b}{T_a - T_b} = \frac{T_2 + T_3}{T_3 - T_p} L_b. \tag{7.39}$$

Equations (7.38) and (7.36) now give

$$\begin{aligned}
\mathrm{IAE} &= \int_0^{L_b} \big(1 - e^{-\frac{t}{T_a}}\big)\,dt + \int_{L_b}^{t_0} \big(-e^{-\frac{t}{T_a}} + e^{-\frac{t-L_b}{T_b}}\big)\,dt \\
&\quad - \int_{t_0}^{\infty} \big(-e^{-\frac{t}{T_a}} + e^{-\frac{t-L_b}{T_b}}\big)\,dt \\
&= \big[t + T_a e^{-\frac{t}{T_a}}\big]_0^{L_b} + \big[T_a e^{-\frac{t}{T_a}} - T_b e^{-\frac{t-L_b}{T_b}}\big]_{L_b}^{t_0} - \big[T_a e^{-\frac{t}{T_a}} - T_b e^{-\frac{t-L_b}{T_b}}\big]_{t_0}^{\infty} \\
&= L_b - T_a + T_b + 2T_a e^{-\frac{t_0}{T_a}} - 2T_b e^{-\frac{t_0-L_b}{T_b}} \\
&= L_b - T_a + T_b + 2T_a e^{-\frac{L_b}{T_a-T_b}} - 2T_b e^{-\frac{L_b}{T_a-T_b}} \\
&= L_b - \tau\big(1 - 2e^{-\frac{L_b}{\tau}}\big),
\end{aligned}$$

where $\tau = T_a - T_b$. To minimize IAE, we take the derivative with respect to τ:

$$\begin{aligned}
\frac{d}{d\tau}\mathrm{IAE} &= -1 + 2e^{-\frac{L_b}{\tau}} + 2\frac{L_b}{\tau} e^{-\frac{L_b}{\tau}} \\
&= -1 + 2(1+x)e^{-x} = 0,
\end{aligned} \tag{7.40}$$

Table 7.1 IAE comparison for the results of Fig. 7.12. The data show a clear reduction of the IAE value for the combination of the rules (7.33) and (7.42)

	No FF	Open-loop rule	K_{ff} reduction	K_{ff} and T_p reduction
IAE	17.39	1.89	1.43	0.63

where $x = L_b/\tau$. A numerical solution of this equation gives $x \approx 1.7$, which gives

$$T_p = T_b - T_a + T_3 = T_3 - \tau \approx T_3 - \frac{L_b}{1.7}. \tag{7.41}$$

Since T_p must be positive and since $L_b = \max(0, L_1 - L_3)$, we finally get the following rule for determining T_p:

$$T_p = \begin{cases} T_3, & L_1 - L_3 \le 0, \\ T_3 - \frac{L_1 - L_3}{1.7}, & 0 < L_1 - L_3 < 1.7T_3, \\ 0, & L_1 - L_3 > 1.7T_3. \end{cases} \tag{7.42}$$

The rule (7.42) gives $T_p = T_3$ when $L_3 \ge L_1$. This is of course correct, since this is the situation where perfect compensation is obtained. When $L_1 - L_3 > 1.7T_3$, the rule (7.42) suggests that $T_p = 0$. However, choosing $T_p = 0$ means that we get derivative action in the compensator and that a step change in the load gives an infinitely large pulse in the control signal. This problem is treated in the next subsection.

Notice that, when (7.42) is combined with the gain reduction rule (7.33), the T_p rule (7.42) must be applied first, so that the correct value of T_p is used in the IE calculation rule (7.37).

An example for the resulting rule is presented in Fig. 7.12. The figure shows the results for feedback without feedforward, with feedforward using the open-loop rule (7.27), with feedforward using only the gain reduction rule (7.33), and with feedforward combining the gain (7.33) and T_p (7.42) reduction rules. It can be observed how the response for the combination of (7.33) and (7.42) provides a faster response and a smaller error. Furthermore, this response obtains also the lower deviation, solving the problem described above when only the gain reduction rule is used.

Table 7.1 shows a comparison of the IAE values for the responses shown in Fig. 7.12. As it can be observed, there is a clear reduction of the IAE value for the case using (7.33) and (7.42) rules, corroborating the results presented in the figure.

Summarizing, the proposed design rule provides a straightforward way to improve the response to load disturbances reducing both the overshoot and the IAE value. However, as observed from Fig. 7.12, these improvements are reached at expense of obtaining a strong initial peak of the control signal, which sometimes may be undesirable. Hence, the next section describes some ideas to limit the high-frequency gain of the compensator and thus allowing the reduction of this initial peak.

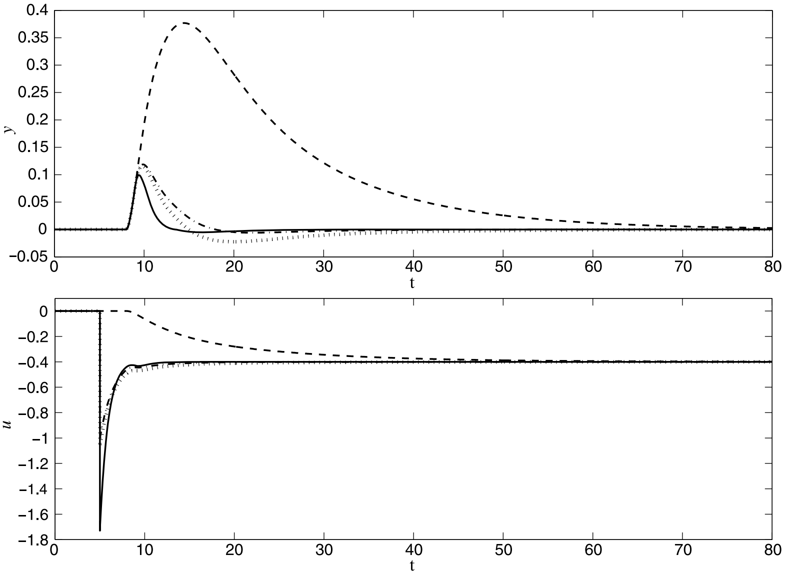


Fig. 7.12 Case $L_3 < L_1$ with load disturbances at time $t = 5$, $d = 0.4$, and $r = 0$, to analyze the T_p design rule. *Solid line*: response for lead-lag filter combining gain reduction (7.33) and T_p design rule (7.42) with $K_{\mathrm{ff}} = 0.977$ and $T_p = 0.904$. *Dash-dot line*: response for lead-lag filer for the case with only gain reduction rule (7.33) getting $K_{\mathrm{ff}} = 0.942$ and $T_p = 1.5$. *Dotted line*: response for lead-lag filter with the open-loop rule (7.27) getting $K_{\mathrm{ff}} = 1$ and $T_p = 1.5$. *Dashed line*: response for feedback without feedforward

7.3.3.3 Reduction of Control Signal Peak

The feedforward compensator has a low-frequency gain equal to K_{ff} and a high-frequency gain equal to $K_{\mathrm{ff}}T_z/T_p$. Feedforward design gives often a high-frequency gain that is considered too high when there are step changes in the load or when there is a high-frequency noise in the load disturbance. See, e.g., the peaks in the control signals in Figs. 7.10, 7.11, and 7.12.

There are many different ways to reduce the peak by modifying K_{ff}, T_z, or T_p. In [11], the following idea to modify only the T_p parameter based on a direct limitation of the high-frequency gain was proposed.

Suppose that an acceptable high-frequency gain is given by $K_{\mathrm{ff}}\kappa$, where κ is a design parameter. To guarantee that the high-frequency gain is below this limit, the time constant T_p must satisfy

$$T_p \geq \frac{T_z}{\kappa}. \tag{7.43}$$

This means that if a lower value of T_p is obtained, e.g., from the original design rule (7.27) or from the IAE optimization rule (7.42), it should be increased to meet the constraint (7.43). Remember that once T_p is modified, this change should be followed by a decrease of K_{ff} according to (7.37) and (7.33).

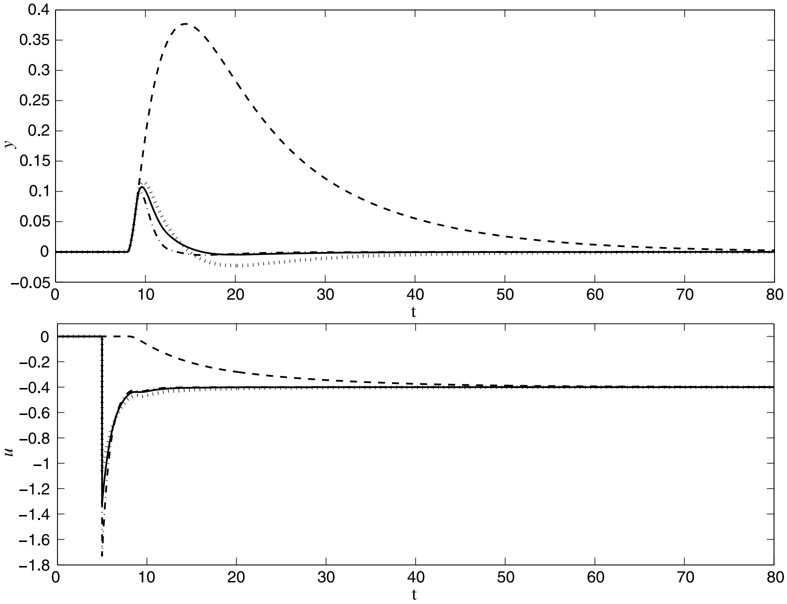


Fig. 7.13 Case $L_3 < L_1$ with load disturbances at time $t = 5$, $d = 0.4$, and $r = 0$, to analyze the reduction of the control signal peak. *Solid line*: response for lead-lag filter with the control signal reduction obtaining a modified $T_p = 1.12$ and $K_{\mathrm{ff}} = 0.96$ in order to avoid a peak lower than -1.3. *Dash-dotted line*: response for lead-lag filter combining gain reduction (7.33) and T_p design rule (7.42) with $K_{\mathrm{ff}} = 0.977$ and $T_p = 0.904$. *Dotted line*: response for lead-lag filter with the open-loop rule (7.27) getting $K_{\mathrm{ff}} = 1$ and $T_p = 1.5$. *Dashed line*: response for feedback without feedforward

The design parameter κ has a clear interpretation in the step load disturbance response of the control signal. When $\kappa < 1$, there is no peak in the control signal. When $\kappa = 1$, we have the static compensator, since $T_p = T_z$. When $\kappa > 1$, there is a peak in the control signal, and the value of κ gives the magnitude of the peak in relation to the static change of the control signal.

Figure 7.13 shows an example for the results presented in Fig. 7.12, but now trying to reduce the control signal peak to -1.3. The figure presents the results for feedback without feedforward, with feedforward using the open-loop rule (7.27) where $K_{\mathrm{ff}} = 1$ and $T_p = 1.5$, with feedforward combining the gain (7.33) and T_p (7.42) reduction rules obtaining $K_{\mathrm{ff}} = 0.977$ and $T_p = 0.904$, and the modification of T_p to reduce the control signal peak where T_p has been increased to 1.2 and the gain recalculated to $K_{\mathrm{ff}} = 0.96$.

The modification of T_p has been performed using the ideas described above. When using the gain (7.33) and T_p (7.42) reduction rules, it results in a value of $\kappa = 4.42$, which gives a high-frequency gain of $K_{\mathrm{ff}}\kappa = 4.32$. Hence, for the performed step change in the load disturbance, $d = 0.4$, a peak of -1.73 in the control signal is obtained. Therefore, κ is tuned to reduce the high-frequency gain in order to limit that peak to -1.3, obtaining a value of $\kappa = 3.33$, resulting in $T_p = 1.2$ according to (7.43).

From the results in Fig. 7.13, it is observed that the control signal peak has been properly reduced to -1.3. Notice also how the process output has been only slightly modified, where the IAE value has increased to 1, but being still smaller than those obtained from the other designs (see Table 7.1).

7.3.4 Guideline summary

This section summarizes the proposed design rule for tuning feedforward compensators of the following structure:

$$G_{\text{ff}} = K_{\text{ff}} \frac{1 + sT_z}{1 + sT_p} e^{-sL_{\text{ff}}}.$$

The different steps to design the compensator are:

1. Set $L_{\text{ff}} = \max(0, L_3 - L_1)$ and $T_z = T_1$.
2. Calculate T_p as

$$T_p = \begin{cases} T_3, & L_1 - L_3 \leq 0, \\ T_3 - \frac{L_1 - L_3}{1.7}, & 0 < L_1 - L_3 < 1.7T_3, \\ 0, & L_1 - L_3 > 1.7T_3, \end{cases}$$

3. Calculate the compensator gain, K_{ff}, as

$$K_{\text{ff}} = \frac{K_3}{K_1} - \frac{K}{T_i}\text{IE},$$

$$\text{IE} = \begin{cases} K_2 K_3 (T_1 - T_3 + T_p - T_z), & L_3 \geq L_1, \\ K_2 K_3 (L_1 - L_3 + T_1 - T_3 + T_p - T_z), & L_3 < L_1. \end{cases}$$

4. Analyze the high-frequency gain, $K_{\text{ff}}\kappa$, based on the design performed in the previous steps ($\kappa = T_z / T_p$). If the resulting high-frequency gain is acceptable, go to step 5. Otherwise, modify κ to reach the desirable high-frequency gain and change T_p as

$$T_p \geq \frac{T_z}{\kappa}.$$

   Go to step 3.
5. End of the design process.

7.4 Conclusions

Feedforward compensators require design methods and tuning rules in the same way as feedback controllers. This chapter has presented methodologies for the design of feedforward control actions to be employed together with feedback PID controllers, where both set-point following and load disturbance rejection have be considered.

Regarding the set-point following task, the design for a causal feedforward action and for a noncausal feedforward action have been presented. In the first case, a (nonlinear) two-state control law was used. In the second case, to be employed when desired process output transitions are known in advance, strategies based on input–output inversion are explained both in the continuous-time and discrete-time frameworks. Several advantages were observed for both methods. The two-state control method exploits full actuator capabilities in order to provide a minimum transition time and there is no preactuation time interval. The inversion-based method provides a smoother control signal and a similar performance of different PID parameters, also in the presence of modeling uncertainties, thus presenting a robust behavior.

Regarding the load disturbance rejection problem, most design methods presented so far are based on an open-loop approach, which means that the feedback loop is not taken into account in the design. This approach is relevant in those cases where perfect feedforward is obtained, but in other situations, the feedback controller should not be neglected in the design. This chapter has presented new simple tuning rules for design of feedforward compensators. The rules are based on simple first-order-plus-dead-time models of the process, and they take the PID controller settings into account. The starting point for the rules is the ideal open-loop tuning. The feedforward gain and the lag-time constant are then reduced with the goal to reach a response to step changes in the load disturbance without any overshoot and where the IAE value is minimized. Feedforward compensator design often results in compensators with a large high-frequency gain. Therefore, the tuning procedure ends with a procedure where the lag-time constant may be adjusted so that a desired high-frequency gain of the compensator is obtained. Robustness analysis for the proposed rule can be found in [11].

References

1. Araki, M.: Two degree-of-freedom PID controller. Syst. Control Inf. **42**, 18–25 (1988)
2. Åström, K.J., Hägglund, T.: PID Controllers: Theory, Design and Tuning. ISA Press, Research Triangle Park (1995)
3. Åström, K.J., Hägglund, T.: Advanced PID Control. ISA, Research Triangle Park (2005)
4. Åström, K.J., Wittenmark, B.: Computer-Controlled Systems—Theory and Design. Prentice Hall, Upper Saddle River (1997)
5. Beschi, M., Piazzi, A., Visioli, A.: On the practical implementation of a noncausal feedforward technique for PID control. In: Proceedings of the European Control Conference, Budapest, Hungary, pp. 1806–1811 (2009)
6. Brosilow, C., Joseph, B.: Techniques of Model-based Control. Prentice-Hall, New York (2002)
7. Consolini, L., Piazzi, A., Visioli, A.: Minimum-time feedforward control for industrial processes. In: Proceedings European Control Conference, Kos, Greece, pp. 5282–5287 (2007)
8. Coughanowr, D.R.: Process Systems, Analysis and Control. McGraw-Hill, New York (1991)
9. Devasia, S.: Should model-based inverse inputs be used as feedforward under plant uncertainties? IEEE Trans. Autom. Control **47**(11), 1865–1871 (2002)

10. Gervasio, P., Piccagli, S., Visioli, A.: On the practical implementation of feedforward control signals given in polynomial form. In: Proceedings IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Mallorca, Spain (2009)
11. Guzmán, J.L., Hägglund, T.: Simple tuning rules for feedforward compensators. J. Process Control **21**(1), 92–102 (2011)
12. Hägglund, T., Åström, K.J.: Supervision of adaptive control algorithms. Automatica **36**(2), 1171–1180 (2000)
13. Hang, C.-C., Cao, L.: Improvement of transient response by means of variable set point weighting. IEEE Trans. Ind. Electron. **43**, 477–484 (1996)
14. Isaksson, A., Molander, M., Modén, P.E., Matsko, T., Starr, K.: Low-order feedforward design optimizing the closed-loop response. Preprints, Control Systems, Vancouver, Canada (2008).
15. Kristiansson, B., Lennartson, B.: Robust and optimal tuning of PI and PID controllers. IEE Proc., Control Theory Appl. **149**(1), 17–25 (2001)
16. Kuo, B.C.: Automatic Control Systems. Prentice Hall, Englewood Cliffs (1995)
17. Leva, A., Cox, C., Ruano, A.: Hands-on PID autotuning: a guide to better utilisation. Technical report, IFAC Technical Brief (2001). Available at www.ifac-control.org
18. Lewis, F.L.: Optimal control. In: Levine, W.S. (ed.) The Control Handbook, pp. 759–778. CRC Press, Boca Raton (1996)
19. Morari, M., Zafiriou, E.: Robust Process Control. Prentice Hall, Englewood Cliffs (1989)
20. O'Dwyer, A.: Handbook of PI and PID Tuning Rules. Imperial College Press, London (2006)
21. Perez, H., Devasia, S.: Optimal output transitions for linear systems. Automatica **39**, 181–192 (2003)
22. Piazzi, A., Visioli, A.: Optimal noncausal set-point regulation of scalar systems. Automatica **37**(1), 121–127 (2001)
23. Piazzi, A., Visioli, A.: Using stable input-output inversion for minimum-time feedforward constrained regulation of scalar systems. Automatica **41**(2), 305–313 (2005)
24. Piazzi, A., Visioli, A.: A noncausal approach for PID control. J. Process Control **16**, 831–843 (2006)
25. Piccagli, S., Visioli, A.: Minimum-time feedforward plus PID control for mimo systems. In: Preprints 17th IFAC World Congress, Seoul, Korea, pp. 12917–12922 (2008)
26. Piccagli, S., Visioli, A.: Minimum-time feedforward technique for PID control. IET Control Theory Appl. **3**, 1341–1350 (2009)
27. Piccagli, S., Visioli, A.: An optimal feedforward control design for the set-point following of mimo processes. J. Process Control **19**, 978–984 (2009)
28. Seborg, D.E., Edgar, T.F., Mellichamp, D.A.: Process Dynamics and Control. Wiley, New York (1989)
29. Shinskey, F.: Process Control Systems. Application Design Adjustment. McGraw-Hill, New York (1996)
30. Sung, S.W., Lee, I.-B., Lee, B.-K.: On-line process identification and automatic tuning method for PID controllers. Chem. Eng. Sci. **53**, 1847–1859 (1998)
31. Veronesi, M., Visioli, A.: A technique for abrupt load disturbance detection in process control systems. In: Preprints 17th IFAC World Congress, Seoul, Korea, pp. 14900–14905 (2008)
32. Visioli, A.: Fuzzy logic based set-point weight tuning of PID controllers. IEEE Trans. Syst. Man Cybern., Part A, Syst. Hum. **29**, 587–592 (1999)
33. Visioli, A.: A new design for a PID plus feedforward controller. J. Process Control **14**, 455–461 (2004)
34. Visioli, A.: Practical PID Control. Springer, London (2006)
35. Visioli, A., Piazzi, A.: Improving set-point following performance of industrial controllers with a fast dynamic inversion algorithm. Ind. Eng. Chem. Res. **42**, 1357–1362 (2003)
36. Visioli, A., Ammannito, M., Caselli, M., Incardona, M.: Optimization of a water for injection control system for a pharmaceutical plant. In: Proceedings 8th European Workshop on Advanced Control and Diagnosis, Ferrara, Italy (2010)
37. Wallen, A.: Tools for autonomous process control. PhD thesis, Lund Institute of Technology, Lund, Sweden (2000)
38. Wallen, A., Åström, K.J.: Pulse-step control. In: Preprints of the 15th IFAC World Congress on Automatic Control, Barcelona, Spain (2002)

Chapter 8
Control Structures for Time Delay Systems

Somanath Majhi

8.1 Performances of a Control System

To analyze, synthesize, and design a control system, it is necessary to specify and measure the required performance of the control system. The system parameters are adjusted to provide the desired response based on the required performance of the control system. As control systems are inherently dynamic, their performance is often specified in terms of their time or frequency response. The performance of the system can be readily improved by inserting a controller in a suitable location within the structure of the system. Since the controller design is directly related to the desired performance of the control system, a brief description of the time and frequency performance measure is given below.

8.1.1 Time Domain Performance Measures

The time domain performance is normally obtained from the response of the process to a test signal. The standard test input signals are step, ramp, and pulse inputs. In this section, the most commonly used step input is considered for performance measure because it yields very useful information about the process dynamics. The transient response is the response that disappears with time and the steady-state response is that which exists as time $t \to \infty$. Figure 8.1 shows a typical step response of a closed-loop system. Note that the output is delayed by some time with respect to the input. The time domain performance measures of a closed-loop system are usually given in terms of maximum overshoot which occurs at peak time, settling time, rise time, steady-state error, and disturbance response. These six quantities give a direct measure of the behavior of a control system to a unit step input. In

S. Majhi (✉)
Indian Institute of Technology Guwahati, Guwahati 781039, India
e-mail: smajhi@iitg.ernet.in

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_8, © Springer-Verlag London Limited 2012

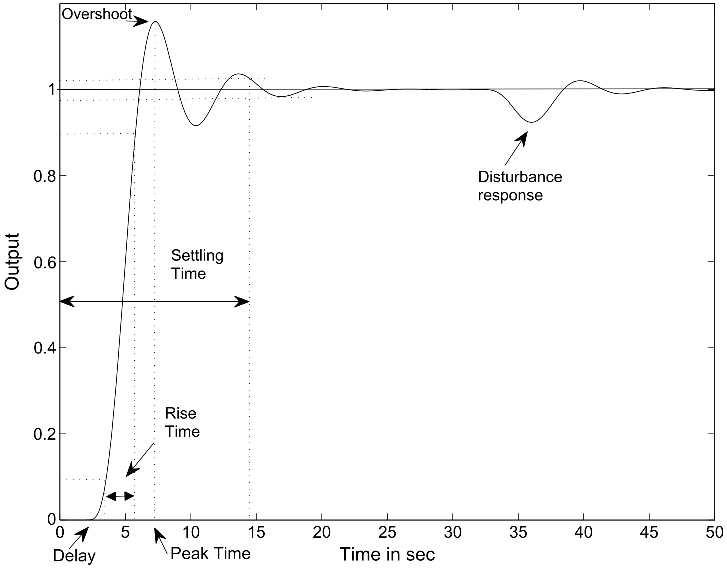


Fig. 8.1 A typical closed-loop output response to step setpoint and disturbance inputs

practice, the system output is usually required to follow the input as closely as possible, which ideally means zero steady-state error and small rise time, settling time, and percentage of overshoot.

8.1.2 Frequency Domain Performance Measures

A very practical and important alternative approach for the analysis and design of a system is the frequency response method [1]. The frequency response of a system indicates the magnitude and phase relationship between the sinusoidal input and the steady-state output of the process. Figure 8.2 illustrates a typical frequency response plot for increasing frequency ω. In the design of linear control system using frequency domain method, the performance of a system can be identified using the appropriate performance measures such as the phase margin (ϕ_m), gain margin (g_m), delay margin, resonant peak, and bandwidth. Another important characteristic of a frequency response measure of a system is the critical point, where the Nyquist curve intersects the point $-1 + j0$ in the negative real axis. This point yields information about the critical gain and the critical frequency of a closed-loop system exhibiting sustained oscillation.

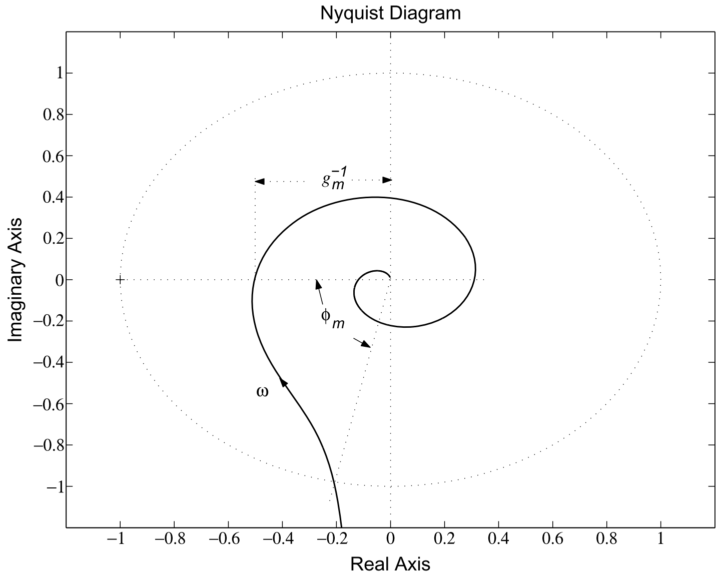


Fig. 8.2 Typical frequency response

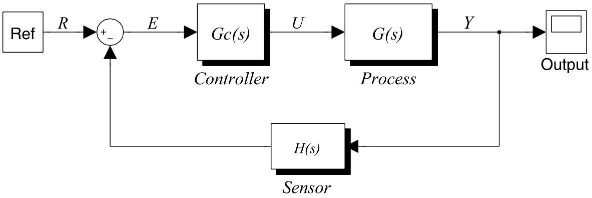


Fig. 8.3 Basic control system

8.2 Conventional Control Structures

The basic control system can be represented by the block diagram shown in Fig. 8.3, where U and Y represent the control signal and the controlled variable, respectively. The sensor, $H(s)$, is assumed infinitely fast and of unity gain. Often, there are limitations of classical controller acting on the error signal for controlling processes such as the unstable process $G(s)$ given in Example 8.3. Limitation of control system performance has resulted in several control configurations. The commonly used system configurations with controller compensation are described below briefly.

8.2.1 Series Compensation

Figure 8.3 shows the most commonly used system configuration where the controller is placed in series with the process in the forward path. The series compen-

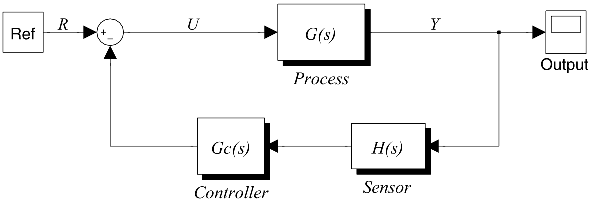


Fig. 8.4 Feedback compensation

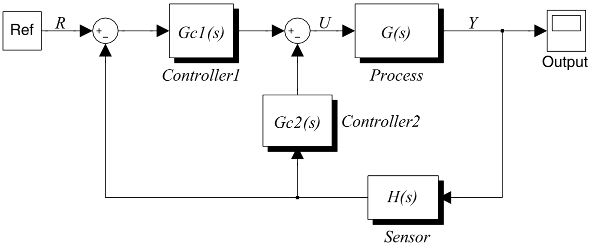


Fig. 8.5 Series-feedback compensation

sation generally use the standard PI, PID, phase lead, and lag type controllers in the loop.

8.2.2 Feedback Compensation

The controller is placed in the feedback path as shown in Fig. 8.4 in the feedback compensation. A phase lead type controller can improve the closed-loop performance.

8.2.3 Series-Feedback Compensation

Figure 8.5 shows the series-feedback configuration where the feedback controller is placed in the inner feedback path and the series controller is subjected to the error signal. PI-D, PID-P, and PI-PD controllers are the best examples for this configuration. Difficulties are often found for controlling plants with resonances, integrators, or unstable poles. This modified control structure can produce improved performances in such situations [2].

8.2.4 Series Compensation with Setpoint Filter

Block diagram reduction of Fig. 8.5 results in the setpoint-weighted compensation scheme shown in Fig. 8.6. However, the reference path controller $F(s)$ cannot influence the disturbance responses. The controllers can be of PI, PID, phase lead, and

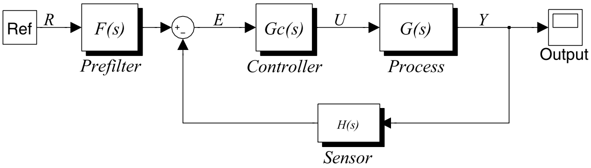


Fig. 8.6 Series compensation with setpoint filter

Table 8.1 PID structure and compensator values

Structure	Compensator/Controllers
PID	$G_c(s) = 0.144(1 + \frac{1}{0.372s} + \frac{5.613s}{0.561s+1})$
PI-D	$G_{c1}(s) = 0.144(1 + \frac{1}{0.595s})$, $G_{c2}(s) = \frac{0.61s}{0.061s+1}$
PI-PD	$G_{c1}(s) = 0.144(1 + \frac{1}{0.142s})$, $G_{c2}(s) = 0.922 + \frac{1.597s}{0.16s+1}$

lag types. The compensation schemes shown in Figs. 8.3 and 8.4 have one degree of freedom, and therefore, the performance criteria that can be realized are limited. For example, if a system is to be designed to achieve a certain amount of relative stability, it may have poor sensitivity to parameter variations. Again, if the roots of the characteristic equation are selected to provide a certain amount of relative damping, the maximum overshoot of step response may still be excessive, owing to the zeros of the closed-loop transfer function. The compensation schemes shown in Figs. 8.5 and 8.6 have two degrees of freedom. The benefit of using the improved compensation techniques is illustrated in the following examples.

Example 8.1 For a resonant plant transfer function $G(s) = \frac{e^{-0.1s}}{s^2+0.02s+1}$, constraining the proportional gain of the forward path controller to the same value for fair comparison, the PID controller for the series compensation and the PI-D and PI-PD controllers for the series-feedback compensation are designed by the integral squared time error (ISTE) optimization [2] and tabulated in Table 8.1. The unit step setpoint responses are shown in Fig. 8.7. As expected, the two-degree-of-freedom PI-PD controller gives satisfactory dynamic response in terms of rise time, overshoot, and other measure of transient response.

Example 8.2 Consider the integrating process transfer function $G(s) = \frac{e^{-0.2s}}{s(s+1)^3}$. The series-feedback compensators suggested in the literature [2] are the four-parameter PID-P and PI-PD controllers as tabulated in Table 8.2. A PID controller designed by the Ziegler–Nichols method [3] is also included in the table. The closed-loop responses for a unit step reference input and unit step load disturbance responses are shown in Figs. 8.8(a) and (b). The series-feedback compensation show satisfactory setpoint tracking and disturbance rejection compared to the series compensation.

Example 8.3 Consider the first-order unstable process transfer function $G(s) = \frac{4e^{-2s}}{4s-1}$. Either series or feedback compensation alone may not be sufficient to provide desirable time/frequency performance measures [4]. Figures 8.1 and 8.2 show

Fig. 8.7 Responses of (**a**) PI-PD, (**b**) PI-D and (**c**) PID control

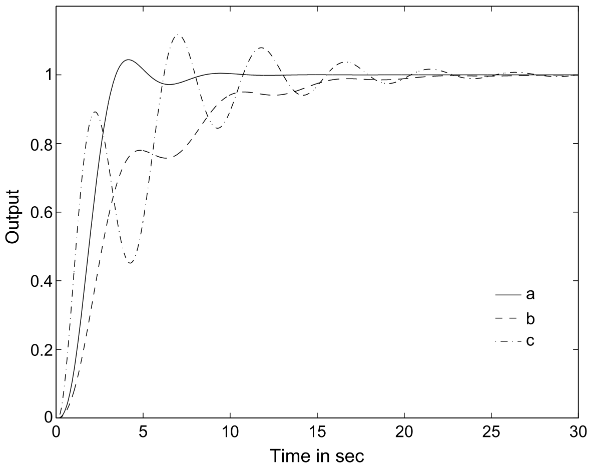
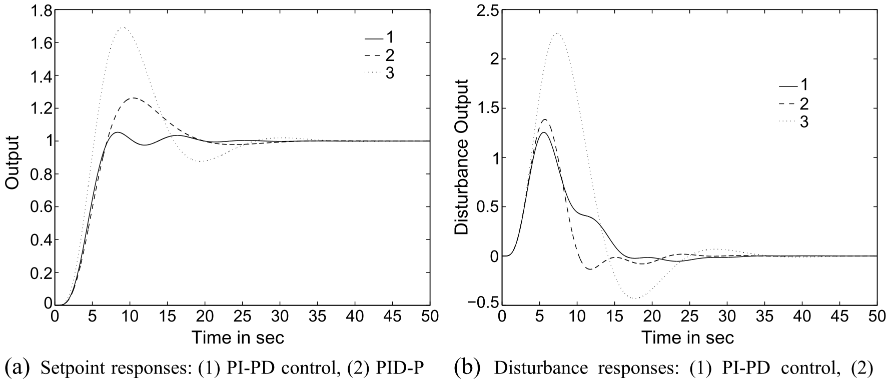


Table 8.2 PID structure and compensator values

Structure	Compensator/Controllers
PID	$G_c(s) = 0.44(1 + \frac{1}{3s} + 0.75s)$
PID-P	$G_{c1}(s) = 0.26(1 + \frac{1}{3.76s} + 2.95s)$, $G_{c2}(s) = 0.2$
PI-PD	$G_{c1}(s) = 0.26(1 + \frac{1}{2.13s})$, $G_{c2}(s) = 0.51 + \frac{1.14s}{0.114s+1}$



(a) Setpoint responses: (1) PI-PD control, (2) PID-P control and (3) PID control

(b) Disturbance responses: (1) PI-PD control, (2) PID-P control and (3) PID control

Fig. 8.8 Responses of Example 8.2

the time and frequency performance resulting from the series-feedback configuration with the controllers $G_{c1}(s) = 0.131(1 + \frac{1}{2s})$ and $G_{c2}(s) = 0.5(s + 1)$. The inner feedback loop enables placement of the open-loop poles in appropriate positions, thereby providing good control for open-loop system transfer functions having resonances, unstable, or integrating poles. To illustrate this point, let the delay term be approximated by $e^{-2s} = \frac{1-s}{1+s}$. Then, the inner loop controller

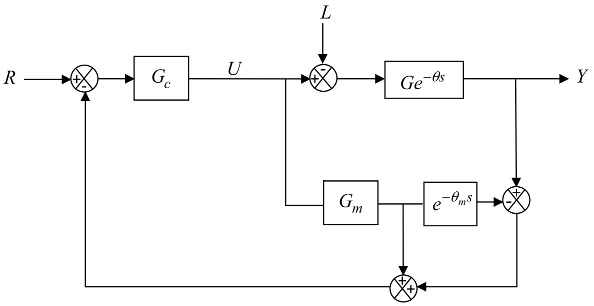


Fig. 8.9 Basic compensation scheme for systems with long dead time

$G_{c2}(s) = 0.5(s + 1)$ results in an approximate stabilized transfer function of the form $G_v(s) = \frac{G(s)}{1+GG_{c2}(s)} = \frac{4e^{-2s}}{2s+1}$. Thus, the inner-loop controller plays an important role in changing the open-loop unstable process $G(s)$ to a stable process $G_v(s)$. Then, the error path controller provides significantly improved performances. The step response shows a maximum overshoot of 10%, a settling time of 15.56 s, a rise time of 2.73 s, and zero steady-state error. The gain and phase margins are $g_m = 1.988$ and $\phi_m = 77.8°$ at phase and gain crossover frequencies of $\omega_p = 1.11$ rad/s and $\omega_g = 0.3$ rad/s, respectively.

8.3 Dead Time Compensation

When there is a large time delay, series or feedback compensation with PI/PID control is difficult because of the limitations imposed by the time delay on the system performance and stability. The conventional Smith predictor compensator shown in Fig. 8.9, where G_m denotes the delay free part of the process model, is a popular and very effective long dead-time compensator for stable processes [5]. The closed-loop transfer function between the output and the setpoint and the input disturbance (L) are given respectively as

$$Y_r(s) = \frac{GG_c e^{-\theta s}}{1 + G_m G_c + G_c(Ge^{-\theta s} - G_m e^{-\theta_m s})}, \tag{8.1}$$

$$Y_L(s) = \frac{Ge^{-\theta s}(1 + G_m G_c[1 - e^{-\theta_m s}])}{1 + G_m G_c + G_c(Ge^{-\theta s} - G_m e^{-\theta_m s})}, \tag{8.2}$$

and $G_m e^{-\theta_m s}$ and $Ge^{-\theta s}$ are the transfer functions of the plant model and the plant, respectively. Based on the assumption that the model used perfectly matches the plant dynamics, that is, $G_m = G$ and $\theta_m = \theta$, (8.1) and (8.2) reduce to

$$Y_r(s) = \frac{GG_c e^{-\theta s}}{1 + GG_c}, \tag{8.3}$$

$$Y_L(s) = \frac{Ge^{-\theta s}(1 + GG_c[1 - e^{-\theta s}])}{1 + GG_c}. \tag{8.4}$$

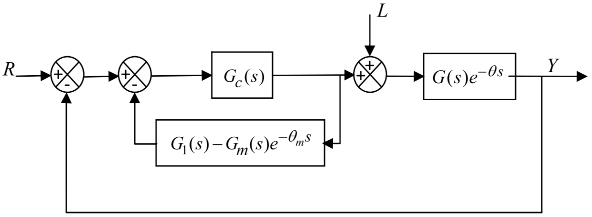


Fig. 8.10 Watanabe and Ito's control structure

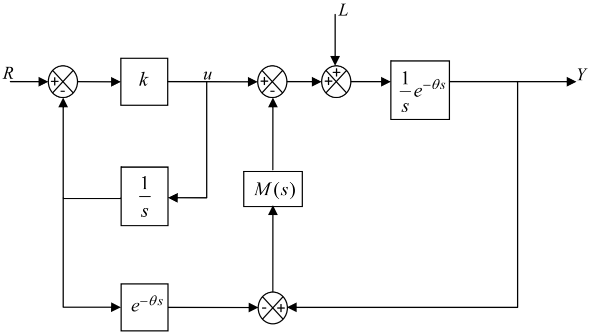


Fig. 8.11 Åstrom et al.'s control structure

The main advantage of the Smith predictor method is that the time delay term is effectively taken out of the denominator of the transfer functions relating the process output to setpoint and load disturbance inputs. Numerous simple compensators can be designed based on the closed-loop dynamics given in (8.3) and (8.4). However, control design to achieve robust stability and robust performance is not discussed here.

Watanabe and Ito [6] pointed out that if the process has poles near the origin in the left half-plane, then the responses by the scheme in Fig. 8.9 may be sluggish enough to be unacceptable. They proposed a structure as shown in Fig. 8.10 for overcoming such problems by choosing

$$G_1(s) = \frac{G_m(s)}{1 + \theta_m s}. \tag{8.5}$$

Since then a number of methods, presented in the following Sect. 8.3.1, have been proposed to overcome the problem of controlling a process with an integrator and long dead time.

8.3.1 Modified Smith Predictors for Integrating Processes

The structure of Åström, Hang, and Lim's Smith predictor [7] is shown in Fig. 8.11. The controller decouples setpoint response from load disturbance response and gives better responses to step input and disturbance signals. In the figure, k is the

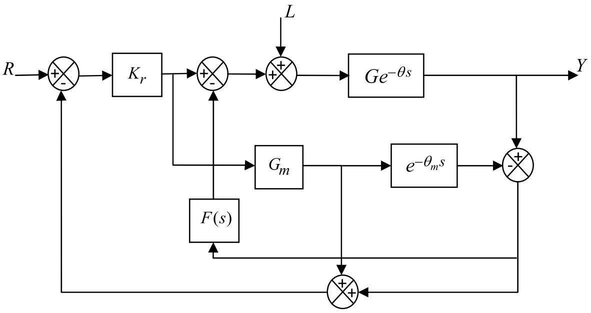


Fig. 8.12 Mataušek and Micić's control structure

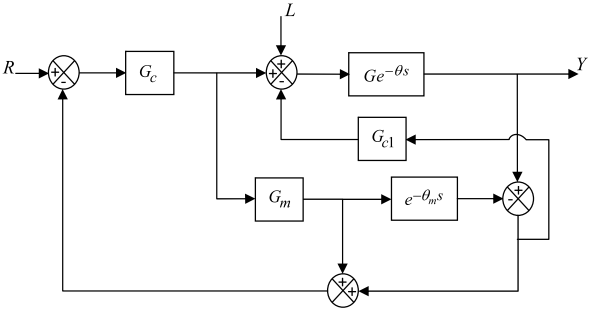


Fig. 8.13 Compensation scheme for integrating systems with long dead time

controller gain. Their choice of

$$M(s) = \frac{C(s)}{1 + C(s)/(\theta s^2 + s) - C(s)e^{-\theta s}/s}, \tag{8.6}$$

where $C(s)$ is a PI or PID controller, gives the same load disturbance as Watanabe and Ito's controller. However, it is possible to improve the load response by choosing a different transfer function $M(s)$.

Mataušek and Micić's dead-time compensator [8] for controlling higher-order processes with integral action and long dead time, as shown in Fig. 8.12, possesses a gain controller K_r and a PD controller of the form $F(s) = \frac{K_b(T_d s+1)}{0.1T_d s+1}$. The control scheme provides considerably faster load disturbance rejection.

The generalized form of the configuration shown in Fig. 8.12 is Fig. 8.13. This general form allows one to have different form of controllers in place of the gains.

The structure of Majhi and Atherton's Smith predictor [9] for controlling stable, unstable, and integrating processes is shown in Fig. 8.14. It is similar to the structure suggested in Fig. 8.13 apart from the controller G_{c1}, which has a major role for an integrating plant and has three controllers designed for different objectives. Of the three controllers, G_{c1} in the inner loop is provided to stabilize an integrating process. The other two controllers, G_c and G_{c2}, are then used to take care of servo-tracking and disturbance rejection respectively by considering the inner loop as an open-loop stable process. The closed-loop response to setpoint and disturbance inputs is given

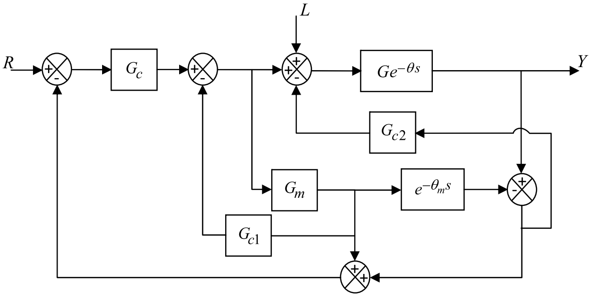


Fig. 8.14 Majhi and Atherton's structure for systems with long dead time

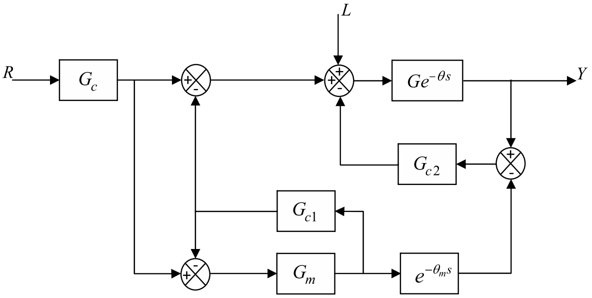


Fig. 8.15 Liu et al.'s control structure

by

$$Y_r(s) = \frac{GG_ce^{-\theta s}(1 + G_{c2}G_me^{-\theta_m s})}{(1 + G_m[G_c + G_{c1}])(1 + G_{c2}Ge^{-\theta s}) + G_c(Ge^{-\theta s} - G_me^{-\theta_m s})}, \quad (8.7)$$

$$Y_L(s) = \frac{Ge^{-\theta s}(1 + G_m[G_c + G_{c1} - G_ce^{-\theta_m s}])}{(1 + G_m[G_c + G_{c1}])(1 + G_{c2}Ge^{-\theta s}) + G_c(Ge^{-\theta s} - G_me^{-\theta_m s})}. \quad (8.8)$$

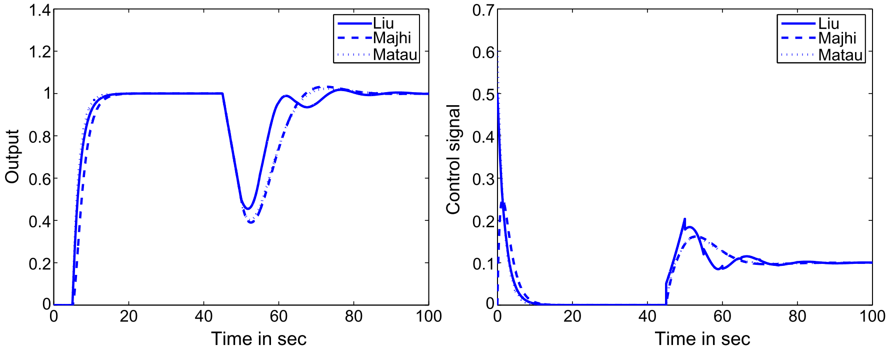
Based on the assumption that the model used perfectly matches the plant dynamics, that is, $G_m = G$ and $\theta_m = \theta$, (8.7) and (8.8) reduce to

$$Y_r(s) = \frac{GG_ce^{-\theta s}}{1 + G(G_c + G_{c1})}, \quad (8.9)$$

$$Y_L(s) = \frac{Ge^{-\theta s}}{1 + G(G_c + G_{c1})} \frac{1 + G(G_c + G_{c1}) - GG_ce^{-\theta s}}{1 + GG_{c2}e^{-\theta s}}. \quad (8.10)$$

It is apparent from (8.9) and (8.10) that the new Smith predictor has decoupled the load response from the setpoint response. Controllers G_c, G_{c1} and G_{c2} designed using simple techniques are of P, PI or PID types.

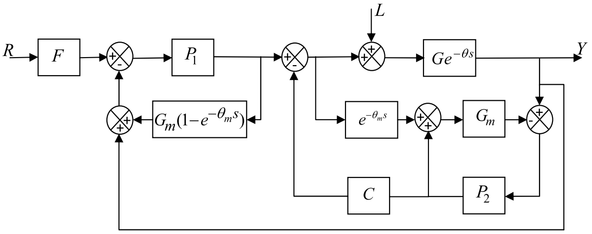
Based on the Majhi and Atherton's control configuration, Liu et al. [10] introduced a modified Smith predictor control structure as shown in Fig. 8.15. G_{c1} is chosen to be a stabilizing gain controller and G_{c2} a PID controller. However, the



(a) Unit step responses (b) Control signals

Fig. 8.16 Simulation results of Example 8.4

Fig. 8.17 Hang et al.'s control structure



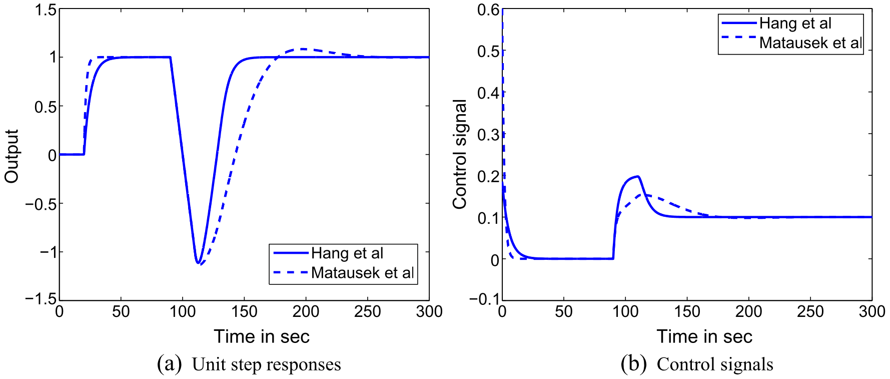
form of the setpoint filter is

$$G_c(s) = \frac{1 + G_{c1}G_m}{G_m} \frac{1}{(\lambda s + 1)^n}, \tag{8.11}$$

where n is the order of the integrating process.

Example 8.4 Consider the integrating process transfer function $G_p(s) = \frac{e^{-5s}}{s}$. Maintaining the same rise time for fair comparison, controller parameters are estimated using the tuning formulae given by Mataušek and Micić, Majhi and Atherton, and Liu et al. Figure 8.16(a) shows the responses to setpoint and load disturbance inputs. The resulting control signals are shown in Fig. 8.16(b). It is seen that apparently improved disturbance response can be obtained by adding more tuning parameters.

Hang et al. [11] proposed a modified Smith predictor (see Fig. 8.17) for processes with double integrators using a rapid load estimator. The load estimator does not involve the solution of a closed-loop equation that contains dead time. Further, the scheme can be extended to the case of single integrator with large time delay by



(a) Unit step responses

(b) Control signals

Fig. 8.18 Nominal responses of Example 8.5

setting $F = C = 1$. The transfer function form of the controllers can be given as

$$\begin{aligned} P_1 &= K_{p1}\frac{1+T_{d1}s}{1+0.1T_{d1}s}, \\ P_2 &= K_{p2}\frac{1+T_{d2}s}{1+0.1T_{d2}s}, \qquad (8.12) \\ C(s) &= \frac{1+(T+\theta)s}{1+Ts}, \end{aligned}$$

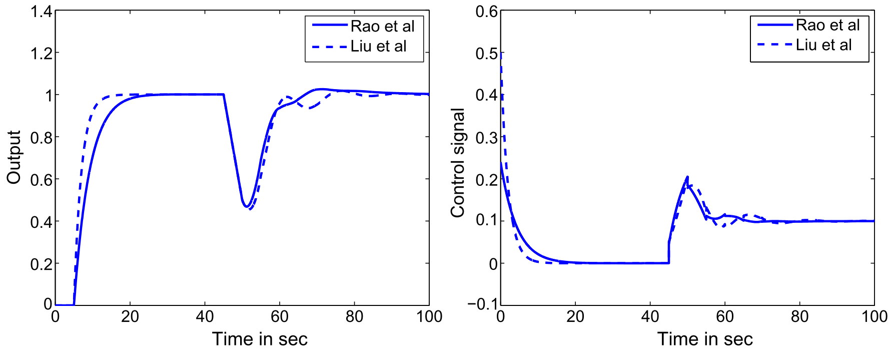
where K's and T_d's are the parameters of the PD controllers, and $T = T_{d2}$. T_d's are set to zero in the case of single integrator.

Example 8.5 Consider the integrating process transfer function with single integrator and large time delay $G_p(s) = \frac{e^{-20s}}{s}$. The nominal performances given by Hang et al.'s and Mataušek and Micić's compensation schemes and tuning algorithms are shown in Figs. 8.18(a) and (b). A faster load disturbance rejection and a similar or better setpoint response are obtained in comparison with the previous method.

Rao and Chidambaram [12] used the generalized compensation scheme shown in Fig. 8.13 for controlling integrating and double integrating processes with time delay. G_{c1} is designed using the classical direct synthesis method, and G_{c2} is chosen as a PD controller designed using optimal gain and phase margin approaches. The benefit of using the simple structure along with the suitable tuning algorithm for the integrating plus time delay process $G_p(s) = \frac{e^{-5s}}{s}$ is illustrated in Fig. 8.19.

8.3.2 Modified Smith Predictors for Unstable Processes

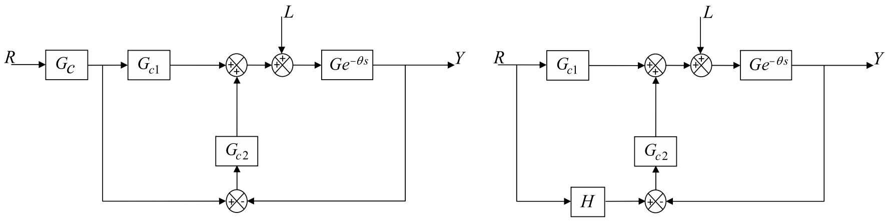
The modified Smith predictor introduced by Kwak et al. [13] and Zhang et al. [14] for the control of unstable processes is shown in Figs. 8.20(a) and (b), respectively.



(a) Unit step responses

(b) Control signals

Fig. 8.19 Results obtained using improved tuning algorithm and simple control structure



(a) Kwak et al.'s structure

(b) Zhang et al.'s structure

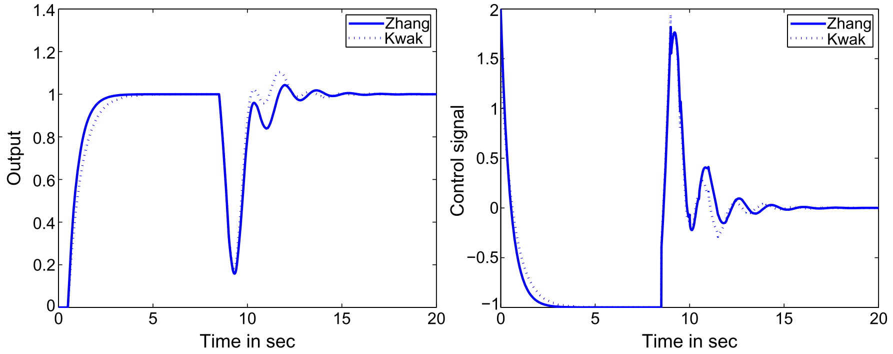
Fig. 8.20 Modified Smith predictors for unstable processes

The transfer function forms of the controllers in Fig. 8.20(a) are $G_c = \frac{e^{-\theta s}}{\tau s+1}$, $G_{c1} = (Ge^{-\theta s})^{-1}$, and $G_{c2} = K_c(1 + \frac{1}{T_i s} + T_d s)$. τ is the user-defined time constant for the desired trajectory, and K_c, T_i, and T_d are the parameters of the parallel PID controller.

Zhang et al. pointed out that G_{c1} cannot be implemented physically. They proposed a simplified structure as shown in Fig. 8.20(b). With the choice of $H(s) = G_{c1}Ge^{-\theta s}$, the feedforward controller becomes $G_{c1}(s) = \frac{G^{-1}}{\tau s+1}$.

Example 8.6 Consider the first-order process model $G_p(s) = \frac{e^{-0.5s}}{s-1}$. The setpoint input and control variable responses given by Kwak et al and Zhang et al.'s methods are shown in Figs. 8.21(a) and (b). As expected, the difference between the simulation results of the two methods is minimum, and the simulated results concur with each other almost perfectly.

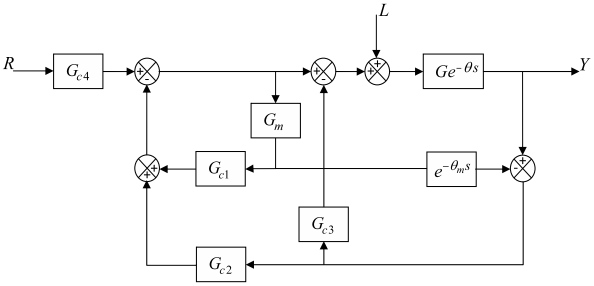
Motivated by the work of Majhi and Atherton (see Fig. 8.14), Lu et al. [15] included four controllers to meet different performance objectives in their modified Smith predictor control scheme as shown in Fig. 8.22. The controllers are of P or PD types without any derivative filters. The scheme has one more degree of freedom to improve disturbance response.



(a) Output responses (b) Control variables responses

Fig. 8.21 Setpoint and control variable responses

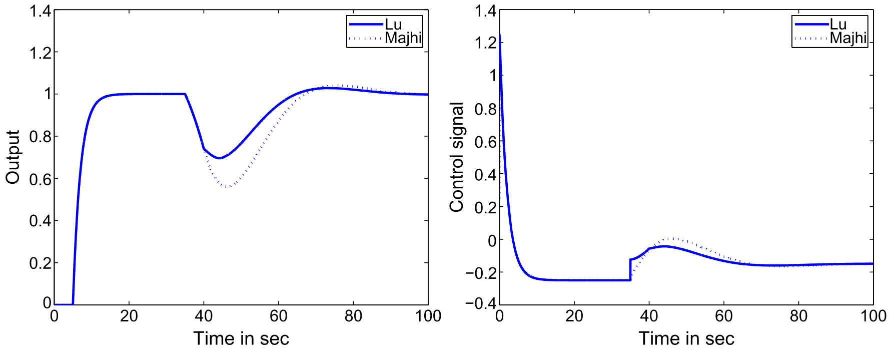
Fig. 8.22 Lu et al.'s structure



Example 8.7 For a comparison of performance, consider the unstable process $G_p(s) = \frac{4e^{-5s}}{10s-1}$. Lu et al. designed the four controllers so as to achieve the same setpoint response as that of Majhi and Atherton. Controller settings for both methods are given in [15]. The simulation results in Fig. 8.23 show that Lu et al.'s compensation scheme has improved disturbance rejection performance.

A modified form of Smith predictor for controlling unstable second-order-plus-time-delay processes with/without a zero, as shown in Fig. 8.24, has recently been proposed by Rao and Chidambaram [16]. The controller G_{c1} is designed by the synthesis method, and G_{c2} and the stabilizing controller G_{c3} are PD and PID controllers, respectively. A first-order filter with $\tau_f = \theta$ is placed in the feedback path to improve robustness.

Example 8.8 Consider the second-order unstable process model $G_p(s) = \frac{2e^{-0.3s}}{(3s-1)(s-1)}$. For qualitative performance comparison, controller settings given in [16] are considered. Figure 8.25 shows the responses given by the compensation schemes of Rao and Chidambaram and Liu et al. under perfect model conditions. With the same control effort, improved performance, particularly for the load disturbance rejection, is obtained by Rao and Chidambaram's method.

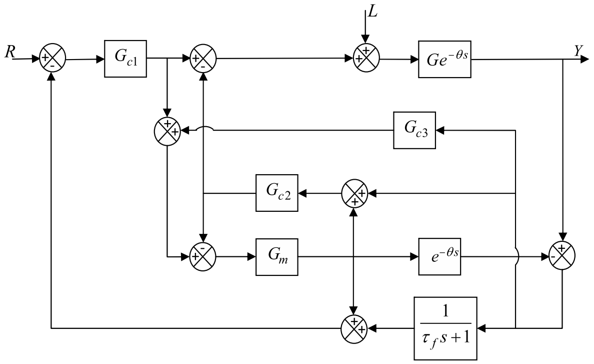


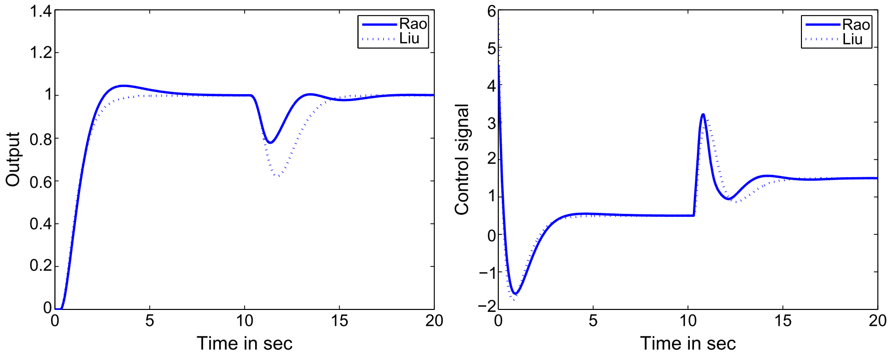
(a) Output responses

(b) Control variables responses

Fig. 8.23 Simulation results of Example 8.7

Fig. 8.24 Rao and Chidambaram's control structure





(a) Output responses

(b) Control variables responses

Fig. 8.25 Nominal performances by the compensation schemes

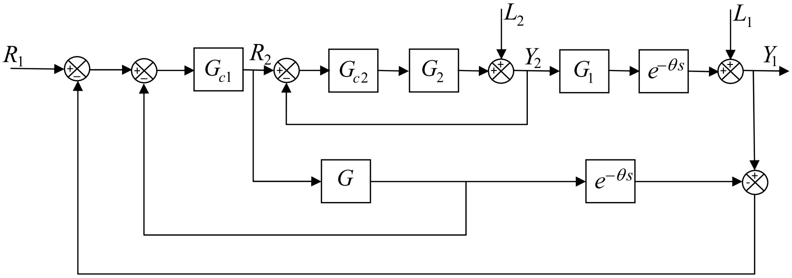


Fig. 8.26 Kaya et al.'s cascade control structure

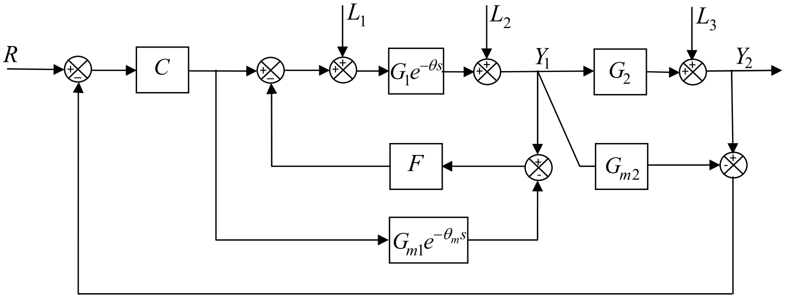


Fig. 8.27 Liu et al.'s control structure

8.3.3 Improved Cascade Control Systems

Classical cascade control can yield improved response of the system to load changes. Kaya's [17] improved cascade control system, a conventional cascade control scheme combined with Smith predictor, is shown in Fig. 8.26, where $G_1e^{-\theta s}$ and G_2 describe the plant dynamics. G_{c1} and G_{c2} are the PI/PID type controllers, whereas $G = \frac{G_{c2}G_1G_2}{1+G_{c2}G_2}$.

Liu et al.'s two-degree-of-freedom cascade control scheme [18] is shown in Fig. 8.27, where G_{m1} and G_{m2} are process models of G_1 and G_2. The decoupled outer- and inner-loop controllers C and F are designed primarily for setpoint tracking and load disturbance rejections.

Example 8.9 Consider the cascade system with $G_2 = \frac{3}{10s+1}$ and $G_1e^{-\theta s} = \frac{e^{-10s}}{(20s+1)(3s+1)}$. Simulation is performed using the set of control parameters proposed in [18]. Figure 8.28 shows the responses obtained by the compensation schemes of Kaya and Liu et al. under perfect model matching conditions. The latter's compensation method yields improved performances because of decoupling of the setpoint and load disturbance responses, both of which can be optimized independently.

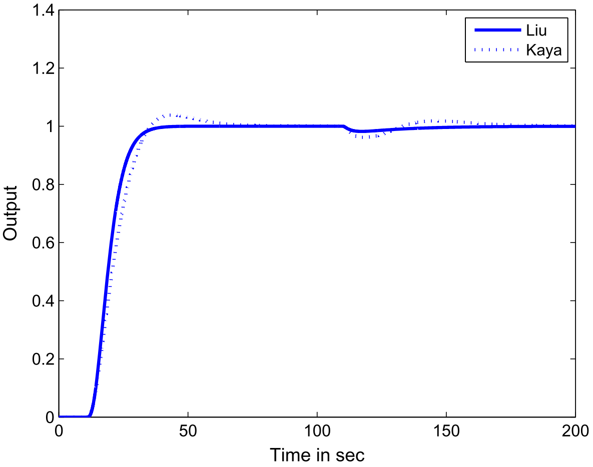


Fig. 8.28 Responses of Example 8.9

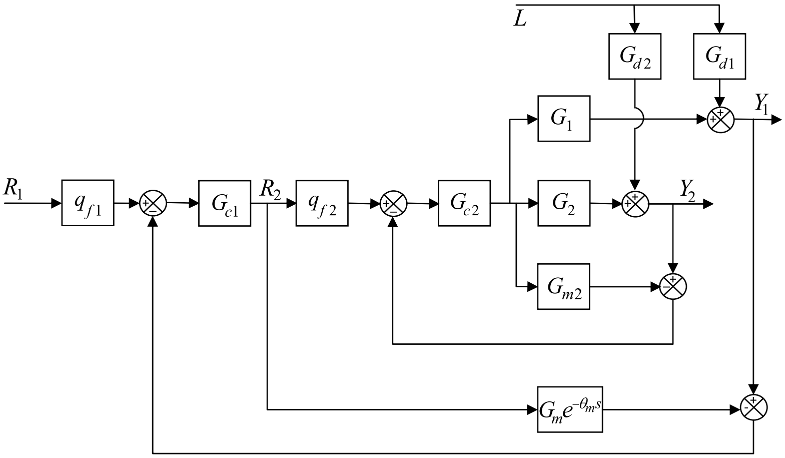


Fig. 8.29 Lee et al.'s parallel cascade control

Parallel cascade is beneficial when the secondary loop has a faster dynamic response, and the rejection of the disturbance in the secondary output reduces the steady-state output error in the primary loop. Figures 8.29 and 8.30 show the parallel cascade control structure of Lee et al. [19] and Rao et al. [20], respectively. In the figures, G_1 and G_2 are the primary and secondary processes, G_{c1} and G_{c2} are the primary and secondary controllers, and G_{d1} and G_{d2} are the disturbances in primary and secondary loops, respectively. q_{f1} and q_{f2} are first-order filters.

Example 8.10 Consider the cascade system with $G_2 = G_{d2} = \frac{3.1e^{-9s}}{30s+1}$ and $G_1 = G_{d1} = \frac{1.24e^{-33s}}{30s+1}$. To assess the performances of the structures, primary and secondary controller settings given in [20] are used. The comparative simulation results

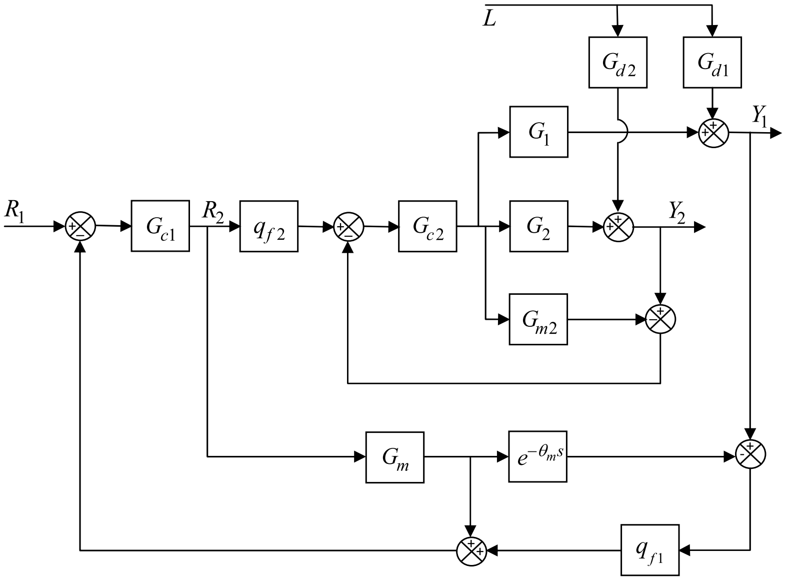
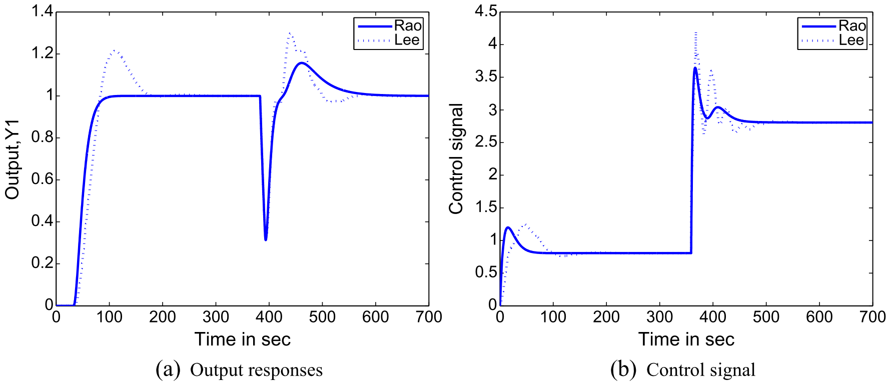


Fig. 8.30 Rao et al.'s structure



(a) Output responses (b) Control signal

Fig. 8.31 Simulation results of Example 8.10

in Fig. 8.31 show significant improvement in the closed-loop performances by the Rao et al.'s method due to the Smith predictor implementation for parallel cascade control system with delay.

8.4 Conclusions and Future Perspectives

It is evident from the simulation Examples 8.1–8.3 that the two-degree-of-freedom series-feedback compensation yields satisfactory dynamic responses. Comparative study of some modified structures of the Smith predictor for controlling integrating and unstable processes with time delay are presented. Simulation results of series and parallel cascade structures are included to show that the secondary process has a major role in rejecting disturbance and improving the dynamic performance of the closed-loop system.

Further, it is observed that improved performances of a closed-loop system are obtained based mainly on suitable choice of a control structure that decouples the servo and regulatory problems. There exist performance limitations in control system because of the wrong choice of feedback control structure. Control structure design has not drawn the attention it deserves compared to the vast amount of the literature available on controller design. From that perspective, the researchers in this area have ample scope for designing unified control structures for decoupling setpoint tracking, feedforward compensation, and disturbance rejection dynamics for a variety of plants encountered in process industries.

References

1. Majhi, S.: Advanced Control Theory—A Relay Feedback Approach. Cengage Learning Asia, Singapore (2009)
2. Atherton, D.P., Majhi, S.: Limitations of PID controller. In: Proc. of American Control Conference, San Diego, pp. 3843–3847 (1999)
3. Ziegler, J.G., Nichols, N.B.: Optimum settings for automatic controllers. Trans. Am. Soc. Mech. Eng. **64**, 759–768 (1942)
4. Majhi, S., Atherton, D.P.: Autotuning and controller design for processes with small time delays. IEE Proc., Control Theory Appl. **146**, 415–425 (1999)
5. Smith, O.J.M.: A controller to overcome deadtime. ISA J. **6**, 28–33 (1959)
6. Watanabe, K., Ito, M.: A process-model control for linear systems with delay. IEEE Trans. Autom. Control **26**, 1261–1269 (1981)
7. Åström, K.J., Hang, C.C., Lim, B.C.: A new Smith predictor for controlling a process with an integrator and long dead-time. IEEE Trans. Autom. Control **39**, 343–345 (1994)
8. Mataušek, M.R., Micić, A.D.: On the modified Smith predictor for controlling a process with an integrator and long dead-time. IEEE Trans. Autom. Control **44**, 1603–1606 (1999)
9. Majhi, S., Atherton, D.P.: Modified Smith predictor and controller for processes with time delay. IEE Proc., Control Theory Appl. **146**, 359–366 (1999)
10. Liu, T., Cai, Y.Z., Gu, D.Y., Zhang, W.D.: New modified Smith predictor scheme for integrating and unstable processes with time delay. IEE Proc., Control Theory Appl. **152**, 238–246 (2005)
11. Hang, C.C., Wang, Q.-G., Yang, X.-P.: A modified Smith predictor for a process with an integrator and long dead time. Ind. Eng. Chem. Res. **42**, 484–489 (2003)
12. Rao, A.S., Chidambaram, M.: Set point weighted modified Smith predictor for integrating and double integrating processes with time delay. ISA Trans. **46**, 59–71 (2007)
13. Kwak, H.J., Sung, S.W., Lee, I.: A modified Smith predictor for unstable processes. Ind. Eng. Chem. Res. **38**, 405–411 (1999)
14. Zhang, W., Gu, D., Wang, W., Xu, X.: Quantitative performance design of a modified Smith predictor for unstable processes with time delay. Ind. Eng. Chem. Res. **43**, 56–62 (2004)
15. Lu, X., Yang, Y., Wang, Q., Zheng, W.: A double two-degree-of-freedom control scheme for improved control of unstable delay processes. J. Process Control **15**, 605–614 (2005)
16. Rao, A.S., Chidambaram, M.: Analytical design of modified Smith predictor in a two-degree-of-freedom control scheme for second order unstable processes with time delay. ISA Trans. **47**, 407–419 (2008)

17. Kaya, I.: Improved performance using cascade control and a Smith predictor. ISA Trans. **40**, 223–234 (2001)
18. Liu, T., Gu, D., Zhang, W.: Decoupling two-degree-of-freedom control strategy for cascade control systems. J. Process Control **15**, 159–167 (2005)
19. Lee, Y., Skliar, M., Lee, M.: Analytical method of PID controller design for parallel cascade control. J. Process Control **16**, 809–818 (2006)
20. Rao, A.S., Seethaladevi, S., Uma, S., Chidambaram, M.: Enhancing the performance of parallel cascade control using Smith predictor. ISA Trans. **48**, 220–227 (2009)

Chapter 9
Robust Multivariable Tuning Methods

Reza Katebi

9.1 Introduction

The rapid growth in the complexity of modern process plants both in terms of material flow and energy exchange have substantially increased the number of feedback control loops for maintaining desired production conditions and product quality. Traditionally, PID controllers are used in large numbers in all industries as a single-loop controller. The controllers come in many different forms and are often packaged as standard products. The popularity of PID controllers is due to their functional simplicity, which allows process engineers to commission them in a simple and straightforward manner. The most attractive practical property of PID is the guaranteed property of the 'Integral' action to eliminate constant set-point error and disturbance offsets.

In practice, most industrial processes have multiple numbers of inputs and outputs. The inputs and outputs are arranged or paired to minimise the interaction between the control loops. Static or dynamic decoupling is also used to minimise the loop interactions and make the system diagonally dominant. Multiloop PID or non-interacting controllers can then be used to control MIMO processes. Koivo and Tanttu [10] gave an early survey of multivariable PID tuning techniques. When the process interactions are modest, a diagonal PID controller structure is often adequate. This type of structure is simple and easy to understand.

While there are a large number of methods to tune scalar PID controllers, the numbers of multiloop PID tuning methods are few. Attempts have been made to extend some SISO methods to multivariable systems, and some new methods have been developed in recent years. This chapter reviews and compares some of the important multivariable tuning methods in terms of their stability and performance robustness.

R. Katebi (✉)
Industrial Control Centre, Department of Electronic and Electrical Engineering, University of Strathclyde, 204 George Street, Glasgow G1 1XW, UK

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_9, © Springer-Verlag London Limited 2012

Despite the success of advanced control systems, the use of PIDs is still widespread, particularly for cheap control systems. The main reason for this popularity is the ease of tuning and commissioning SISO PID controllers. In most systems, SISO PIDs are used. However, many systems exhibit significant interactions between the various loops. Thus a true multivariable strategy is necessary. Another objective of this chapter is to review and compare the existing simple multivariable PID methods and assess their applicability to industrial control problems. Only those methods which require minimum modelling efforts are considered. These methods are often referred to as model-free control design techniques. Katebi et al. [9] surveys the various MIMO PID tuning methods within the control literature and suggests some techniques, which are more appropriate for simple designs. The model-based methods are also reviewed, and some model-free on-line 'auto tuning' techniques are discussed, so-called because it is possible to automate them to the extent of simply pushing a button. Zhuang and Atherton [30] have used a MIMO extension of the Ziegler–Nichols method for auto tuning of two-input two-output (TITO) systems. Loh et al. [12] extend the relay feedback idea of Åström, [1], by tuning one loop at a time in the manner of Sequential Loop Closing. Palmor et al. [18] also use relay feedback, but tune both loops of a TITO process together. The Internal Model Control (IMC) technique was introduced in [20]. Wang and Wu [25] describe a method for autotuning fully cross-coupled multivariable PID controllers from decentralised relay feedback. A thorough review of the state of PID auto tuning is given in [28].

The model-free off-line methods considered in this chapter given by Davison [2], Penttinen and Koivo [19] and Maciejowski [14] are the focus of this study, as they require little design effort and are based on step tests or frequency responses at a single point. The robust servomechanism problem was introduced by Davison [2] to provide a general controller design method with guarantees of asymptotic stability and asymptotic tracking given disturbances of a particular form, and plant model perturbations that did not result in closed-loop instability. More pertinently, a detailed mathematical model of the plant is not necessary, and the controller may be constructed based on simple open-loop tests. Penttinen and Koivo [19] suggest a way to diagonalise the plant at very low and very high frequencies, which is in fact an extension of Davison's work. The decoupling matrix at a particular bandwidth frequency, suggested in [14] with regard to the Sequential Return-Difference method, is also investigated, as an intermediate approach to that of Penttinen and Koivo [19]. Based on these three techniques, a new combined method (Martin–Katebi Method) is proposed which incorporates all the relevant features required for cheap and simple control system application [15].

The chapter is organised as follows. Section 9.2 is devoted to the description of tuning methods where both parametric and nonparametric methods are presented. The stability and performance robustness measures are defined in Sect. 9.3. The criteria for tuning the controllers are presented in Sect. 9.4. A simulation study to compare the stability and robustness performance of the tuning methods is presented in Sect. 9.5. Section 9.6 describes the application of methods to the ship positioning control problem. Finally conclusions are drawn in Sect. 9.7.

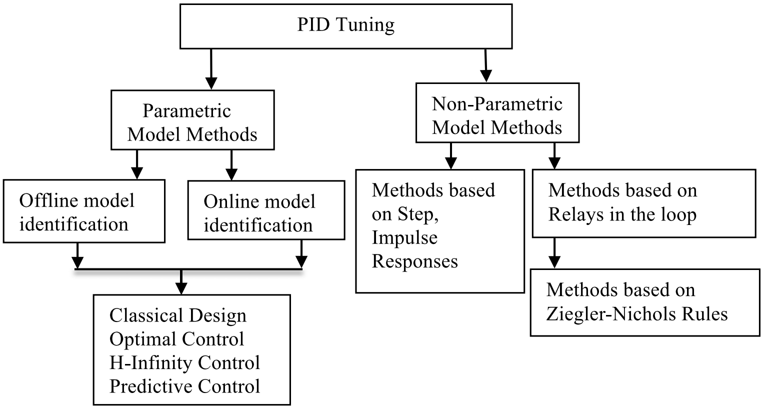

PID Tuning
Parametric Model Methods
Non-Parametric Model Methods
Offline model identification
Online model identification
Methods based on Step, Impulse Responses
Methods based on Relays in the loop
Methods based on Ziegler-Nichols Rules
Classical Design
Optimal Control
H-Infinity Control
Predictive Control

Fig. 9.1 The PID tuning methods classification

9.2 Multivariable PID Tuning Methods

The tuning methods described in the literature can be classified under two groups, as shown in Fig. 9.1, namely, the parametric and non-parametric model methods. For the first group of methods, a model is built using either historical data, or online parameter identification is used to develop a model for control design. These methods require a linear model of the process, in transfer function matrix form or a state space model, over the frequency range of interest. These methods are more suitable for off-line PID tuning due the high design and computation effort, but they can also be used in an online mode if fast computing processors are used for real-time control. The second group of methods uses only partial modelling information, usually steady-state model or critical frequency points. These methods are more suitable for online use where they can usually be applied without the need for extensive a priori plant information and hence are preferred by the plant operators [8].

9.2.1 Parametric Tuning Methods

Some important methods, which are mostly extension of the scalar case to multivariable systems, are briefly described here.

9.2.1.1 BLT Method

Luyben [13] proposed the Biggest Log modulus Tuning (BLT) method. It is an iterative method to tune a set of multi-loop PI/PID controllers. The Ziegler–Nichols settings for each individual loop are first calculated using the ultimate gains and periods of the diagonal elements the plant transfer function. These settings are then detuned by some factor F that is usually between 2 and 5. A stability test is then performed based on the Nyquist plot of the closed-loop characteristic equation. Nyquist plots are used to determine an appropriate value for F. The BLT method is an offline method that requires good knowledge of the process dynamics. Assuming a plant of transfer function matrix $G(s)$ and a diagonal controller, $C(s)$, the steps of tuning are:

i. Calculate the Ziegler–Nichols setting for each individual loop.
ii. $G_{ci}(s) = K_{ci}(1 + \frac{1}{\tau_{ci}}s)$, $K_{ci} = \frac{K_{\mathrm{ZN}i}}{F}$, $\tau_{ci} = \tau_{\mathrm{ZN}i}$, ZN = Ziegler–Nichols.
iii. Define the function $W(j\omega) = -1 + \det[I + G(j\omega) \cdot C(j\omega)]$, $2 < F < 5$.
iv. Calculate the closed-loop function $L_c(\omega) = 20\log_{10}|W(j\omega)/(W(j\omega)+1)|$.
v. Adjust the tuning factor F until the peak in the L_c is equal to $2N$, where N is the number of SISO loops.

9.2.1.2 Gain and Phase Margin Method

The Direct Nyquist Array (DNA) method (Rosenbrock [21]) is based on shaping the Gershgorin bands using a trial and error and adopting a graphical approach to perform the control design. This is often tedious and time consuming. Ho and Xu [7] have proposed a method of tuning multiloop PID controllers based on gain and phase margins specifications. The method can tune the multiloop controllers online using the Direct Nyquist Array Method to meet specified system robustness and performance. When interactions are significant, the multiloop PID controller design method often fails to give an acceptable response. Using feed-forward controllers or de-couplers, the interaction can be appropriately compensated so that effective non-interacting single SISO processes are obtained. Hang et al. [6] have also reported a similar tuning method using de-coupling controllers.

9.2.1.3 Minimum Variance Control

Yuosf et al. [29] have proposed a multivariable PID controller based on the solution of the minimum variance control problem. The ARMAX model is used to represent the plant as follows:

$$A(z^{-1})y(t) = B(z^{-1})u(t) + C(z^{-1})w(t). \tag{9.1}$$

The PID controller is assumed to have the following structure:

$$\begin{aligned}\Delta u(t) = {} & K_I w(t) - [K_P + K_I + K_D]y(t) \\ & - [K_P + 2K_D]y(t-1) - K_d y(t-2)\end{aligned} \tag{9.2}$$

where the PID gains can be calculated from

$$
\begin{aligned}
K_P &= V[-\tilde{F}_1 - 2\tilde{F}_2]\rho, \\
K_P &= V[\tilde{F}_0 + \tilde{F}_1 + \tilde{F}_2]\rho, \\
K_P &= V\tilde{F}_2\rho,
\end{aligned}
\tag{9.3}
$$

where $\tilde{F} = \tilde{F}_0 + \tilde{F}_1 z^{-1} + \tilde{F} z^{-2}$ is the solution to a matrix polynomial equation of the type

$$
\begin{aligned}
C &= EA + z^{-k}F, \\
\left(1 - z^{-1}\right)V^{-1} &= EB + Q.
\end{aligned}
\tag{9.4}
$$

The method is more suitable for off-line application as it requires an ARMAX model and extensive computation.

9.2.1.4 Optimal Control-Based Methods

In cost function-based methods, the control design objectives are expressed in the form of a quadratic performance criterion. The aim is to minimise a single objective function for determining PID controller setting. A robust LQ design procedure has been reported by Sandelin et al. [22]. A multi-objective optimisation method is used to tune PID controllers by Wang and Wu [25]. The design specifications used are nominal performance, minimum input energy, operational constrains, and robust stability.

Wang and Wu [25] have proposed a multi-objective optimisation method to calculate the parameters of the PID controller. A cost criterion for nominal plant performance is defined and minimised subject to input constraints. There is no guarantee that the optimisation algorithm converges to a solution and the method is computationally expensive.

Yanchevsky [27] presented an optimal procedure (with respect to the overall plant performance) for selecting the PID parameters of a multiloop controller. This method is based on a centralised LQ problem. The solution is given in discrete time and a linear feedback results with constant optimal PID parameters calculated from the matrix Ricatti equation.

9.2.1.5 Internal Model Control (IMC)

Garcia and Morari introduced IMC [5], and the general design methodology of IMC with PID structure was studied by Rivera et al. [3]. They introduced the concept of obtaining PID controller parameters by approximating the simple feedback form of an IMC controller. As a straightforward approach, the control design is based on an a priori process model and a low-pass filter included for robustness. Lieslehto and Koivo [23] developed a multivariable tuning method based on the IMC technique. Dong and Brosilow [11] have designed robust multivariable PID controllers via IMC. They have obtained a general form for the parameters of the multivariable PID controllers by Maclaurin series expansion of the simple feedback form of a multivariable IMC controller.

9.2.1.6 Robust Decentralised Methods

Skogestad et al. [4], have reported an interesting method for simultaneously tuning decentralised PI controllers using the standard μ-synthesis criterion:

$$J = \min_{K(s)} \bigl\{ \sup \mu_{\Delta G(s)} M(S) \bigr\}.$$

For a full description of this approach, see Gagnon et al. [26].

9.2.2 Non-parametric Methods

The second group of tuning methods is based on the assumption that a detailed model of the plant is not available. The steady-state gain matrix is usually assumed to be known or found from the historical data. The following assumptions are usually made:

- The plant is linear and time invariant.
- The uncontrolled plant is square and stable.
- The controlled variables are measurable.
- The classes of input disturbance and reference inputs are known.
- The system is controllable by a diagonal PID controller.

The multivariable PID tuning techniques to be studied are the Davison method [2], the Penttinen and Koivo method [19], the Maciejowski method [14], a new proposed method (Martin–Katebi method) [15], Generalised Zeigler–Nichols Method [16] and the Relay Feedback Method [1].

9.2.2.1 Davison Method

The approach outlined in the paper [2] is taken with the only assumptions on the plant being that it is linear, time-invariant and open-loop stable. It is shown that conditions for a feedforward and robust feedback controller to exist, as well as the controller structure itself, can be expressed in terms of the steady-state gain parameters of the plant. Therefore, it is only necessary to consider the case where disturbances are constant or at least slowly varying. With these conditions in mind, Davison's method reduces to finding the steady-state gain matrix of the plant for a step input. The feedback controller is then the inverse of this matrix if it is of full rank. Note that there is no proportional term in this case and a multiplier ε is included for tuning the resulting closed-loop system.

The expression for the controller is

$$\underline{u}(s) = K_i \frac{1}{s} \underline{e}(s), \quad K_i = \varepsilon G^{-1}(0) \tag{9.5}$$

where K_i is essentially an integral feedback gain, $G(s)$ is the square open-loop transfer function matrix, and the scalar, ε, is the tuning parameter. The procedure for determining ε is known as 'tuning the regulator on-line' and simply consists of making adjustments, starting with a very small positive value and increasing so that the output response of the closed-loop plant for a step function input has a maximum speed of response. Note that each of the multivariable loops is adjusted simultaneously. This approach has been applied successfully to chemical processes, where step tests can be used to find $G(0)$.

9.2.2.2 Penttinen–Koivo Method

This technique [19] alters the Davison method slightly to achieve a diagonalised plant at very low and very high frequencies. The expression for the controller is:

$$\underline{u}(s) = \left(K_p + K_i \frac{1}{s}\right) \underline{e}(s), \quad K_p = (CB)^{-1}, \ K_i = \varepsilon G^{-1}(0). \tag{9.6}$$

The CB matrix comes from the state-space plant model, or in the absence of a model, it is possible to perform tests for determining the value of CB. Observe that $\underline{\dot{y}} = C\underline{\dot{x}} = CA\underline{x} + CB\underline{u}$. If $\underline{x} = 0$ or the plant is at an operating point, then $\underline{\dot{y}} = CB\underline{u}$ or $\Delta\underline{\dot{y}} = CB\Delta\underline{u}$ at the instant an input is applied. Thus, by applying a unit step to each input in turn and measuring the gradient of each output immediately after,

$$CB = \begin{bmatrix} \underline{\dot{y}}_1 & \underline{\dot{y}}_2 & \cdots & \underline{\dot{y}}_m \end{bmatrix} \tag{9.7}$$

where m is the number of plant inputs, and $\underline{\dot{y}}_k$ is the output gradient in response to the kth input step.

The reasoning behind this choice of matrix can most easily be seen using an argument given in [19] as follows. Given a plant in state-space form, the Laurent series expansion of the transfer function

$$G(s) = C(sI - A)^{-1}B \tag{9.8}$$

is

$$G(s) = \frac{CB}{s} + \frac{CAB}{s^2} + \frac{CA^2B}{s^3} + \cdots. \tag{9.9}$$

Therefore, at high frequencies, $G(s) \to CB/s$, and $G(s)K_p \to I/s$.

The proportional gain matrix can be selected as $K_p = (CB)^{-1}p$, where p is a constant scalar tuning parameter. To 'tune the regulator on line', p is increased from a small positive value until the closed-loop response for a step-input reference signal is acceptable. p is then reduced slightly, and ε is increased from a small positive value until the maximum speed of closed-loop response is achieved.

The product of $G(s)$ with K_p and K_i/s approaches pI/s and $CB\varepsilon G(0)^{-1}/s^2$, respectively, at high frequencies. The K_i/s term will generally be negligible compared to K_p at high frequencies, resulting in a closed-loop transfer function,

$$(I + GK)^{-1}GK = \begin{bmatrix} (s+p)^{-1}p & \dots & 0 \\ \vdots & 0 & \vdots \\ 0 & \dots & (s+p)^{-1}p \end{bmatrix}_{\text{at large } s} \tag{9.10}$$

where $K = (K_p + K_i/s)$. From this, it is evident that properly selected p and ε will produce good high- and low-frequency tracking.

9.2.2.3 Maciejowski Method

This method is aimed at decoupling the system at the bandwidth frequency. The gains can be calculated as follows:

$$K_p = pG^{-1}(j\omega_b), \qquad K_i = \varepsilon G^{-1}(j\omega_b), \qquad K_d = \delta G^{-1}(j\omega_b) \tag{9.11}$$

where p, ε and δ are scalar tuning parameters. If a plant model is available, then this method requires the frequency response at a single point. Otherwise, experimental application of sinusoidal inputs to the actual plant at the desired frequency will give values for gain and phase. In the case of a nonlinear system, this experimental approach is not strictly valid, but low-amplitude sinusoidal excitation about the operating point yields a very close approximation to the linearised result.

Clearly, $G^{-1}(j\omega_b)$ will produce complex gains, but to realise such a controller, the gains must be real. Hence, the 'Align' algorithm of Matlab is used to produce a real approximation of the inverse of $G(j\omega)$. This algorithm finds a constant real gain matrix, M, such that

$$J(M,\theta) = \left(G(j\omega_b)M - e^{j\theta}\right)^{\mathrm{T}}\left(G(j\omega_b)M - e^{j\theta}\right), \quad \theta = \mathrm{diag}(\theta_i) \tag{9.12}$$

is minimised. The product of $G(j\omega_b)$ and M is then as close as possible to a diagonal matrix with elements of unity magnitude. If we let $K_p = M$, this produces desirable properties in a multivariable system as each loop will be almost decoupled. In summary, this method aims to create nearly decoupled unity gain open-loop transfer functions from a coupled transfer function matrix. If the open-loop phase is close to $-\pi/2$, then the bandwidth will be close to ω_b, and the closed-loop system will be stable. Of course, this is the ideal case, and if the open loop phase is greater than $-\pi/2$, the bandwidth will be less than ω_b. Conversely, an open-loop phase lower than $-\pi/2$ will result in a higher bandwidth. The system will be unstable if the open-loop phase is $-\pi/2$ or less. This analysis applies to proportional control only, as M is a constant gain matrix. Fine tuning may be achieved with the addition of the integral or derivative terms.

9.2.2.4 Martin–Katebi Method [15]

Maciejowski's control design technique has many tractable properties and an intuitive control structure. Initial results also indicated that the controller was very effective. However, since Maciejowski's control design technique involves plant frequency analysis experiments (to obtain the process model), it is possible that industry acceptance for the technique will be low. Martin and Katebi [15] proposed a new control design technique that retains some of the properties that makes the Maciejowski controller tractable, but eliminates the need for a frequency analysis. The proposed control design technique assumes the following control structure:

$$\underline{u}(k) = K\left(1 + \frac{T_s}{1 - z^{-1}}\right)\underline{e}(k) \tag{9.13}$$

where

$$K = \left[\alpha G(0) + (1 - \alpha) C G_p\right]^{-1}.$$

The proportional and integral feedback gain of the proposed controller is a blend between the inverse of the plant dynamics at zero frequency and the inverse of the plant dynamics at high frequency. Thus, provided that the plant has low-pass frequency characteristics, a good approximation of $G^{-1}(jw_b)$ can be obtained by appropriately selecting the additional controller tuning parameter, α. The fine tuning of α is usually in the interval (0, 1).

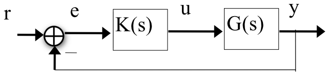
9.2.2.5 Generalised Zeigler–Nichols Method

Niederlinski [16] proposed an extension of SISO Ziegler–Nichols method to MIMO system. A critical point is found by closing the loops with P-controller and bringing the system under stable oscillation. The method is however difficult to use for autotuning. Zhuang and Atherton [30] have used this method for autotuning PID controllers.

9.2.2.6 The Relay Feedback Method

In recent years, the popular relay feedback method has been extended for autotuning multivariable systems. Unlike SISO case, there are an infinite number of critical points for a multivariable system, and this makes the autotuning a difficult task.

Fig. 9.2 Closed-loop system



Instead, the research has been concentrated on extending relay feedback to multivariable systems. Two approaches are followed. The first uses partial relay feedback using a sequential or an iterative tuning procedure by tuning the system loop by loop, closing each loop once it is tuned, until all the loops are closed. To tune each loop, a relay feedback configuration is set up to determine the ultimate gain and frequency. The PI/PID settings are then computed on the basis of this information. Examples of this approach can be found in Loh et al. [12] and Menani and Koivo [18].

In the second approach, the relay feedback configuration is simultaneously applied to all the loops. The system output can in general oscillate at different frequencies and with different amplitudes. For typical multivariable systems, the outputs will oscillate at the same frequency but with different phases. In addition, the system steady state is also needed to determine the critical points. For autotuning purposes, it is more convenient to find the critical points of all loops at the same frequency. Examples of this approach can be found in Zhuang and Atherton [30], Palmor et al. [24] and Wang et al. [17].

9.3 Robustness Test

Assuming a nominal model denoted by $G(s)$, the closed-loop system of Fig. 9.2 may be formed. The sensitivity, control sensitivity and complementary sensitivity functions are defined as follows:

$$\text{Sensitivity:}\quad S = (I + GK)^{-1}, \tag{9.14}$$

$$\text{Control Sensitivity:}\quad C = K(I + GK)^{-1}, \tag{9.15}$$

$$\text{Complementary Sensitivity:}\quad T = GK(I + GK)^{-1}. \tag{9.16}$$

Davison [2] defined the robust multivariable PID controller as the one which stabilises the plant under first-order perturbation in the plant state space matrices and asymptotically regulates the system to the desired set-points. Since then, the robustness issue has been extensively researched, and useful results have been developed to quantify the robustness of multivariable controllers.

For input multiplicative uncertainty defined in Fig. 9.3, the standard closed-loop system is defined as

$$M(s) = \begin{bmatrix} -W_1 T & W_1 T G^{-1} \\ -W_2 SG & W_2 S \end{bmatrix}(s). \tag{9.17}$$

Note that other type of perturbation may also be considered, but for comparison purposes, the analysis is restricted to input multiplicative uncertainty.

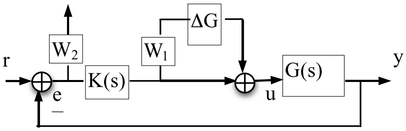


Fig. 9.3 The input multiplicative uncertainty

Nominal Performance Closed-loop system achieves nominal performance if and only if

$$\left\| W_2(I+GK)^{-1} \right\|_\infty < 1. \tag{9.18}$$

The weighting function (W_2) is used to reflect the relative importance of various frequency ranges for which performance is desired. Since W_2 is chosen to be a scalar, the maximum singular value plot of the sensitivity transfer function at every frequency must lie below the plot of $1/|W_2|$, i.e.,

$$\sigma\left[W_2(I+GK)^{-1}(j\omega)\right] < 1. \tag{9.19}$$

Robust Stability Closed-loop system achieves robust stability if and only if:

$$\left\| W_1 KG(I+KG)^{-1} \right\|_\infty < 1. \tag{9.20}$$

W_1 is the weight on multiplicative uncertainly of the system. Since W_1 is chosen to be a scalar, the maximum singular value plot of the sensitivity transfer function must lie below the plot of $1/|W_1|$ at every frequency.

$$\sigma\left[W_1 KG(I+KG)^{-1}(j\omega)\right] < 1. \tag{9.21}$$

Robust Performance The closed-loop system achieves robust performance if

$$\left\| W_2(I+G_C K)^{-1} \right\|_\infty < 1, \quad G_C = (I + W_{\mathrm{del}}\Delta G)G \tag{9.22}$$

where W_{del} is the weighting on the error signal. The robust performance is guaranteed if $\mu(M) < 1$. An interesting criterion proposed by Gagnon et al. [4] to assess the robust performance of multi-loop PID controller is given by

$$J = \min_{K(s)} \left\{ \sum_{i=1}^{m} \left| \mu_{\Delta G(s)}\left[M(s)\right] - 1 \right|^n \right\} \tag{9.23}$$

where n is an integer penalty weighting, and m represents the number of frequency points at which the criteria is evaluated. This criterion does not however penalise the actuator variations, which is often important in terms of control energy cost and actuator saturation, wear and tear and hence plant availability. The criterion is therefore modified in this study to include the control sensitivity:

$$J = \min_{K(s)} \left\{ \sum_{i=1}^{m} \left[W_M \left| \mu_{\Delta G(s)}\left[M(s)\right] - 1 \right|^2 + W_C \mu_{\Delta G(s)}\left[C(s)\right] \right] \right\} \tag{9.24}$$

where W_M and W_C are appropriate weightings.

Robust Stability The closed-loop system is stable for all perturbations $\bar{\sigma}(\Delta G) < l$ if and only if

$$\|T_{\text{cl}} l\|_\infty < 1 \quad \forall\omega \tag{9.25}$$

where $T_{\text{cl}} = GK(I + GK)^{-1}$.

9.4 Tuning of Controllers

To allow for an objective comparison of the performance achieved by the multivariable controllers investigated in this study, the tuning parameters of each of the controllers have been selected such that the following penalty function was minimised (Wahab et al. [8]):

$$J = \int_0^\infty \left[\left\{\tilde{x}(t)^{\mathrm{T}} Q \tilde{x}(t) + u(t)^{\mathrm{T}} R u(t)\right\}\right] dt. \tag{9.26}$$

The weighting matrices, Q and R, are non-negative definite symmetric matrices, tuned in such a way that satisfactory closed-loop performance is obtained. It was assumed that the process dynamics and controller states could be described using

$$\tilde{x}(t) = A\tilde{x}(t) + Bu(t), \tag{9.27}$$

$$y(t) = C\tilde{x}(t) \tag{9.28}$$

where $\tilde{x}(t) = [x(t)\ v(t)]^{\mathrm{T}}$, and where $v(k)$ denoted the controller integrator states. Under these assumptions, the multivariable PID control laws could be expressed using

$$u(t) = -K\tilde{x}(t), \tag{9.29}$$

$$u(t) = K\tilde{x}(t) = K\big(A\tilde{x}(t) + Bu(t)\big) \tag{9.30}$$

where $K = [K_c\ \ K_i]$. Then, by substituting (9.30) into (9.26) the following was obtained:

$$J = \int_0^\infty \tilde{x}(t)^{\mathrm{T}}\big(Q + K^{\mathrm{T}} R K\big)\tilde{x}(t) = \tilde{x}(t)^{\mathrm{T}} P \tilde{x}(t). \tag{9.31}$$

Then by letting

$$A_c^{\mathrm{T}} P + P A_c + Q + K^{\mathrm{T}} R K = 0 \tag{9.32}$$

the penalty function could be written as

$$J = \int_0^\infty \tilde{x}(t)^{\mathrm{T}} P \tilde{x}(t) = \tilde{x}(0)^{\mathrm{T}} P \tilde{x}(0) \tag{9.33}$$

where P denotes the solution to the discrete-time Lyapunov equation. Thus, for each of the PID control schemes, the controller parameters Θ were selected such that the matrix norm of P was minimised, i.e.,

$$\min_{K} \| P \|. \tag{9.34}$$

Therefore, the controller parameters Θ were optimal in the sense of minimising the cost function J for specific Q and R. In each of the control design cases, the above problem was solved using a numerical optimisation method. This approach is justified when the process interaction is strong and the trial-and-error tuning approach for all controller parameters would be time consuming.

9.5 Simulation Studies

The example used for testing the tuning methods is the eight-tray distillation column [9] for separating methanol and water. The model of the system is a 2 × 2 system with the following open-loop transfer function:

$$G(s) = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.6e^{-3s}}{14.4s+1} \end{bmatrix}.$$

The outputs of the model are the reflux and steam flow, and the inputs are the distillate and bottoms compositions.

Nine methods have been used to design a PI controller for the above system. The methods are:

1. Davison Method
2. Penttinen–Koivo method
3. Maciejowski method
4. The BLT method
5. Optimal control
6. Ho and Xu
7. IMC method
8. Loh et al.
9. Wang et al.

The Frequency and Time domain indices achieved for each method are shown in Tables 9.1 and 9.2. Nominal Performance, Robust Stability and Robust Performance Comparison of methods according to (9.2), (9.3), (9.5) and (9.7) have been shown in Figs. 9.4, 9.5, 9.6 and 9.7, respectively.

Fig. 9.4 Nominal performance comparison

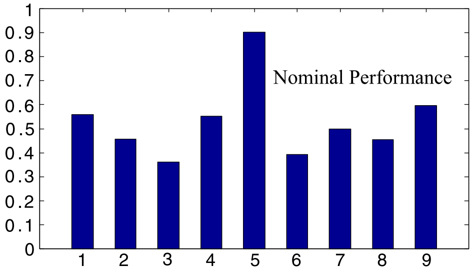


Fig. 9.5 Robust stability comparison

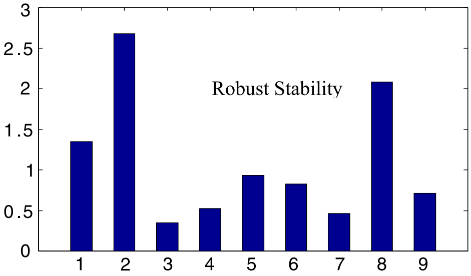


Fig. 9.6 Robust performance comparison

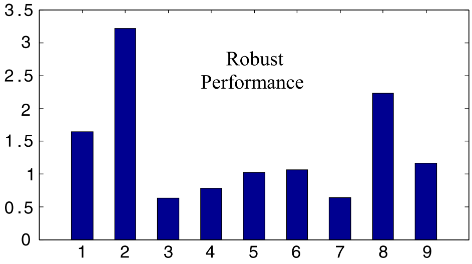
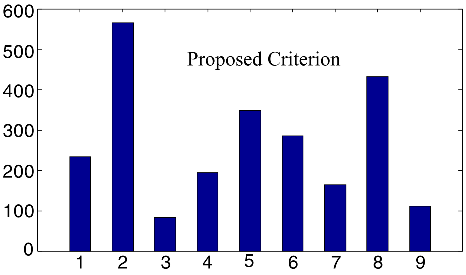


Fig. 9.7 Robust performance comparison using the proposed criterion



9.5.1 Discussion

Figure 9.4 shows that the controlled system has achieved nominal performance for all methods. This conclusion follows from the fact that maximum singular values are less than one. Methods 1, 2, 8 have not achieved robust stability, and method 3 has best stability robustness as shown in Fig. 9.5.

Table 9.1 Frequency domain specification

M	S_m	$S(j.01)$	ω_m	BW	M_m
1	11.4	−31.5	3.6	0.29–0.65	2.6
2	18.4	−25.9	10	1.2–1.3	13.7
3	04.7	−25.4	0.9	0.18–0.18	−8.9
4	06.4	−09.3	2.3	0.4–0.12	0.40
5	04.5	−19.3	2.4	0.47–0.25	−7.1
6	08.4	−16.2	2.5	0.6–0.21	−12.4
7	12.3	−04.6	3.4	0.75–0.03	1.2
8	09.9	−17.1	2.8	1.0–0.17	0.92
9	09.4	−21.8	1.1	0.35–0.18	0.33

Table 9.2 Time domain specifications

M	OS %	RT (s)	ST (s)
1	35–31	9–2.5	43–42.5
2	54–56	8–2	31–41
3	12–3.5	9–5	30–9.4
4	10–0	4.2–84	20–130
5	2.3–4	10–5	12–26
6	0–0	3–32	34–48
7	6–0	2.5–51	25–85
8	53–0	2–25	19–38
9	39–11	4–8	24–37

OS: Over Shoot
RT: Rise Time
ST: Settling Time

Figure 9.6 shows the maximum value of upper bound of $\mu(M)$. Robust performance is achieved if $\mu(M) < 1$. Therefore the methods of 1, 2, 6, 8, 9 did not achieved robust performance. This is also confirmed by the proposed criterion of (9.23), Fig. 9.7.

Frequency Domain Specifications Using Table 9.2, the frequency domain indices show that method 2 has maximum peak sensitivity and control sensitivity. Method 2 also gives the largest bandwidth.

The Step response specification of the methods has been shown in Table 9.1. The methods are compared for the performance characteristics as follow:

Settle Time Methods 5 and 3 have minimum settling time, and methods 1 and 6 have the maximum settling time.

Overshoot Methods 6 and 5 have minimum overshoot, and methods 2 and 9 have the maximum overshoot.

Rise Time Methods 2, 3 and 8 have minimum rising time, and method 4 has the maximum rising time.

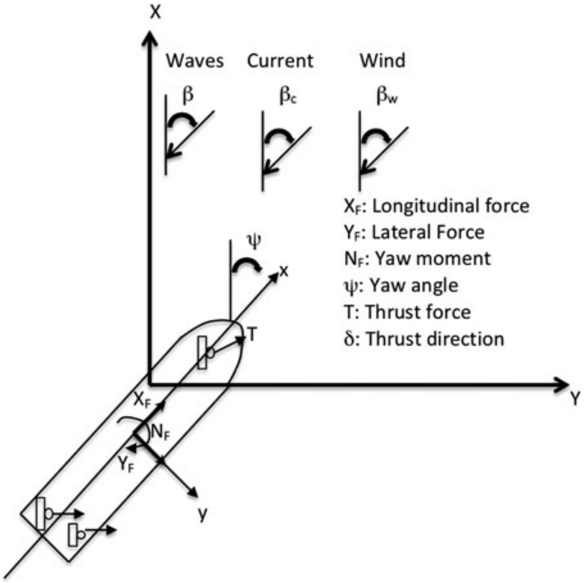


Fig. 9.8 Plan view of the ship with thrusters illustrated

Maximum Actuator Excursion Method 2 has the biggest, and method 6 has the smallest bandwidth, respectively. This is confirmed by the fact that method 2 has the largest bandwidth in open loop and therefore lowest disturbance rejection. In other words, method 6 has narrower bandwidth and consequently more disturbance rejection.

A qualitative comparison of the methods shows that methods 1, 2, 8, 9 require the least computation and coding requirements. Methods 3, 6 and 7 require a simple dynamic model of the system. Methods 4 and 5 require the most design effort and are often used in off-line applications.

The results demonstrate that a method which combines the best features of methods 1, 2 and 3 will be most suitable for industrial application as these methods require a low modelling requirement and a minimum number of tuning parameters.

9.6 Dynamic Ship Position Controller Tuning

Dynamic Positioning (DP) of a vessel (Fig. 9.8) refers to the process of automatically controlling the vessel's thrusters and/or main screws to maintain the vessel at a fixed position and heading and/or at a precise speed along a selected track [15]. Despite the success of advanced DP control systems, the use of PIDs is still widespread, particularly for the low-cost DP systems. The main reason for this popularity is the ease of tuning and commissioning PID controllers. In most systems, SISO PIDs are used. However, a ship exhibits significant interactions between the various loops, thus a true multivariable strategy is preferable.

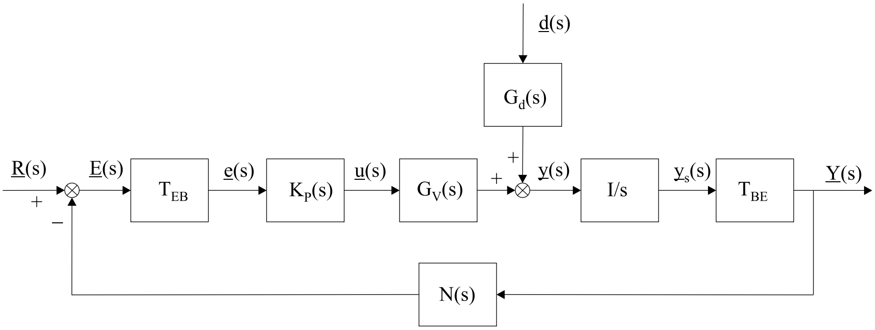


Fig. 9.9 Control scheme with output notch filter

The forces incident on a vessel due to the environment can be considerably greater than the available force from the ship's thrusters. Thus, it is necessary to both avoid actuator saturation and to save fuel by ignoring disturbances, which cannot be effectively cancelled. To this end, filters can be employed to attenuate non-essential components of the spectrum of measured variables. To be specific, notch filtering of position measurement at the frequency of dominant wave exciting forces, $\omega_n = 0.6$ rad/s, produces good steady-state tracking, whilst removing high-frequency zero-mean forces that do not affect the average position of the ship.

A typical block diagram of the control system is shown in Fig. 9.9. The closed-loop ship control has the transfer function

$$\begin{aligned}\underline{Y}(s) &= (sI + T_{BE}G_vK_pT_{EB}N)^{-1}T_{BE}G_vK_pT_{EB}\underline{R}(s)\\ &\quad + (sI + T_{BE}G_vK_pT_{EB}N)^{-1}T_{BE}G_d\underline{d}(s). \qquad (9.35)\end{aligned}$$

However, noting that $T_{BE}T_{EB} = I$ and performing suitable block diagram manipulations, the simplified equivalent closed-loop system may be expressed as

$$\underline{Y}(s) = T_{BE}(sI + G_vK_pN)^{-1}G_vK_p\underline{R}(s) + T_{BE}(sI + G_vK_pN)^{-1}G_d\underline{d}(s). \quad (9.36)$$

The coordinate transformations do not appear within this equivalent closed-loop system as $T_{BE}T_{EB} = I$ and it is clear that they do not affect the dynamics, which are controlled.

9.6.1 Position and Velocity Feedback

To stabilise the ship, it is necessary to feedback the velocity to increase the closed-loop damping. Hence, a full multivariable PID is needed to meet the stability and performance criteria. If the velocity gain $K_V(s)$ is sufficiently large, then the inner loop can provide enough phase margin such that notch filters can be introduced while closed-position-loop performance and stability are maintained. Much lower gains in the position loop set the bandwidth of the position control and keep actuator activity to acceptable levels. The closed-loop transfer functions of the velocity loop are:

$$\underline{y}_s(s) = T_V(s)\underline{r}_V(s) + S_V(s)G_d(s)\underline{d}(s), \tag{9.37}$$

$$\underline{u}_s(s) = U_V(s)\underline{r}_V(s) - V_V(s)G_d(s)\underline{d}(s), \tag{9.38}$$

where

$$S_V(s) = \frac{I}{s}(I + G_V K_V N)^{-1}, \tag{9.39}$$

$$T_V(s) = \frac{I}{s}(I + G_V K_V N)^{-1} G_V K_V, \tag{9.40}$$

$$U_V(s) = \frac{I}{s}(I + K_V N G_V)^{-1} K_V, \tag{9.41}$$

$$US_V(s) = \frac{I}{s}(I + K_V N G_V)^{-1} K_V N. \tag{9.42}$$

In the position loop, the closed-loop transfer functions of note are:

$$\underline{Y}(s) = T_p(s)\underline{R}(s) + S_p(s)G_d(s)\underline{d}(s), \tag{9.43}$$

$$\underline{u}(s) = U_p(s)\underline{R}(s) - V_p(s)G_d(s)\underline{d}(s), \tag{9.44}$$

where

$$S_p(s) = T_{BE}(I + T_V K_P N)^{-1} S_V, \tag{9.45}$$

$$T_p(s) = T_{BE}(I + T_V K_P N)^{-1} S_V K_P T_{EB}, \tag{9.46}$$

$$U_p(s) = (sI + U_V K_P N G_V)^{-1} s U_V K_P T_{EB}, \tag{9.47}$$

$$V_p(s) = (sI + U_V K_P N G_V)^{-1}(U_V K_P N + sV_V). \tag{9.48}$$

The design trade-offs for this system are interesting. When selecting feedback gains for both loops, characteristics such as speed of response, disturbance rejection, decoupling, and actuator constraints must all be taken into account. High gains may produce fast responses and good disturbance rejection but with unacceptable actuator activity. Reducing the gain may bring thruster forces to within the constraints, but the system will respond slowly and be pushed away from the set point more easily. The problem of coupling may also appear whereby control energy is expended on one input to cancel the effect of another control input. This interaction is clearly undesirable and gives an inefficient overall control system, but may have to be tolerated to some extent given other possibly opposing design issues. The velocity loops are assumed to be tuned, and hence the position loops will be discussed here. For details of velocity loop tuning, see Martin and Katebi [15].

As discussed in Sect. 9.5, the following methods are most appropriate for online tuning, and hence they are applied to the ship positioning control problem.

9.6.2 Davison Method

Moving to the position loop control, it is evident that the Davison method will produce zero gains, due to the integrators in $T_V(s)$ (9.40). Therefore, the method is of no value in this case either.

9.6.3 Penttinen–Koivo Method

The Penttinen–Koivo method presents another difficulty because there is no readily available state-space representation for $T_V(s)$ (9.40). Hence, the method of identifying CB from step tests is utilised. However, when the Penttinen–Koivo, Maciejowski or Martin–Katebi design methods are used in the velocity loop, the result is $CB = 0$. This result is confirmed by inspection of $T_V(s)$, because every transfer function element has relative degree between numerator and denominator of at least 2. Therefore, a step appearing at any input will not instantaneously produce an output. The inverse of CB is infinite in this case, and the Penttinen–Koivo method is of no further interest.

9.6.4 Maciejowski Method

The initial selection of bandwidth is 0.02 rad/s, because reference filters with this bandwidth are to be employed in order to keep actuator excursion low and to give a smooth transition from one position reference to another. Therefore, there is little point in designing a faster ship controller than this. The bandwidth selected will be sufficient to reject any low-frequency disturbances encountered.

$$K = \begin{bmatrix} 0.0199 & 0 & 0 \\ 0 & 0.0198 & -0.00125 \\ 0 & 0.000000652 & 0.0185 \end{bmatrix} = \big(T_V(j0.2)N(j0.02)\big)^{-1} \tag{9.49}$$

where

$$K_P(s) = K_p + \frac{K_i}{s} + K_d s = p_P K + \varepsilon_P K \frac{1}{s} + \delta_P K s, \tag{9.50}$$

$$T_V(j0.02)N(j0.02)K$$
$$= \begin{bmatrix} -0.0144 - 0.999j & 0 & 0 \\ 0 & -0.0143 - 0.999j & -0.254 + 0.00362j \\ 0 & 0.000135 - 0.00000133j & -0.00927 - 0.931j \end{bmatrix}. \tag{9.51}$$

The system is clearly well decoupled at this frequency, and once again, the Characteristic Loci are plotted to provide stable values for p_P and ε_P. When $p_P = 1$, the integral action scaling may take values $0 < \varepsilon_P < 0.91$.

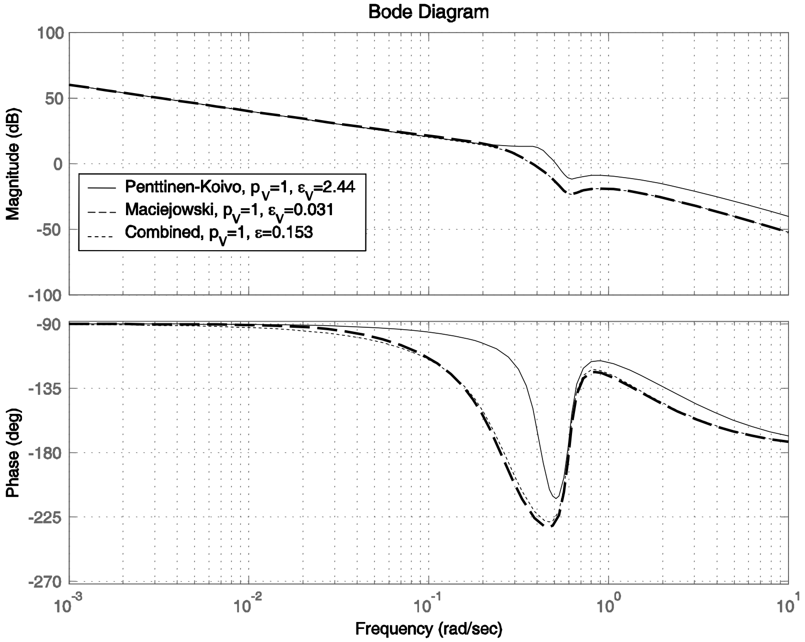


Fig. 9.10 Disturbance-to-control-input Bode plot

The Bode plot, in Fig. 9.10, demonstrates the presence of a notch in the frequency response from wave disturbances to the thruster input at 0.6 rad/s. The three plots are for different tuning methods in the velocity loop and with the gains indicated. Similar plots may be obtained for the sway and yaw loops.

9.6.5 Martin–Katebi Method

As with the Davison method, the K_i gains will be zero, hence this is just a special case of the Maciejowski method where $\varepsilon_P = 0$. For each position loop controller, $\delta_P = 0$ to avoid large thruster forces, and ε_P is chosen to be 10% of the upper limit of stable values as a first guess for controller tuning. The final values of p_P and ε_P will depend on stability; control input magnitude, overshoot, settling time and decoupling as given in the next section on simulation.

9.6.6 Controller Tuning and Simulation

In this section the performance of the three different control design schemes are evaluated on a nonlinear model using Simulink. Bearing in mind that the Davison method was not applicable to the system under investigation, there are three permutations of controller—a Penttinen–Koivo, Maciejowski or Martin–Katebi velocity

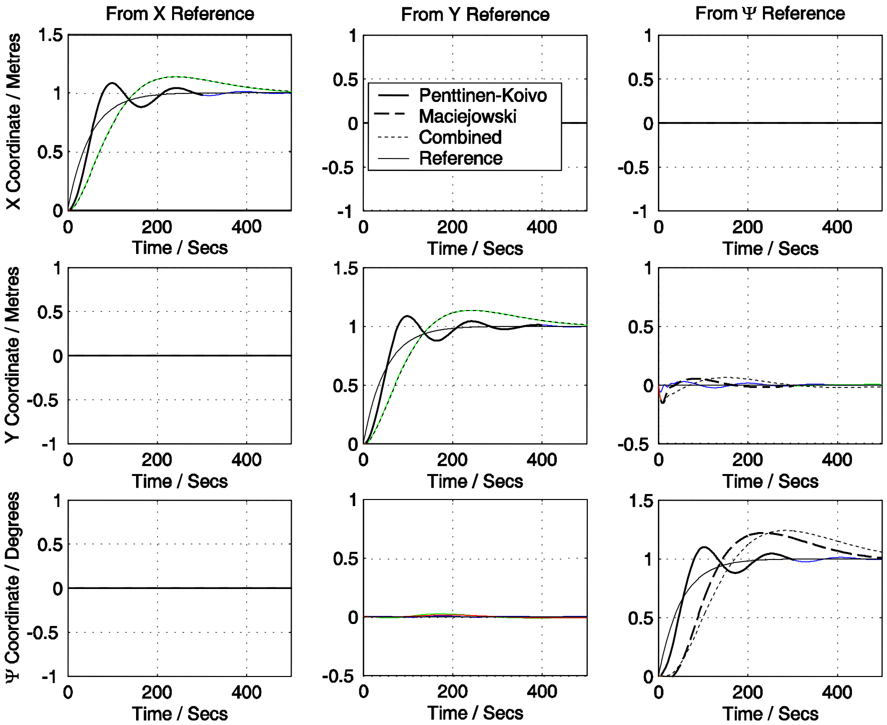


Fig. 9.11 Responses of $T_P(s)F_R(s)$ to filtered step reference demand

loop design with a Maciejowski position loop design. The results and subsequent tuning of each control design now follow.

9.6.7 Transfer Function Responses

To make a quick initial assessment of each controller, responses from (9.37) to (9.42) are required. In Sect. 9.6.4, it was noted that reference filters of bandwidth 0.02 rad/s are to be employed to keep actuator excursion low and to give a smooth transition from one position reference to another. Figure 9.11 shows the multivariable responses of $T_P(s)V_R(s)$ to unit step reference demands on each input, with zero disturbances. For the Pentinnen–Koivo velocity loop case, the parameters are $p_V = 1$, $\varepsilon_V = 2.44$, $p_P = 1$, $\varepsilon_P = 0.091$. In the Maciejowski velocity loop case, the values are $p_V = 1$, $\varepsilon_V = 0.031$, $p_P = 1$, $\varepsilon_P = 0.008$, and in the Martin–Katebi case, $p_V = 1$, $\varepsilon_V = 0.153$, $p_P = 1$, $\varepsilon_P = 0.0073$.

Cross coupling is not a problem, judging from the Y response to a step in ψ and vice versa. However, the diagonal elements exhibit considerable overshoot and long settling times for all three cases, and oscillation in the Penttinen–Koivo case. Increasing p_P in fact improves the responses in all three cases. Letting $p_P = 3$ and al-

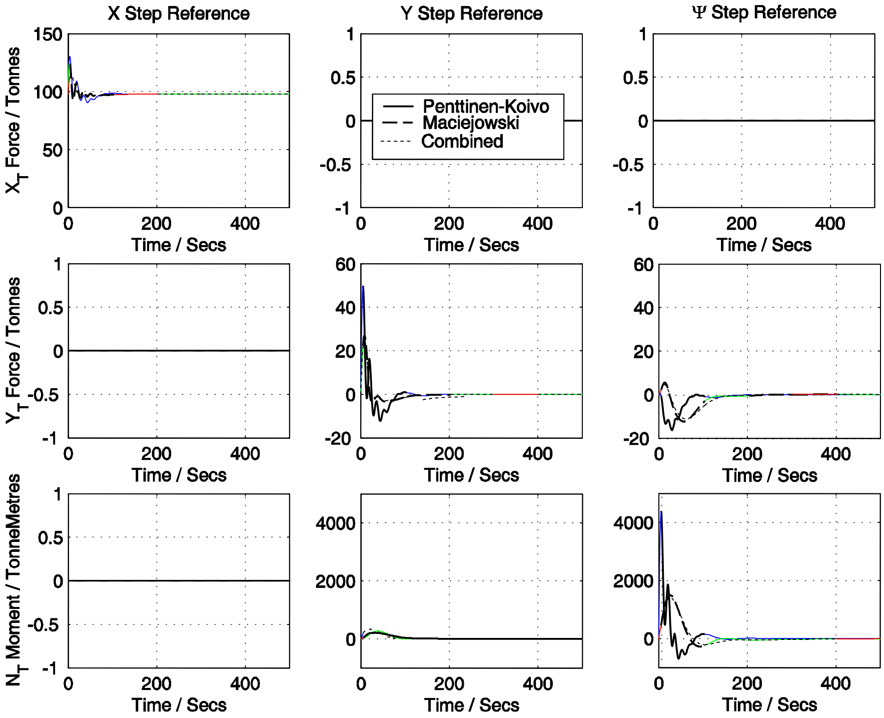


Fig. 9.12 Thruster responses of $U_P(s)F_R(s)$ to step reference demand

tering ε_P to the new 10% value of the upper limit of stability, overshoot and settling times are now reduced, and the performance is acceptable for this application. Figure 9.11 shows the corresponding surge and sway thrust responses of $U_P(s)F_R(s)$ with the parameters given, and a bias of 98 T added to the surge plot to account for the operating point. In the nonlinear simulation, there would also be a slight offset on the right-hand plots due to the current acting at one degree to the ship heading when the reference point is reached. All peaks are well within the thruster maximum force of 470 T and moment of 10^5 T m, and the rate limit of 50 T/s is not violated.

In the simulation later, the average wind velocity U_{A0} is set to 20 m/s at an angle $\beta_A = \psi_R + \pi/4$, and the waves are for heavy seas at an angle $\beta_W = \psi_R + \pi/12$, where ψ_R is the heading reference.

Comparing the magnitude of these forces and moments with the thruster responses of $V_P(s)G_d(s)$ in Fig. 9.12, it is clear that the notch filter and bandwidth selection prevents total rejection of the disturbance, instead focusing on the low-frequency components.

The high-frequency zero-mean disturbances are allowed to influence the ship motion, as depicted in the responses of $S_P(s)G_d(s)$ in Fig. 9.13.

The surge thrust reaches peaks of greater than 470 T in the Penttinen–Koivo linear responses and the rate of change peaks at 200 T/s, so clearly some tuning is

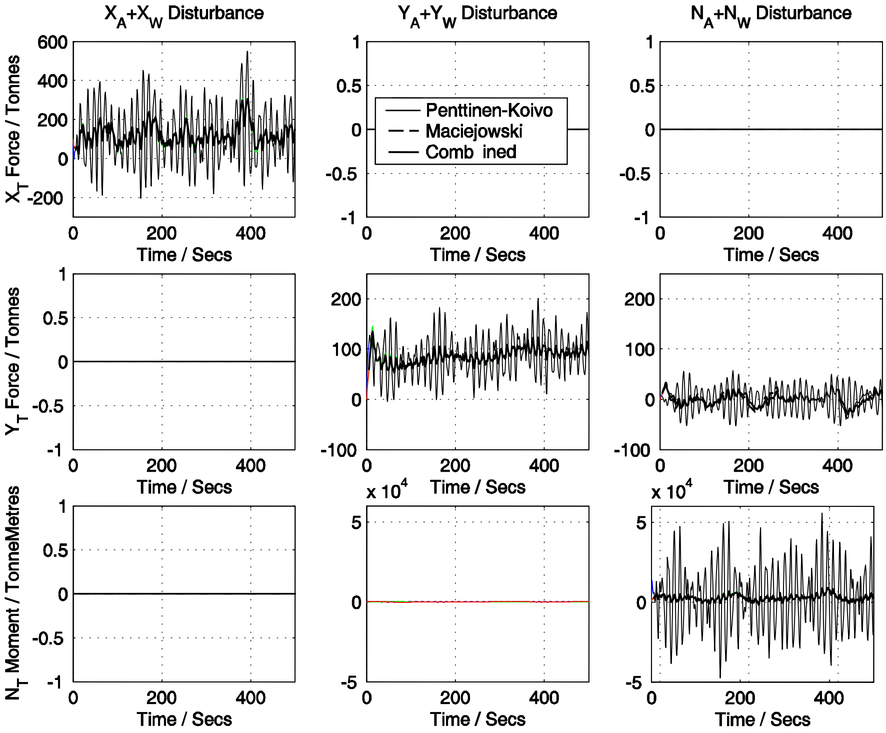


Fig. 9.13 Thruster responses of $V_P(s)G_d(s)$ to disturbances

required. Unfortunately, simply reducing pP does not reduce the thruster magnitude as desired.

As mentioned in [15], adjusting $K_V(s)$ affects the spectral characteristics of the position loop above the bandwidth frequency. To proceed, p_V must be reduced in the velocity loop and a new gain matrix, K, found in the position loop. The thruster peaks are in excess of the 470 T maximum by around 15%, but the rate of change is four times larger than the acceptable 50 T/s. Decreasing p_V from 1 to 0.3, ε_V from 2.44 to 0.155 and recalculating K at 0.02 rad/s for the Penttinen–Koivo velocity loop produces

$$K = \begin{bmatrix} 0.0202 & 0 & 0 \\ 0 & 0.0201 & -0.094 \\ 0 & 0.00000479 & 0.0197 \end{bmatrix}. \tag{9.52}$$

By comparison with (9.38), the off-diagonal elements show an increase due to greater velocity loop interaction between the sway and yaw directions. Hence, thruster forces are likely to increase slightly so that one input can cancel the effect of the other control input (see Fig. 9.14). The reduction of p_V should produce smaller thruster forces overall, however. The attenuation of frequencies above 0.2 rad/s is now much greater and appears to be similar to the responses in the Maciejowski and Martin–Katebi cases.

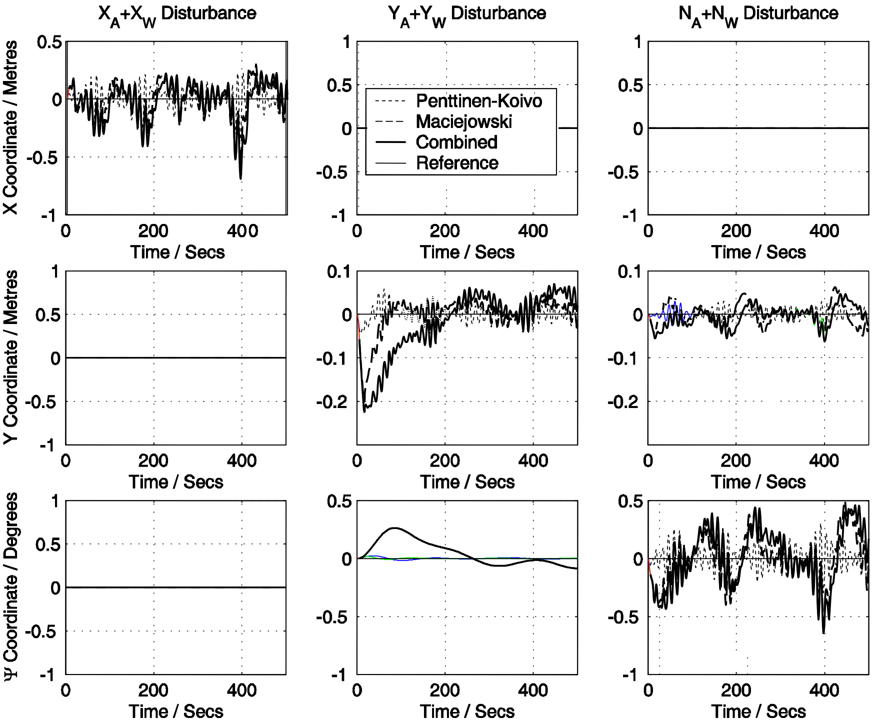


Fig. 9.14 Position responses of $S_P(s)G_d(s)$ to disturbances

9.7 Conclusions

In this chapter, nine PID tuning methods for multivariable systems were compared for their robustness properties in this chapter. The Gain and phase margin method was shown to outperform other methods. The Penttinen–Koivo method shows the poorest performance in tracking error. The BLT shows poor performance in tracking error but has good disturbance rejection. The feedback relay method performed well for disturbance rejection with poor behaviour in tracking error. More examples and robustness of methods should be investigated to establish a better conclusive comparison of the methods.

This was followed by application and comparison of nonparametric tuning methods for an industrial example. The study demonstrated that the proposed method can provide better performance and incorporates most of the design features available in the other methods.

The main disadvantage of the nonparametric methods described here is that the open-loop system should be stable. Hence, there is still a need for research and development of more general multivariable tuning methods to deal with unstable systems as well as systems with strong loop interactions.

Acknowledgements I would like to express my thanks and gratitude to all my students and colleagues who carried out the work reported here, in particular, Professor Michael Johnson, Dr. Peter Martin, Dr. Norhalisa Wahab, Dr. Jonas Balderud, Dr. M. Moradi and Mr. Mahdi Elkawafi.

References

1. Äström, K.J., Hägglund, T.: PID Controllers: Theory, Design and Tuning, 2nd edn. Instrument Society of America (1995)
2. Davison, E.: Multivariable tuning regulators: The feedforward and robust control of a general servomechanism problem. IEEE Trans. Autom. Control **21**(1), 35–47 (1976)
3. Dong, J., Brosilow, C.B.: Design of robust multi-variable PID controllers via IMC. In: Proc. of the American Control Conference, pp. 3380–3384 (1997)
4. Gagnon, E., Pomerleau, A., Desbiens, A.: Mu-synthesis of robust decentralised PI controllers. IEE Proc., Control Theory Appl. **146**(4), 289–294 (1999)
5. Garcia, C.E., Morari, M.: Internal model control. 1. A unifying review and some new results. In: Industrial and Engineering Chemical Process Design and Development (1982)
6. Hang, C.C., Tay, T.T., Vasnani, V.U.: Auto-tuning of multivariable de-coupling controllers. Report CI-90-6 (1990)
7. Ho, W.K., Xu, W.: Multivariable PID controllers design based on the direct Nyquist array method. In: Proc. American Control Conference, pp. 3524–3528 (1998)
8. Katebi, M.R., Grimble, M.J., Zhang, Y.: H-infinity robust control designs for dynamic ship position. Proc. IEEE **144**(2), 110–120 (1997)
9. Katebi, M.R., Moradi, M.H., Johnson, M.A.: A comparison of stability and performance robustness of multivariable PID tuning methods. In: Proceedings of PID 2000 Conference, Terrassa, Spain (2000)
10. Koivo, H.N., Tanttu, J.T.: Tuning of PID controllers: Survey of SISO and MIMO techniques. In: Proceedings of the IFAC Intelligent Tuning and Adaptive Control Symposium, Singapore, pp. 75–81 (1991)
11. Lieslehto, J., Koivo, H.N.: An expert system for interaction analysis of multivariable systems. In: Proceedings of the 1987 American Control Conference, Minneapolis, MN, USA, pp. 961–962 (1987)
12. Loh, A.P., Hang, C.C., Quek, C.K., Vasnani, V.U.: Autotuning of multiloop proportional-integral controllers using relay feedback. Ind. Eng. Chem. Res. **32**, 1102–1107 (1993)
13. Luyben, W.: Process Modelling, Simulation and Control for Chemical Engineers. McGraw-Hill, New York (1990)
14. Maciejowski, J.M.: Multivariable Feedback Design. Addison-Wesley, Wokingham (1989)
15. Martin, P., Katebi, M.R.: Multivariable PID tuning for dynamic ship positioning. J. Mar. Des. Oper. **7**, 11–24 (2005)
16. Menani, S., Koivo, H.N.: Relay tuning of multivariable controllers. In: IFAC, 13th Triennial World Congress, pp. 139–144 (1996)
17. Niederlinski, A.: A heuristic approach to the design of linear multivariable interacting control system. Automatica **7**, 691–701 (1971)
18. Palmor, Z.J., Halevi, Y., Efrati, T.: A general and exact method for determining limit cycles in decentralized relay systems. Automatica **31**(9), 1333–1339 (1995)
19. Penttinen, J., Koivo, H.N.: Multivariable tuning regulators for unknown systems. Automatica **16**, 393–398 (1980)
20. Rivera, D.E., Morari, M., Skogestad, S.: Internal model control. 4. PID controller design. Ind. Eng. Chem. Process Des. Dev. **25**, 252–265 (1986)
21. Rosenbrok, H.H.: Computer Aided Control System Design. Academic Press, New York (1974)
22. Sandelin, P.M., Toivonen, H.T., Osters, M., Waller, K.V.: Robust multi-objective linear quadratic control of distillation using low-order controllers. Chem. Eng. Sci. **46**(11), 2815–2827 (1991)

23. Skofestad, S., Lundström, P., Jacobsen, E.W.: Selecting the best distillation control configuration. AIChE J. **36**, 753–764 (1990)
24. Wahab, N.A., Katebi, M.R., Balderud, J.: Multivariable PID control design for activated sludge process with nitrification and denitrification. Biochem. Eng. J. **42**, 239–248 (2009)
25. Wang, F., Wu, T.Y.: Multiloop PID controllers tuning by goal attainment trade-off method. Trans. Inst. Meas. Control **17**(1), 27–34 (1995)
26. Wang, Q., Zou, B., Lee, T., Bi, Q.: Auto-tuning of multivariable PID controllers from decentralized relay feedback. Automatica **33**(3), 319–330 (1997)
27. Yanchevsky, A.E.: Optimal decentralised controller design for multivariable plants. Int. J. Syst. Sci. **18**(1), 177–187 (1987)
28. Yu, C.C.: Autotuning of PID Controllers. Advances in Industrial Control. Springer, London (1999)
29. Yusof, R., Omatu, S., Khalid, M.: Self-tuning PID control: A multivariable derivation and application. Automatica **30**(12), 1975–1981 (1994)
30. Zhuang, M., Atherton, D.P.: PID controller design for a TITO system. IEE Proc., Control Theory Appl. **141**(2), 111–120 (1994)

Part III
Issues in PID Control Systems

Chapter 10
Identification for PID Control

Kostas S. Tsakalis and Sachi Dash

10.1 Introduction

Proportional-Integral-Derivative (PID) controllers are still the most common control algorithms used in the process industry. PID is a reduced complexity controller that provides the bare essentials for control: an integrator for low-frequency disturbance attenuation, and two-zeros-worth of phase lead for stabilization and phase margin adjustment. Practical experience shows that its tuning can be accomplished with very little information about the plant from the point of view of standard design techniques. This celebrated feature carefully underplays the fact that as a limited degree of freedom controller, the PID may be unsuccessful in controlling arbitrary plants and the tuning techniques may become progressively more complicated as the class of plants is expanded. For example, it is quite straightforward to show that PID controllers can always stabilize a single integrator and that they cannot stabilize a chain of three integrators. Furthermore, techniques that have a well established track record for the typical process control application, e.g., a heating process, are shown to fail when the plant contains flexible modes, e.g., a pendulum with flexible shaft. It is in these cases where PID control must ultimately escape the back-of-the-envelope calculations and the nearly-model-free framework. The most popular controller can now utilize the most recent computational methods and design understanding. The benefits are in the hardware simplicity, including anti-windups and fast execution times, and ease of scheduling.

Many methods for PID tuning are found in the literature, and [1] is an excellent introduction. Typical examples include an analytic derivation of the tuning, based on

K.S. Tsakalis (✉)
School of Electrical Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287-5706, USA
e-mail: tsakalis@asu.edu

S. Dash
Honeywell Technology Center, 3660 Technology Drive, Minneapolis, MN 55418, USA
e-mail: Sachi.Dash@honeywell.com

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_10, © Springer-Verlag London Limited 2012

a low-order model of the plant (e.g., [2]). In other methods [3–5], the tuning is based on the optimization of some performance measure related to the characteristics of the frequency and/or time response of the system. Evaluations and comparisons of PID tuning schemes are always a topic of research and publications [6, 7]. More recently, considerable effort has been devoted to the development of methods that connect the identification properties with the controller design to produce more reliable control systems [8–12]. For PID tuning, examples of work in this direction are [13–15].

To address the feasibility of PID control, we may focus on the plant and view PID controllers as pole-placement controllers for "arbitrary" second order systems. We can then ask when our plant is sufficiently close to a second order system. An alternative to this is to ask when the PID is an optimal controller [16]. The other approach is to focus on the controller and ask when a full order controller can be approximated sufficiently close by a PID. In both cases, a simple small-gain argument quickly shows that any approximation error must be weighted by the target loop properties. Such designs can be resolved with some degree of conservatism by means of efficient (convex) optimization techniques [14, 17] or non-conservative but non-convex optimization [18], with all the associated difficulties and shortcomings. Yet, the tuning of PID is always a subject of great practical interest and, despite the small number tunable parameters, remains as a research topic. One reason for this is the existence of several performance trade-offs (e.g., bandwidth, integral action, overshoot, robustness), which may obscure the proper choice of the control objective. More severe implementation problems arise from the requirements of some user expertise for successful tuning, in conjunction with the large number of PID loops that need to be tuned. For example, in a typical refinery there could be 3000 PID loops. As a result, it is not surprising that many of these controllers are poorly tuned ($\sim$30%, [19]).

What is perhaps obscure and deserves more attention here is the importance of the objective. Both of the seemingly straightforward approximations (plant by a low-order model or controller by a PID) hinge on the proper choice of a closed loop objective or a closed loop target. Understanding the basic properties and limitations of these targets is an important step towards consistently successful PID implementations. From an optimization point of view, using the PID as a reduced degree-of-freedom controller will in general result in multiple optima. While certain objectives may still lead to unique solutions, the sensitivity of the solution to perturbations becomes important when model uncertainty is included in the optimization. This is of particular practical significance when the model is obtained from a system identification step, which will invariably involve model sets that cannot fully describe the data, e.g., data containing disturbances, or parametric models that can only capture some of the system modes. In such cases, the optimization of the PID tuning for the nominal model becomes a lesser problem and gives its place to the robustness of the PID with respect to the model uncertainty and the trade-offs among the various performance measures.

One particular design point of view that has emerged and has been well-received in applications [20] is the delivery of multiple tunings, e.g., on a Pareto optimal curve, that allow the user to select among different performance-robustness metrics. A typical implementation of this idea involves high-level tuning parameters, such as closed-loop bandwidth, damping, etc., and the use of simulation to visualize the performance of the tuning with the identified model. Here, it is essential to present the user with choices that represent different Pareto-optimal trade-offs but avoid the unintended optimization of a particular feature, say the set-point step response. This approach brings in some new dimensions in the PID tuning process, such as varying trade-offs that may involve operator experience or changing assessment of risk (and, therefore, robustness margin), behavior with saturating actuators and effectiveness of the employed antiwindup scheme [1], and the ability to address simultaneous stabilization of multiple models [21] with, possibly, different objectives for each one. Thus, the back-of-the-envelope calculation and the single optimal tuning are becoming only intermediate steps, or tools, in the tuning process. Furthermore, the utilization of uncertainty estimates in the PID tuning procedure allows for a quantitative assessment of risk that improves the reliability of the controller and minimizes the need for retuning or trial and error.

In this context, system identification techniques offer the ability to tune PID controllers from data, with ever-increasing degrees of sophistication. The identified models must and have progressed from a minimal information of a single frequency point, accompanied with strong assumptions or prior knowledge about the plant to guarantee that this information is all that is needed to tune the PID. The identification now is expected to provide one or more nominal models and a quantitative description of uncertainty. While in some cases a frequency response is sufficient, state-space or transfer function models are essential to handle more difficult cases that require an intermediate full-order controller design step before the reduction to a PID. There are two subtle points associated with this approach, and both emanate from the difficulty of selecting a suitable excitation for the identification experiment. The first is the need to maintain a reasonable model structure at both low and high frequencies, in addition to the usual crossover information required by the typical PID tuner. For example, model overparametrization of plants with slow poles or instabilities often leads to near-pole-zero cancellations of very slow modes that attempt to model slow drifts from small disturbances. While this is essentially irrelevant for crossover-based tuning methods, it prohibits the computation of a general stabilizing controller as an intermediate step to assess the limits of performance. Avoiding such pathologies can be performed easily in a suboptimal, non-automated fashion, e.g., either through a slow–fast model decomposition and reduction, or through the identification of a lower order model. The second point is that most of the standard identification methods exhibit a strong, albeit natural, dependence of the optimum on the excitation spectrum. This is a direct consequence of the data being outside the model set, i.e., the input–output responses that can be described by the assumed family of parametric models, say, second order or third order transfer functions. The situation is exacerbated by models with non-invertible elements (right-half plane poles and zeros). A solution of this general problem may be achieved as an iteration in identification and design of excitation, e.g., [22, 23], also found in the so-called "windsurfer approach" to adaptive control

[24, 25]. A alternative approach is to use min–max techniques to perform identification in $\mathscr{L}_\infty/\mathscr{H}_\infty$ which normalize the effect of excitation. The difference of the two approaches is between the identification using shorter data sets from multiple experiments versus the identification of the model from a longer data set and a single experiment.

In the following, we discuss some of the basic PID tuning principles from the perspective of system modeling and identification. It is not meant to be a sharp distinction since, as we argued above, both tuning and target/objective selection are not independent of the identification method. A variety of approaches have been taken through the years to quantify the necessary model features to tune PID controllers. We distinguish three cases with their representatives:

1. *Ziegler–Nichols and classical one-frequency point methods*: They assume a simple model structure, e.g., a first order plus dead-time approximation (FOPDT), for which an optimal PID tuning has been tabulated in terms of two free parameters [26]. As such, it is suitable for process control applications and back-of-the-envelope calculations. It comes in two flavors, open and closed loop data, with the closed loop being naturally weighted with closed-loop objectives and often being more successful. The tuning rules focus on disturbance attenuation, and may not provide flexibility to cope with other types of models or different levels of uncertainty in the model. But the value of this pioneering technique for the intended class of applications is unquestionable, with a high point reached by Åström's ingenious autotuner [27]. The common characteristics of these two cases are the ease of experimental data collection and the computation of the PID gains. Also common is the difficulty to assess a priori the success of the PID controller.
2. *Indirect identification and controller design*: The indirect terminology here comes from adaptive control. That is, the plant is identified in open or closed loop and the controller is then tuned for the identified model. It allows for the possibility of tuning a general controller before reducing it to a PID. While this provides ample generality and flexibility, its limitations are in the selection of the a priori unknown optimal identification weights. The choice of the control objective is conceptually less restrictive but practically still unclear. Our preference is with loop-shaping methods that have an immediate connection with uncertainty-imposed constraints. The reduction to the PID is a redeeming quality of this approach. For a general class of frequency-domain objectives, the linear-in-the-parameters PID structure reduces the tuning problem to a convex approximation problem and, hence, it is easily and reliably solvable. Nevertheless, we acknowledge a great variety of other objectives, typically of Linear Quadratic Regulator (LQR) or Model Predictive Control (MPC) type that may also be useful in PID controller tuning.
3. *Direct identification of the PID controller parameters*: This approach relies on a brilliant observation, due to G. Stein, that the loop transfer function can be directly optimized to approximate a prescribed target, based on input-output data alone [28, 29]. It inherently makes use of a natural frequency weighting by the target loop sensitivity and can be used independent of input signal saturation, but

requires the a priori knowledge of a nearly feasible target loop. Another approach that falls in this category is the PID tuning using the so-called unfalsified control concept [30]. Unfalsification and cost-based controller switching have produced a long list of very interesting theory and applications, e.g., see [31–35], but they extend beyond the scope of this study.

The last two methods can be stated in terms of the small gain theorem so that they can provide some closed loop performance guarantees (as sufficient conditions). They can be further enhanced by system-relevant identification optimization.

10.2 Ziegler–Nichols and One-Frequency Point Methods

Classical controller design techniques on lead-lag compensators can yield PID tunings based on magnitude and frequency information at one frequency point, namely the loop transfer function crossover frequency. One fairly successful approach is to design the compensator to achieve a given phase margin at the gain crossover frequency. Assuming that the plant transfer function $P(s)$ is well behaved around the intended crossover and rolls-off without resonances or other large increases, achieving a positive phase margin would also imply the closed-loop stability by virtue of the Nyquist theorem.

The PID tuning equations for such a design are

$$\begin{aligned}&\angle P(j\omega_c)C(j\omega_c) \geq -180 + \text{P.M.},\\ &|P(j\omega_c)C(j\omega_c)| = 1.\end{aligned}$$

These are two equations for the three PID parameters that can be solved sequentially, given the phase margin P.M. and the gain crossover frequency ω_c, for the PID free parameters K_p, K_i, K_d where

$$C(s) = \left(K_p + \frac{K_i}{s} + \frac{K_d s}{\tau s + 1}\right).$$

In the above PID definition, we included a pseudo-differentiator filter with time constant τ to limit the PID gain at high frequencies. Typically, τ is selected an order of magnitude higher than the crossover so that its phase lag contribution is limited to under 6°. The traditional PID description with $\tau = 0$ has the advantage that the proportional part contributes only a real correction while the integral and derivative parts contribute a pure imaginary correction, which facilitate the computations. The rest of the equations for the PID tuning come from the reasonable constraints that the PID should be minimum phase and that both zeros—if a derivative part is used—should occur at the same point. A quick example of such a design is when the plant is an integrator

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

to be controlled to a crossover $\omega_c = 1$ with $P.M. = 45°$. It follows that the following PI is sufficient to meet these objectives

$$C(s) = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}s}.$$

The Ziegler–Nichols method is an excellent example where the generality of this design procedure is sacrificed for a quick experimental approach to determine a suitable (phase) crossover frequency and the plant transfer function at that point. It is an early manifestation of the importance of not only the tuning method but of the ease and reliability of extracting the necessary model information for it. In a similar vein, one can consider other typical system parametrizations, e.g., by the step response, and convert the parameters into one frequency point information, thereby generating new classes of tuning rules from easily identified system characteristics. Thus, Ziegler–Nichols and other comparable methods have tabulated the results for models that typically arise in process control (e.g., FOPDT) with emphasis placed on input disturbance rejection, e.g., see [1, 6]. But, for the simple example of the integrator, Ziegler–Nichols depends on the ubiquitous high order dynamics and delays to find the phase crossover frequency and from that tune the PID. It is a method that is well-suited to an experimental application instead of a more abstract nominal model.

At this point, we should also mention the ingenious method by Åström using a relay feedback experiment to find the necessary crossover information [27]. In this case, using a describing function argument, e.g., [36], one can show the system will enter a limit cycle with frequency approximately the phase crossover and amplitude $|P(j\omega_c)|\frac{4}{\pi}$, for which the Ziegler–Nichols tuning can be applied. While fundamentally obtaining the same information, this approach avoids the drawback of the original Ziegler–Nichols procedure where by the loop was destabilized with possibly catastrophic consequences.

On the other hand, a limitation of this (and similar) methods is that it provides no tuning knobs and, if it fails, it provides no direction for a correction, other than decreasing the system bandwidth for stable plants. One such example is the inverted pendulum with flexible shaft which requires the bandwidth to be high-enough for stabilization but low-enough to avoid the excitation of the flexible modes. Such an example is considered below to show that it is possible, and often advantageous, to use a general full order controller design method and reduce it afterwards to fit the PID structure. At a first glance, this approach may seem as an overkill, i.e., to enter in a full order controller design quest before ultimately reducing it to a PID. However, with the general theoretical and computational tools that have been developed in the last two decades, it is now much easier and more reliably automated to use such general methods to obtain a stabilizing controller that obey uncertainty constraints. The rationale for such an increase in complexity is in the ability to provide a more systematic and reliable approach to modeling and PID tuning that provides a final controller with high confidence, and possibly requiring no trial-and-error. With this in mind, we provide a quick review of the frequency loop-shaping method, using a variety of industrial and educational control problems as motivating applications.

10.3 Loop-Shaping and the Use of General Controller Design Methods in PID Tuning

Loop-shaping in frequency domain is a method that converts the controller design into a model matching problem, which can be elegantly solved in the general case using $\mathscr{H}_\infty$ tools [37]. Variants of this approach have been mentioned in [1, 38, 39], and in the so-called internal model control (IMC) rules for PID tuning [40]. Here, we focus on the convex optimization method proposed by [14, 41] because it plays an instrumental role by allowing the incorporation of identification-induced constraints in the PID tuning.

10.3.1 Frequency Loop-Shaping Tuning Algorithm

It is assumed that a frequency response of the linear, time-invariant, SISO plant $P(s)$ is available. This can be obtained, e.g., via a system identification experiment [42], or from first-principles modeling. For such a system, a PID controller is to be designed, with transfer function

$$C(s) = \left(K_p + \frac{K_i}{s} + \frac{K_d s}{\tau s + 1} \right) \tag{10.1}$$

where K_p, K_i, and K_d, are the proportional, integral, and derivative gains, respectively. A low pass filter with an *a priori* selected time constant τ is used in the derivative term, in order to limit the susceptibility of the controller to measurement noise and/or reflect the finite-difference implementation of the differentiator. The Frequency Loop-Shaping (FLS) objective is to determine the PID gains $K = \{K_p, K_i, K_d\}$, so that the compensated open Loop Transfer Function (LTF) is close, in an $\mathscr{L}_\infty$ sense, to a desired or target LTF, say $L(s)$.[1] The main advantage of this linear parameterization is that any functional of the form $\|W(PC - L)\|_{\mathscr{L}_\infty}$, with W a weighting function, is convex in the design parameters K.

In order to gain some further insight on the translation of the proposed PID tuning into a convex optimization problem, we first analyze conditions to ensure the stability of the closed-loop. Let $\Delta = L - PC$ denote the error LTF and $S_o \equiv (1 + L)^{-1}$, the nominal sensitivity. In addition, suppose L is such that $S_o\Delta$ is stable, i.e., L and PC have the same unstable poles. A general version of this decomposition is depicted in Fig. 10.1. Rewriting the closed-loop system in terms of L and Δ, and applying the small gain theorem [43], a sufficient condition for closed-loop stability is

$$\|S_o\Delta\|_{\mathscr{H}_\infty} = \|(1 + L)^{-1}(PC - L)\|_{\mathscr{L}_\infty} < 1. \tag{10.2}$$

[1] $\|F\|_{\mathscr{L}_\infty} \equiv \sup_\omega |F(j\omega)|$. If F is stable, $\|F\|_{\mathscr{H}_\infty} = \|F\|_{\mathscr{L}_\infty}$.

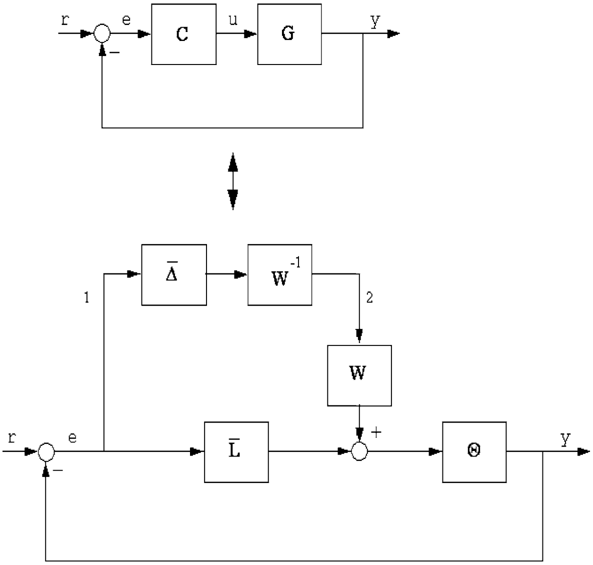


Fig. 10.1 Loop transformation to illustrate the derivation of the frequency loop shaping optimization objective. Here, Θ denotes an all-pass factor containing the unstable poles and W is a frequency weight to reduce the conservatism of the small-gain theorem. The use of Θ is to enable the conversion of the $\mathscr{H}_\infty$ norm to a $\mathscr{L}_\infty$ norm, as long as the Nyquist encirclement condition satisfied; the latter is guaranteed by assuming that $L = \bar{L}\Theta$ and CP have the same unstable poles

Expression (10.2) can be regarded as a cost functional for solving the weighted approximation problem of L by PC. The FLS tuning of the PID parameters is then cast as the following optimization problem

$$\min_K \|S_o(PC_K - L)\|_{\mathscr{L}_\infty}. \tag{10.3}$$

Notice that for internal stability *PC* should contain no right-half plane cancellations. This requirement is easily met by restricting our attention to minimum phase controllers. Such a restriction makes practical sense since the PID gains are expected to be of the same sign. Without much loss of generality, it is assumed that the PID gains are positive. All of these constraints are convex and can be easily incorporated in the optimization.

Inequality (10.2) guarantees closed-loop stability with the full-order nominal plant model. Robust stability with respect to modeling errors, however, should be handled separately. While the margin of the inequality (distance between the left- and right-hand side) has some robustness implications, a more suitable robustness measure can be assessed in terms of the target loop. For example, in a typical design the nominal plant is given together with the uncertainty in terms of sensitivity and complementary sensitivity bounds. A target loop is then selected to meet the uncertainty constraints, as well as constraints imposed by right-half plane poles and zeros. If the matching error is small, then the nominal loop is close to the target and inherits approximately the same robustness properties. Some additional consideration should be given to the multivariable case when the PIDs are designed in a decoupled fashion. In this case, the off-diagonal elements of the plant can be treated as an additional perturbation via Perron eigenvalue computations [44] or mu-analysis [45]. Because of the conservatism of the small gain theorem and the limited flexibility of the controller structure, these measures serve only as indicators to aid the target loop selection. After computing the controller parameters, the closed loop robustness properties can be evaluated more accurately against the complete nominal model and the design can be iterated, if necessary.

10.3.2 Choosing a Target LTF

The FLS tuning procedure is equally applicable for any reasonable target loop. The use of target LTFs can be motivated by classical loop-shaping ideas where considerable experience and insight are available. For example, the target loop must contain any dynamics (plant poles or zeros) that should not be canceled by the compensator. These dynamics include integrators, slow or unstable poles, time delays, and non-minimum-phase zeros near the target bandwidth. In the general case, the selection of the target loop may be quite involved, (e.g., see [38]) and may require the solution of a full order controller design problem.

Typical targets, covering most of the cases of interest, are first and second order, depending on the rolloff rate of the plant around the intended crossover frequency ω_c. For example, if the plant frequency response has approximately zero roll-off rate around ω_c, then $L = \omega_c/s$. If the plant rolls-off with -20 dB/dec around ω_c, i.e., integrators or slow poles are present, the second order target $L = \omega_c(s + a\omega_c + \varepsilon)/s(s + \varepsilon)$. Here, $-\varepsilon$ is the slow pole of the plant and a (typically 0.25–0.5) defines the trade-off between disturbance attenuation and step response overshoot. The target loop bandwidth also needs to be selected to satisfy fundamental limitations from right-half plane poles and zeros (for more guidelines, see [21, 41]).

The above simplified selection of the target loop is well-suited to problems where only the frequency response of the plant is given. On the other hand, plants with more elaborate pole–zero structure may require a more involved selection of the target loop. A more systematic target selection can be based on the design of a full-order compensator, e.g., via linear quadratic or $\mathscr{H}_\infty$ methods, assuming that a state-space model (or transfer function) of the plant is available. In this case, L is formed by the series connection of the full-order compensator and the nominal plant.

For example, a full-order controller can be designed using LQR theory. In this manner, a desirable and feasible (with state feedback) target loop is obtained regardless of the relationship between open- and closed-loop bandwidth. Then, PID tuning becomes essentially a model order reduction of the high-order controller, weighted by the plant and the nominal loop sensitivity. More specifically, consider a state-space realization $[A, B, C, D]$ of the augmented plant model $G(s)\frac{1}{s}$. Define

$$Q = C^\top C + \rho\big(A^\top C^\top C A + \mu\|C\|^2 I\big)$$

where ρ is a parameter affecting the amount of integral action (e.g., 0.001 is often a good choice), and μ is a small number to ensure observability. Then compute the state controller gain K by solving the LQR problem $K = \mathrm{lqr}(A, B, Q, R)$ where R

is iterated such that the Sensitivity $[A - BK, B, K, -I]$ has a prescribed crossover frequency. Then, use $[A, B, K, D]$ as the target loop for the FLS tuning of the PID. This procedure has been found to be very reliable in generating reasonable target loops, given essentially one free parameter, the desired Sensitivity crossover of the closed-loop system.[2]

In cases where the plant has right-half plane zeros near the intended closed-loop bandwidth, the simplified LQR-based technique may produce state feedback controllers that cannot be approximated by dynamic output feedback controllers. In such a case, solving a full order controller design using $\mathscr{H}_\infty$ methods provides a solution in that such controllers are feasible for output feedback implementation. The only question remaining is whether they can be approximated well enough by the PID structure, and this is precisely the problem that the FLS method addresses. In particular, our experience indicates that the Glover–McFarlane loop shaping method using normalized coprime factorizations [46] is very successful in generating reasonable robust controllers without much user input, especially for difficult cases of systems with flexible modes.

Finally, note that when the minimum value of the objective (10.3) is sufficiently small, e.g., 0.2–0.3, the nominal sensitivity properties are approximately preserved for the actual closed-loop system. Large values of the objective (e.g., 0.5 or higher) may allow for unacceptable performance deterioration and can be used as an indication that the target loop bandwidth may need to be reduced. In the same vein, the value of the objective together with the frequency where the maximum occurs, can serve as indicators of confidence in the tuning process. Notice that, due to the conservatism of (10.2), a PID tuned with objective value greater than one is not necessarily a destabilizing one. However, if this maximum occurs near the crossover, the sensitivity properties and overshoot of the closed-loop will be unacceptable.

10.3.3 FLS Application to a Flexible Inverted Pendulum Model

The following example is rather extreme, but an impressive one, showing how a PID augmented with lowpass filters can be used for control of a difficult case. Needless to say, the quick PID tuning methods fail to produce a stabilizing controller. This problem was motivated by a common inverted pendulum experiment that we have in operation in our lab. The pendulum is $\frac{1}{2}$ m length with a 300 g weight at the end, and driven by a servomotor [47, 48]. One of the early experiments was the design of parameter adaptation to stabilize the PID without using any knowledge of the physical parameters. The system model was identified from data and as the classical second order rigid-body transfer function, for which the desired PID gains had been tabulated in an off-line PID tuning process for all parameters on a grid. This process proved very successful and reliable to stabilize the pendulum when the

[2] The Sensitivity crossover is related to the closed-loop crossover (Complementary Sensitivity) but it is better behaved in cases of right half-plane zeros near the desired bandwidth.

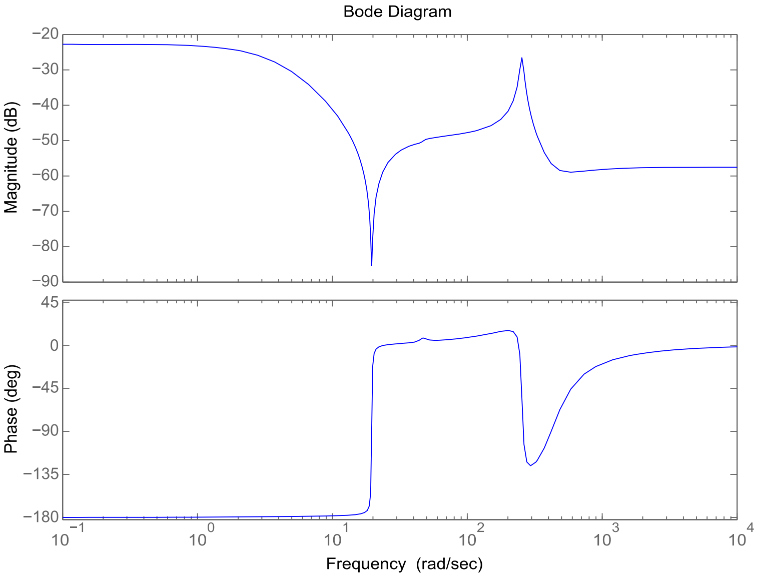


Fig. 10.2 Bode plots of the inverted pendulum model with flexible shaft

sensor data was obtained through a belt-and-gear system to increase resolution [48]. However, it failed to stabilize the pendulum when the sensor was replaced with a new, more accurate one, that eliminated all friction and natural low-pass filtering of the data. A careful system identification from high sampling rate data showed that the system model, shown below, now exhibited flexible modes that were excited by the high-power servomotor (Fig. 10.2)

$$P(s) = \frac{0.00066s^2 + 0.0029s + 1.478}{s^2 + 0.0635s - 19.54} + \frac{0.000332s^2 + 0.3785s + 177.5}{s^2 + 15.52s + 64750}.$$

The key difficulty in designing a PID controller for this system is that it requires significant phase lead and that causes the high frequency gain of the controller to increase, which in turn excites and destabilizes the flexible mode. Omitting most of the details, a successful general procedure that produced the desired PID controller with a lowpass filter was to perform a full order design and then use FLS tools to perform a weighted model order reduction of the full order controller to a PID. For the high order controller, we used the Glover–McFarlane loopshaping method [46] because it has excellent characteristics when controlling flexible modes. The initial design suggested that the controller structure could be approximated by a PID with a lowpass filter. It turns out that a good choice of a filter order is three with bandwidth just outside the system bandwidth but well-below the flexible modes to provide adequate attenuation. This controller compares well with the full order Glover–McFarlane in

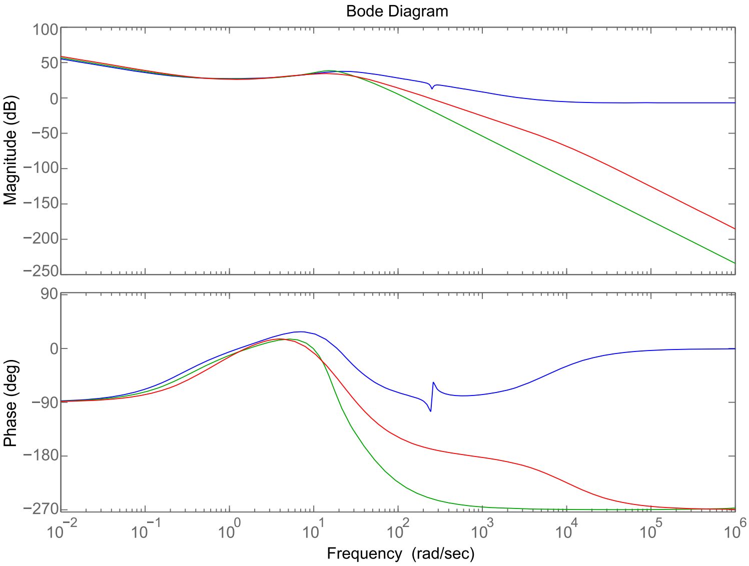


Fig. 10.3 Bode plots of the controllers for the inverted pendulum model. *Blue trace* (zero high-frequency roll-off): Glover–McFarlane optimal controller with integral action for the original pendulum. *Green trace* (lowest gain at high frequencies): Glover–McFarlane controller for the filtered pendulum model. *Red trace*: PID-plus-filter controller. After introducing the lowpass filter, the PID controller can match the robust controller response reasonably well

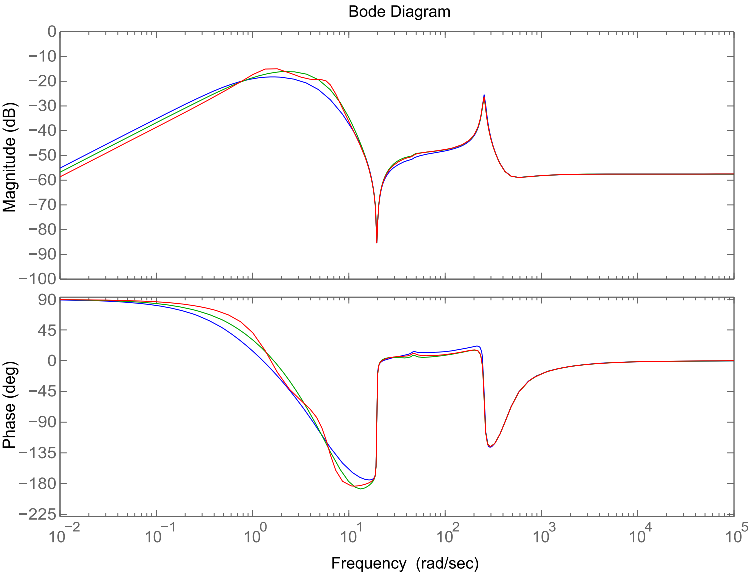


Fig. 10.4 Bode plots of the closed loop sensitivities to input disturbances for the inverted pendulum model. *Blue trace*: Glover–McFarlane optimal controller with integral action for the original pendulum. *Green trace:* Glover–McFarlane controller for the filtered pendulum model. *Red trace*: PID-plus-filter controller. Verification of the disturbance rejection properties of the PID controller

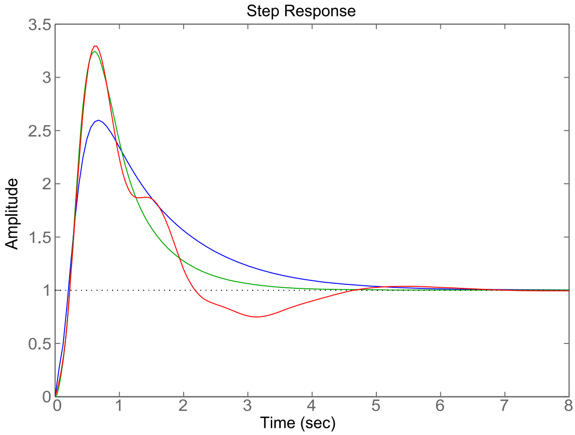


Fig. 10.5 Step responses to set-point changes for the closed loop inverted pendulum model. *Blue trace* (lower peak): Glover–McFarlane optimal controller with integral action for the original pendulum. *Green trace* (higher peak, smooth): Glover–McFarlane controller for the filtered pendulum model. *Red trace*: PID-plus-filter controller. Verification of the step response approximation of the full order controller with a PID. The optimal controller for the original model cannot be matched by a PID but the filtered one can

both disturbance attenuation and set-point tracking (Figs. 10.3, 10.4, and 10.5). The final PID-plus-filter controller computed here is

$$C_{\mathrm{PID}}(s) = \left(6.659 + \frac{20.66}{s} + \frac{8.59}{0.0001s+1}\right)\frac{1}{(0.05s+1)^3}.$$

For fairness, it should be mentioned that once a suitable controller structure is known, the phase margin method can be successfully applied as well.

10.4 Integration of System Identification with Frequency Loop-Shaping PID Tuning

One of the main advantages of using general controller design tools is that the designer has the ability to introduce frequency dependent constraints in the closed loop sensitivities. This is the case when system identification is used to provide a nominal model and an uncertainty estimate, based on the fitting residuals. We outline this method below, using as a motivating application the development of an integrated and systematic tool for system identification and controller design, for the specific problem of temperature control of diffusion furnaces. In this problem, arising in semiconductor manufacturing, it is desirable to obtain satisfactory control performance quickly and without relying on extensive trial-and-error experimentation. The control objective is to maintain a uniform temperature distribution at all times inside the furnace (zone matching) according to a set-point command that is

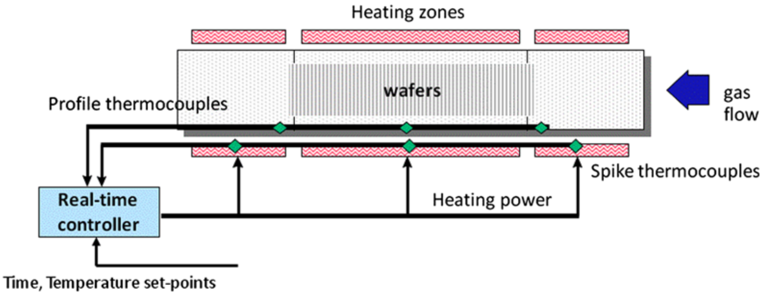


Fig. 10.6 Schematic of the temperature control loop in a three zone, horizontal diffusion furnace. The power is applied independently to each heating zone, while temperature is measured at three points corresponding roughly to the middle of the furnace and the two ends. The temperature measurements are collected near the element ("spike") and inside the heating tube ("profile")

specified by the processing recipe. The set-point undergoes ramp-up/down changes that must be followed with minimal overshoot. The motivation for that is that the heating of the wafers has a cumulative annealing effect on both the pattern geometry and the distribution of the various implants. Disturbances that affect the temperature across the furnace may be due to heat losses, gas flow, and loading/unloading operations.

A solution to this problem can be obtained by using a set of cascaded, decoupled PID controllers. One set controls the so-called "spike" temperatures (measured near the heating element) by manipulating the input power for each zone. The other controls the so-called "profile" temperatures (measured inside the furnace tube, near the wafers) by manipulating the spike set-points. A simplified schematic of a three-zone, horizontal diffusion furnace is shown in Fig. 10.6.

Despite its conceptual simplicity, the practical implementation of this approach requires considerable effort to tune the PID controllers by trial-and-error, translated into furnace down-time and loss of productivity. Based on practical experience, the main problems for such a design arise from the near-integrating nature of the spike subsystem and the differences in the dominant time-constants of each heating zone. While a full multivariable controller can address such problems in a unified manner [49, 50], the PID solution is still of interest as a basic, safe controller mode. Moreover, PID controllers can be effective in practice for furnaces where the temperatures exhibit a low degree of coupling between zones, providing performance comparable to the full order multivariable solution.

The FLS tuning technique was successfully applied to the temperature control problem of a diffusion furnace. In the current example, we are only concerned with the spike loop, whose tuning is the easier of the two because of the typically low coupling and simpler dynamics. As such, it offers a clean illustration of the application of the integrated identification and PID tuning approach.

The main steps are briefly discussed below, for the purpose of understanding the lessons learned during the development of a suitable identification procedure.

10.4.1 Identification

Several system identification methods are available in the literature (see, e.g., [42]). Among other issues, they cover input signal design, model structure, and representation of uncertainty. The scope of this section is to present a brief overview and practical guidelines for the methods that we have successfully applied.

Excitation Good input signal design is essential to obtain a model relevant for control tuning. Typical excitation sequences include random binary (two-level) signals and sums of sinusoids with a suitably selected phase, e.g., [52, 53]. While the latter offer advantages in terms of maximum amplitude for a given excitation power, they are not considered in this work.

Concentrating on the random binary signals, the excitation should have sufficient energy content around the desired closed-loop bandwidth. The signal level should be high enough for good signal-to-noise ratio, but low enough for the system to be approximately linear around the selected operating point. The maximum frequency that the data can span is roughly $\omega_{\max} = \frac{\pi}{\tau_s}$, where τ_s is the sampling time; $\omega_{\max}$ should be at least one decade above the desired closed-loop bandwidth. The frequency resolution and the minimum frequency in the data is $\omega_{\min} = \frac{2\pi}{N\tau_s}$, where N is the number of samples. Usually, for systems without sharp resonant peaks, it is enough to choose $\omega_{\min}$ one decade below the desired closed-loop bandwidth; this determines the duration of the experiment. The distribution of the signal energy into different frequencies depends on its switching frequency. The average switching frequency should be around two times the desired closed-loop bandwidth, while the minimum switching time may be constrained by system limitations. In addition, with Fast Fourier Transform (FFT)-based spectral estimation techniques, it is desirable to design the excitation signal to start and end at steady state in order to minimize aliasing effects. The implementation of these rules can often be constrained by practical limitations, e.g., operator adjustments to prevent excessive process drifts, that may alter the excitation itself or introduce a disturbance.

Uncertainty Description and Bound Estimation: Method 1 A fairly standard approach in uncertainty estimation is to consider an additive uncertainty structure. In this, the difference between the plant output (y) and the identified model output (Pu) is represented as the output of an uncertainty operator acting on the plant input u; that is, $\Delta_a u = y - Pu$. One approach to estimate a bound on the gain of Δ_a (at each frequency) is based on the asymptotic properties of Auto-Regressive with eXternal input (ARX) identification [42]. This yields the following expression for the uncertainty magnitude

$$|\Delta_a(j\omega)| = |P^n(j\omega) - P(j\omega)| + 3\sqrt{\frac{n}{N}\frac{\Phi_v}{\Phi_u}} \tag{10.4}$$

where P^n is a “sufficiently high-order” model, n is the number of parameters in P^n, and P denotes the identified model (all in a continuous-time interpretation). Φ_u

and Φ_v denote the power spectral density of the input and the residuals $y - P^n u$. According to the asymptotic ARX theory, for open-loop identification and for a plant corrupted by output noise, as n increases, the residuals $y - P^n u$ are zero-mean and normally distributed with variance expression given by the last term in (10.4). In this manner, the uncertainty bound contains two contributions: a "bias", which originates from neglected higher order dynamics $(P^n - P)$, and a "variance", which is related to the random noise present in the data. It should be noted that the high-order ARX model appears only in the uncertainty estimation and n should be large enough to obtain white residuals $y - P^n u$. From (10.4) it is obvious that n should be the smallest value that results in white residuals and whose variance does not improve significantly with any further increase of n. A common rule-of-thumb is $n \simeq \sqrt{N/40}$. (For further details, see also [54].)

A drawback of the method is that a large data set may be needed to satisfy the conditions for the asymptotic theory. In practice such data sets are often the exception. Not only identification experiments are costly, but real-life plants are routinely hit by large, unmeasured disturbances. As a consequence, the elegant reduction of conservatism achieved by this theory may need to be abandoned for the more crude estimate $|y(j\omega)/u(j\omega) - P(j\omega)|$, computed via standard spectral methods.

Finally, it is often more convenient to normalize the uncertainty estimate by expressing it in a multiplicative form defined by $P\Delta_m u = y - Pu$. In this case, the estimate of uncertainty bound becomes

$$|\Delta_m(j\omega)| = \frac{|\Delta_a(j\omega)|}{|P(j\omega)|}. \tag{10.5}$$

Invoking a standard small-gain argument [43], this estimate translates into a bound on the closed-loop complementary sensitivity (T) and, hence, a limit in the achievable closed-loop bandwidth. More precisely, the controller should be such that

$$|T(j\omega)||\Delta_m(j\omega)| < 1. \tag{10.6}$$

Allowing for a safety margin to accommodate robust stability/performance considerations, a rule-of-thumb is that an upper-bound on the closed-loop bandwidth ω_c should be $\{0.1\text{–}0.33\}\Omega$, where Ω is the unity crossover frequency of Δ_m $(|\Delta_m(j\Omega)| = 1)$.

In some cases, the multiplicative uncertainty estimate may be larger than unity at both ends of the frequency spectrum and small in mid-frequencies where the closed-loop bandwidth is placed. Identification of plants that contain near-integrators (like our motivating application of furnace temperature control) or unstable modes are such examples, particularly pronounced when the excitation interval is relatively short. For these cases, it is still possible to design a controller provided that there is a sufficiently large interval of mid-range frequencies (e.g., a decade) where Δ_m is small. The justification of such a design, however, requires a different description of the uncertainty and is briefly discussed next.

Uncertainty Description and Bound Estimation: Method 2 Here, the uncertainty estimation is based on coprime factorizations and, more specifically, follows the methodology presented in [50]. The model structure is in the form $P = D^{-1}N$, where D and N are coprime factors. The uncertainty is described in terms of two stable operators, Δ_N and Δ_D, such that the input/output equation $y = (D + \Delta_D)^{-1}(N + \Delta_N)u$ is satisfied. The estimation error is related to the coprime factor uncertainty as:

$$e = \Delta_D y + \Delta_N u. \tag{10.7}$$

This is the same error that is minimized by the usual equation error estimators. By the small gain theorem, the following robust stability condition can be derived for the closed-loop system

$$|N^{-1}(j\omega)T(j\omega)||\Delta_N(j\omega)| + |S(j\omega)D^{-1}(j\omega)||\Delta_D(j\omega)| < 1. \tag{10.8}$$

Since the controller has not been designed yet, the sensitivity S and complementary sensitivity T are unknown. Moreover, the dependence of y on u does not allow a derivation of individual bounds for the two uncertainty operators. Still, as a first step, we can compute their achievable bounds and set-up constraints that S and T should satisfy. For this, we adopt an "unfalsification" approach whereby we seek to determine the most favorable decomposition of the uncertainty in the sense of (10.8), subject to the constraint (10.7). The result of this approach could be interpreted as a pseudo-necessary condition for robust stability. Of course, a key difficulty is that the uncertainty estimates depend on the loop sensitivities and the latter can only be selected after the uncertainty bounds become available. Nevertheless, if we assume some bounds on S and T the "unfalsification" problem translates into the minimization of the left-hand side of (10.8) subject to (10.7).

A suboptimal but attractive solution can be computed with the additional constraint of $\Delta_D y$ being orthogonal to $\Delta_N u$ at each frequency. The latter would imply that at each frequency point the error is attributed to only one of the two uncertainty operators. (This is true in the multivariable case as well.) At high frequencies, $T \simeq 0$ and $S \simeq 1$, while $e \simeq y$ (i.e., poor model), implying that the typical behavior of the optimal solution to (10.8) with a reasonable controller will make Δ_D inactive. Thus, the high-frequency portion of the estimation error is translated into a maximum bandwidth condition or, more precisely, an upper bound on the complementary sensitivity. The reverse situation occurs at low frequencies, allowing large modeling errors there to be translated into a minimum bandwidth requirement, in the form of an upper bound for the sensitivity. This analysis indicates that the following asymptotic bounds should hold

$$|T(j\omega)| < |\frac{\mathrm{FFT}(u)}{\mathrm{FFT}(e)}||N(j\omega)| \quad \text{at high frequencies,}$$

$$|S(j\omega)| < |\frac{\mathrm{FFT}(y)}{\mathrm{FFT}(e)}||D(j\omega)| \quad \text{at low frequencies.}$$

Guided by these easily computable bounds, it is usually straightforward to determine constraints for the closed-loop bandwidth as an interval where both bounds are greater than unity. This interval should span at least a decade of frequencies to allow a smooth transition of the LTF between high and low loop gains. The lack of such an interval indicates failure of the identification for the purpose of robust controller design.

The actual computation of bounds for the coprime factor uncertainty is deferred until the target loop is selected. Nevertheless, an approximate LTF can be used at this point to solve the above optimization problem and obtain an estimate of the robust stability condition (10.8). The distance of the latter from unity is an a priori confidence indicator for the controller. This computation is not critical in the SISO case where the T and S bounds are just as conservative as the robust stability condition itself.

Parameter Estimation There is a great variety of methods for parametric system identification, and [42] is a standard reference. Our personal preference also follows similar guidelines with different emphasis in some details, such as, initial condition estimation and parameter estimate regularization. This method, described in more detail in [50, 51], uses a multiple-input, single-output (MISO) approach and relies on a least squares parameter estimation algorithm to obtain parameter estimates for a linear model that describes the process locally around an operating point. For a multiple-input, single-output system and under an observability assumption the model is written as:

$$\dot{x} = Fx + \theta_1 u + \theta_2 y, \quad y = qx + \theta_3 u$$

where F and q are selected a priori so that F is Hurwitz, (F, q) is a completely observable pair, and $\theta_1, \theta_2, \theta_3$ are adjustable parameters. The usefulness of this description is that it can be readily converted into a linear model form, which is convenient for parameter estimation, that is, $y = w^\top \Theta$. Here, Θ is a vector containing all the adjustable parameters (elements of $\theta_1, \theta_2, \theta_3$) as well as the initial conditions $x(0)$. The latter term, often ignored, has been found to have an appreciable impact, especially for short data sets that begin on a transient. The regressor vector w contains the signals $(sI - F^\top)^{-1} q^\top u$, $(sI - F^\top)^{-1} q^\top y$, u, and $(sI - F^\top)^{-1} q^\top$, where the last term corresponds to the unknown initial conditions. After generating the regressor vector, Θ can be determined in a least squares sense by minimizing the estimation error $\| y - w^\top \Theta \|_2$. The above description is repeated for each output and the resulting state-space model is concatenated to produce the overall model of the system. While this approach may result in a non-minimal model, with a proper selection of the model orders (dimensions of F), a model reduction is rarely necessary. Emphasis of the model accuracy around the crossover frequency is, of course, crucial and should be reflected by the proper selection of the excitation sequence (typically a random binary sequence) and the identification design parameters (input signal, prefilters, etc. [42]). This formulation of the identification problem is similar to the more standard ARX formulation with the addition of the filter described by (F, q). The state-space formulation offers some advantages in numerical conditioning properties, while the error expression that is minimized is also the error that enters in the uncertainty formulation. Nevertheless, depending on user's preference, other identification algorithms can be used to perform this step just as well, as long as the estimated model is accompanied with the corresponding description of uncertainty.

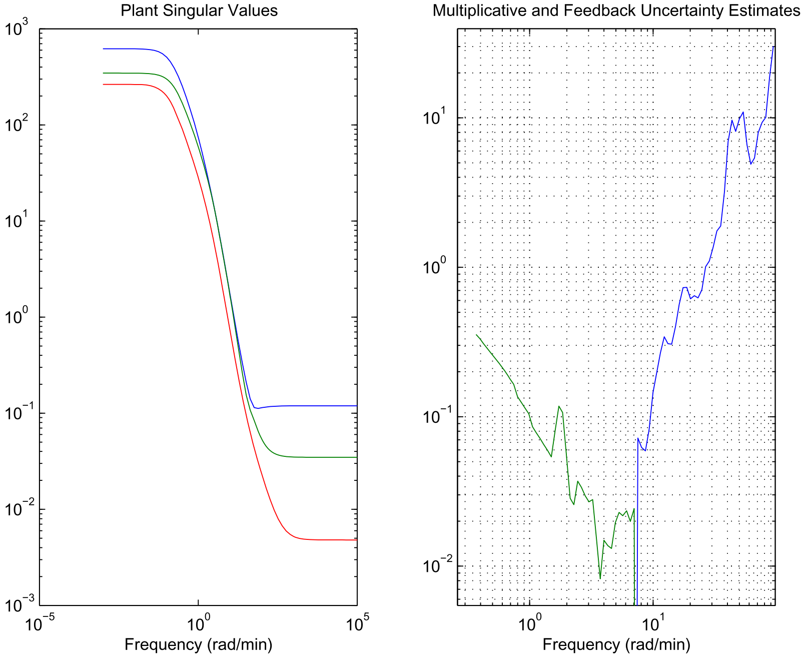


Fig. 10.7 Singular value plots of the spike temperature model (*left*). Uncertainty estimates expressed as inverse bounds for the Sensitivity and Complementary Sensitivity (*right*)

10.4.2 Integrated Identification and PID Tuning for the Furnace Temperature Control Problem

The procedure was used to tune the inner (spike) PID controllers. The results reported here (courtesy of SEMY Eng. Inc.) are from a three-zone, 5 in, quartz, Thermco furnace. The models of the power-to-spike temperature were obtained from an identification experiment, performed around the 900 degree steady-state, using a third degree transfer function for each zone. The singular value plots of the resulting ninth order model and the corresponding uncertainty estimates obtained for the furnace temperature control problem are shown in Fig. 10.7.

For this particular furnace, the models show a relatively low coupling between zones (about 10%), assuring the viability of the decoupled PID approach. The spike PIDs were tuned based on the diagonal entries of the power-to-spike model, augmented by a first-order noise filter with bandwidth 10 rad/min. Also, in all cases the derivative term was implemented with a low-pass filter with the same bandwidth ($\tau = 0.1$ min). The tuning was performed with the second-order target LTF, for a desired closed-loop bandwidth of 3 rad/min. This choice of the bandwidth takes into account the uncertainty estimate associated with the model, the mismatch due to the diagonalization of the plant, as well as the ability of the PID to achieve a low approximation error (around 0.3). Right-half plane zeros did not impose any considerable limitations in this example.

To assess the limitations imposed by the decoupling, the Perron eigenvalues of the off-diagonal entries of the plant are used, after normalization with the minimum singular value of the diagonal plant. This computation produces an effective multiplicative uncertainty that translates into a loop-bandwidth constraint, a technique introduced in [44] to reduce the conservatism of the small gain theorem with diagonal uncertainty.

Nevertheless, for this particular example the main design constraint comes from the ability of the PID to match the target loop without excessive sensitivity peaking. The three PID controllers reported below (time scale is in min) give an indication of the differences among the three zone controllers that are required to match the response of all zones:

$$
\begin{aligned}
C_{\mathrm{PID},1}(s) &= 0.035 + \frac{0.025}{s} + \frac{0.017}{0.1s+1},\\
C_{\mathrm{PID},2}(s) &= 0.082 + \frac{0.069}{s} + \frac{0.039}{0.1s+1},\\
C_{\mathrm{PID},3}(s) &= 0.038 + \frac{0.038}{s} + \frac{0.017}{0.1s+1}.
\end{aligned}
$$

The zone matching is illustrated in Fig. 10.8 for the simulated plant model. Also notice, from the input disturbance sensitivity singular values, that the controller possesses significant integral action, something that is required to achieve adequate disturbance attenuation. This integral action introduces a necessary trade-off of an overshooting response under spike control.

The simulation results showed excellent agreement with the actual furnace response. For this, the PID controllers were augmented with an anti-windup mechanism and were implemented on a 486-based embedded controller board. The overall performance of this controller, including control input activity, was at least as good as the existing PID for this furnace whose tuning was the result of considerable trial-and-error experimentation. Using the FLS approach, the tuning of the PID parameters was essentially a “one-pass” procedure. Furthermore, its performance was comparable to a multivariable $\mathscr{H}_\infty$-based controller [50], in terms of disturbance attenuation and only somewhat worse in terms of zone matching and end-of-ramp overshoot. It should be mentioned, however, that the successful application of the decoupled PID structure relies on the low coupling between the heating zones. In general, industrial furnaces may—and often do—exhibit considerably higher zone interactions, especially in the profile temperature subsystem. In such a case, the PID controller will not provide a satisfactory closed-loop performance. Nevertheless, the PID tuning is still of interest in this application, especially for spike temperature control. In this, it provides a reference point and a basic control mode that is compatible with previous practice.

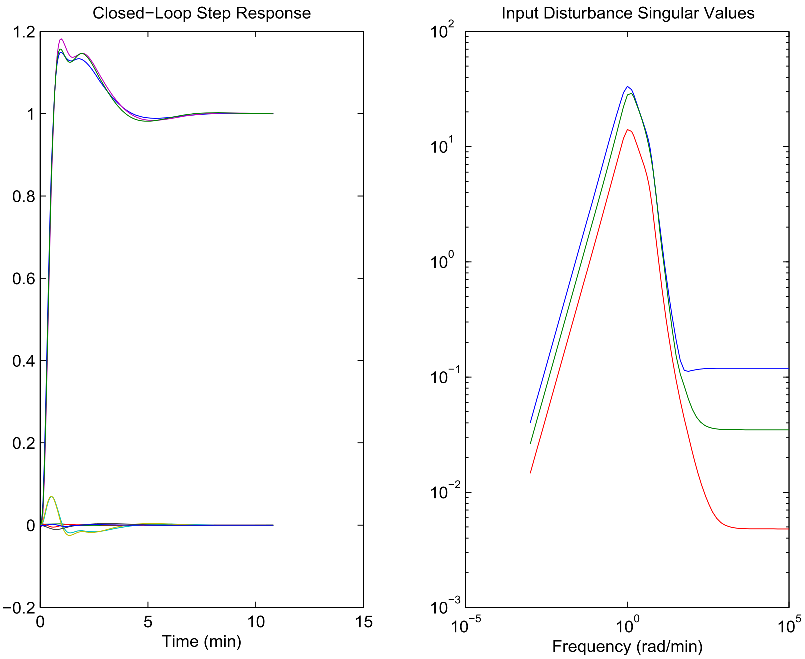


Fig. 10.8 Simulated step response of the furnace temperature closed loop system, showing the decoupling and excellent matching of all channels (*left*). In this figure, the responses to a single step in each channel are overlayed to illustrate the similarity of the responses of the stepped channels in terms of settling time and overshoot and the decoupling of the rest of the non-stepped channels that remain close to zero. Singular values of the input disturbance sensitivity illustrate the integral action (*right*)

10.4.3 Other Applications: Tuning for Multiple Models

Multiple models can arise when disjoint data sets are available for a single plant. These data sets may be the result of identification experiments performed at different operating points, in different time periods, or under different disturbance conditions. In this case, each of the sets could be processed following one of the methods in Sect. 10.4.1 to produce a different model. Sometimes, in the industry practice, only the models of the plant are recorded and data and uncertainty information are discarded. In other cases, the fitting error for each set is very small and most of the uncertainty can be attributed to changes in the operating conditions.

Under these circumstances, the methods described in Sect. 10.4.1 for the estimation of multiplicative uncertainty are not directly applicable and it is necessary to find other means of reflecting plant variations. A viable solution is to consider each one of the available models as nominal, and compute the uncertainty based on the remaining models as:

$$|\Delta_{mj}(j\omega)| = \max_{P_i} |\frac{P_j(j\omega) - P_i(j\omega)}{P_j(j\omega)}|. \tag{10.9}$$

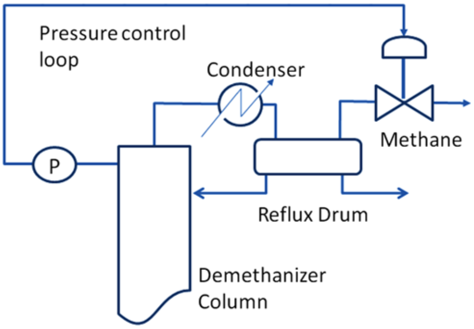


Fig. 10.9 Pressure control loop abstraction in a demethanization column. The feed stream undergoes significant variations in temperature and composition that affect the column pressure. The pressure is controlled by adjusting the methane flow drown from the top of the column. (For proprietary reasons, only an abstraction of the operations is shown)

Then, the smaller of these bounds is selected as an estimate of the multiplicative uncertainty, indicating that the corresponding nominal model is a better representation of the set. This uncertainty description is somewhat arbitrary. Alternatively, one could select as a center model the one derived from the most likely operating condition, or some sort of mean (geometric or arithmetic) of all the models available to describe the system.

A single controller that takes into account all of the existing models can be tuned with the FLS method outlined, with the objective (10.3) substituted by its multiple-model counterpart:

$$\min_K \left\| \begin{pmatrix} S_1(P_1 C_K - L_1) \\ \vdots \\ S_m(P_m C_K - L_m) \end{pmatrix} \right\|_\infty . \tag{10.10}$$

Here, L_i and S_i denote the target loop shape and the resulting target sensitivity, respectively, for each of the plant models P_i. The selection of different L_i may be necessary, especially in the case where the models differ in slow dominant dynamics. Since the controller C is common to all models, it is clear that $L_i = P_i C$ should be different for different P_i's. Of course, if the differences among the multiple models are significant, the optimal solution to (10.10) may not reach a satisfactory value, and other strategies should be considered, e.g., controller scheduling. Finally, in the multiple model case, the minimum value achieved in (10.10) can also be regarded a measure of the closed-loop robustness since the plant uncertainty or variability is already embedded in that expression.

Here, as an application we consider the pressure control loop in a "demethanization" unit whose goal is to control the inlet header pressure of the unit by adjusting the downstream compressor settings. A simplified schematic of the process is shown in Fig. 10.9. The header is often hit by significant disturbances in supply gas stream pressure and composition. The header pressure also has a safety high limit, which corresponds to the relief valve settings for the plant. From a unit optimization view-

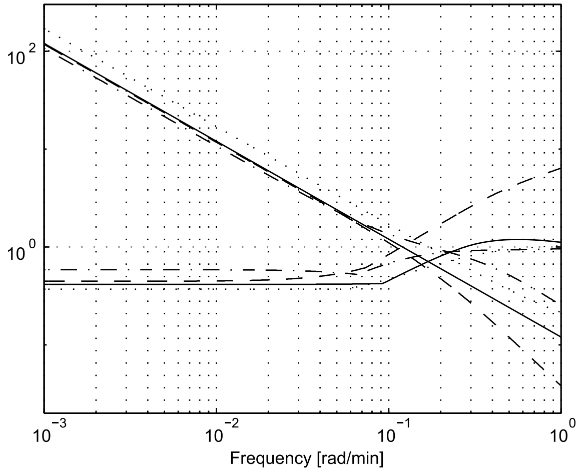


Fig. 10.10 Pressure control loop plant frequency response and multiplicative uncertainty obtained by considering each model in Case 2 as nominal

point, it is advantageous to operate the header pressure as close to the safety limit as possible. This requires good pressure regulation. The plant personnel had spent considerable time in tuning this loop. A specific tuning would work for a certain period, and then it would fail to work. The operators had to retune the loop quite often. Based on their operating experience, it was suspected that the plant model changes often and unpredictably. Consequently, four different identification experiments were conducted at different time periods. They produced the following four process models (time in min):

$$P_1(s) = -\frac{0.12}{s},$$

$$P_2(s) = -\frac{-0.0072245(s-7.543)(s+0.8763)}{s(s+1.563)(s+0.2611)},$$

$$P_3(s) = -\frac{0.23834(s+0.6486)(s+0.1196)}{s(s^2+0.7634s+0.1727)},$$

$$P_4(s) = -\frac{11.9643(s+1.269)}{s(s+10)(s+8.929)}.$$

Figure 10.10 shows a plot for the multiplicative uncertainty computed by (10.9). This indicates that G_4 (solid line) is a better choice as nominal model since it produces the more relaxed bound. The corresponding unity cross-over frequency is $\Omega = 0.22$ rad/min.

All the above models contain an integrator, and therefore, the second order target LTF was selected as $L(s) = \omega_c(s+0.444\omega_c)/s^2$, with the zero corresponding to roughly 20% overshoot. Since the objective of the design was to maximize the closed-loop system performance, ω_c was selected as the maximum recommended limit, i.e., 0.07 rad/min. The PI controller gains were computed as $K_p = -0.5$, $K_i = -0.0167$, tuned according to (10.10) and corresponding to a fitting error of 0.3. In the attempt to improve regulatory performance, PI tunings with bandwidth of 0.1 and 0.2 rad/min were also tested. The response became lightly damped

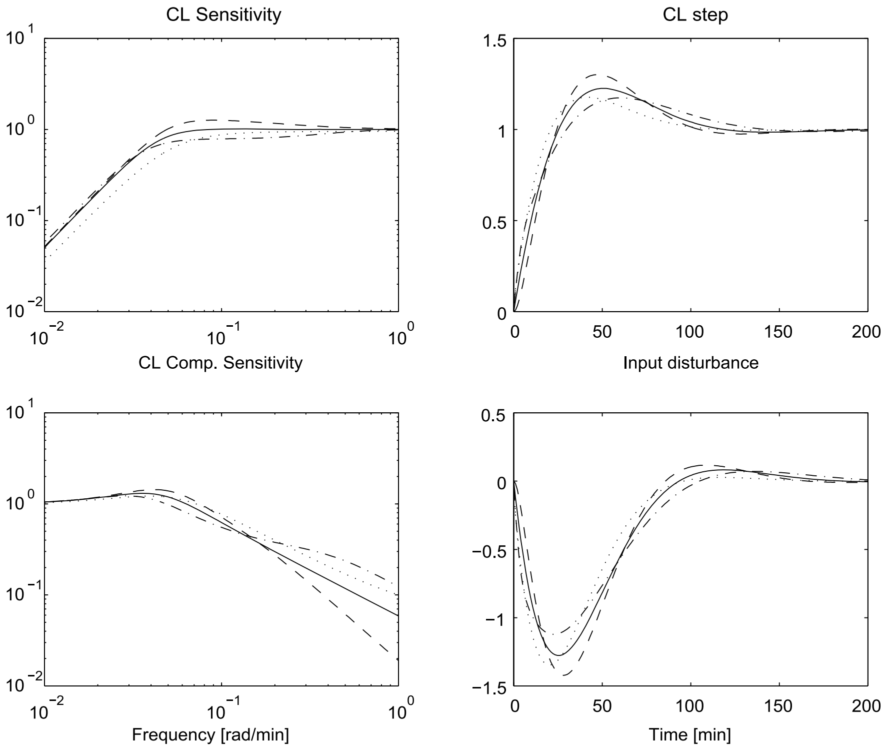


Fig. 10.11 Simulated closed-loop performance for the different models in the pressure control loop application

for $\omega_c = 0.1$ rad/min, and sustained oscillations were present for $\omega_c = 0.2$ rad/min. This observation is in accordance with the general recommendation that the loop bandwidth should not exceed the crossover frequency of the uncertainty estimate.

The simulated behavior, illustrated in Fig. 10.11 for all the four models, is consistent with the observed closed-loop performance.

10.5 Direct Identification of the PID Parameters

With the recent advancements (and acceptance) of system identification techniques, many of the off-line PID tuning strategies can be readily converted to on-line tuners following the indirect adaptive control paradigm [25, 55]. Of course, such an approach may necessitate the execution of the parameter adjustment at a slower rate than the sampling rate of the closed-loop. This is not a serious drawback since, typically, the PID tuning occurs under monitored conditions and with the injection of carefully designed excitation, while the tuned parameters are not expected to exhibit fast variations. On the other hand, direct adaptation/tuning of the PID parameters is an interesting alternative, with potential benefits in speed of execution as well as performance optimization. That is, direct adaptation has the potential of optimizing the controller parameters to the specific properties of modeling mismatch, in relation to the tuning objective, instead of relying on plant identification and the subsequent conservative controller tuning.

In the following, we present results on the problem formulation and application of an approach for the direct tuning and adaptation of PID parameters with a loop-shaping objective. With the filter-bank framework of [56], we employ an update law that approximates the constrained minimization of the operator norm of the error system rather than the energy of the error itself. Under persistent excitation, the tuning with this adaptive scheme is comparable (and ideally the same) with the off-line design. One advantage of this approach is that the values of the minimization objective can be interpreted directly as a bound on the modeling mismatch or uncertainty. As such, it provides a simple indicator to assess tuning confidence. Practically, this offers a monitoring signal to determine on-line the state when the injected excitation has provided sufficient information to tune the controller and whether the target loop selection leads to a stabilizing controller. Coupled with the ability to perform this monitoring and tuning without the controller actually being in the loop, this approach is very attractive for industrial applications.

Another advantage of this approach is the observation that the controller tuning is relatively insensitive to the detailed properties of the injected excitation, especially when the mismatch between target loop and feasible loops is large. This is the case when the plant contains significant contributions of high order dynamics, and/or simple target loops are used with delays and right half-plane zeros near the desired bandwidth. This robustness property to excitation is a by-product of the normalization of the error signal, effectively resulting in the approximate optimization of the $\mathscr{H}_\infty$ norm of the error operator. In contrast, a typical least-squares minimization of the error signal results in a different trade-off depending on the excitation and the disturbances affecting the system at that particular time, which may correspond to a poor controller.

10.5.1 Filter-Bank Implementation of FLS

The development of an on-line direct adaptive algorithm that approximates the off-line FLS solution begins with the transformation of the error operator minimization (10.3) into an error signal minimization. Such a transformation essentially aims to utilize the plant input output data instead of its transfer function (or other model) in order to solve the optimization problem (10.3). More specifically, with (u, y) denoting the plant input–output pair, we consider the estimation error signal $e_e = S(CP - L)u$. Since $y = Pu$, $e_e = SCy - Tu$ which is a signal that can be constructed from input/output data, processed by stable filters. Then the minimization objective in (10.3) can be expressed as

$$J = \|S(CP - L)\|_\infty = \sup_{\|u\|\neq 0} \frac{\|e_e\|_2}{\|u\|_2}.$$

Notice that, due to the special PID structure, the estimation error has the familiar linear-in-the-parameters form

$$\begin{aligned} e_e &= w^\top \theta - z, \\ w^\top \theta &= SC(\theta)y, \;\; z = Tu. \end{aligned} \tag{10.11}$$

In general, this equation should contain a swapping term for time-varying θ, and a term describing the decaying effect of initial conditions. A recursive algorithm that minimizes the energy of such an error signal can be found in [55]. It was also used in [57] in combination with a suitable dead-zone to ensure that the contribution of the error system would eventually be bounded by the dead-zone, in an operator gain sense.

Here, however, we seek to minimize the error system gain, at least approximately. To achieve this, we employ a filter-bank of bandpass filters $\{F_i\}$ to decompose the input and output into several frequency bands and then minimize the maximum cost functional. Although this is only an approximation, it has been found to produce reasonably accurate results for the plants of interest with a relatively small number of filters. Of course, its accuracy improves as the number of filters increases and their transitions become sharper. In any case, this decomposition makes sense independent of the frequency domain interpretation, since the $\mathscr{H}_\infty$ norm is the worst case ratio of output norm to input norm for all possible inputs. In our case, we only minimize the maximum over a finite set of inputs but the required operations do not suffer the usual time-to-frequency domain conversion problems. The only assumption is that the plant is approximately linear in the operating region so that the filters and the plant operator commute.

Thus, the optimization objective becomes approximately

$$\max_i \frac{\|S(CP - L)F_i u\|_2}{\|F_i u\|_2} \leq \max_i \frac{\|SCF_i y - TF_i u\|_{2,\delta}}{\|F_i u\|_{2,\delta}} \tag{10.12}$$

where $\|\cdot\|_{2,\delta}$, $\delta > 0$ denotes the exponentially weighted 2-norm, used here to allow for the recursive implementation of the resulting estimator (bounded cost functional for bounded signals).

The adaptation algorithm is based on the discrete-time implementation of a recursive least squares algorithm [55]. The optimization problem to be solved is a discrete-time approximation of (10.12), subject to parameter constraints:

$$\begin{aligned} &\min_{\theta \in \mathscr{M}} \max_i \frac{J_{i,k}}{m_{i,k}}, \\ &J_{i,k}(\theta_k) = \sum_{n=0}^{k} \lambda^{k-n} |z_{i,n} - w_{i,n}^{\mathrm{T}} \theta_k|^2, \\ &m_{i,k+1} = \lambda m_k + |[F_i u]_{k+1}|^2 \end{aligned} \tag{10.13}$$

where $J_{i,k}$ are the partial costs, with signals z and w generated with the filtered (by F_i) input–output pair; $\lambda = e^{-2\delta} \in (0, 1]$ is a constant forgetting factor; $m_{i,k+1}$ is the $\{2, \delta\}$-norm of the corresponding filtered version of u. Values of λ near 1 (small δ) produce approximately the same result as the off-line tuning. Smaller values of λ speed up the convergence of the cost functionals to a steady-state but they overestimate the uncertainty bound and minimize a shifted $\mathscr{H}_\infty$ norm. The convex optimization problem (10.13) can be solved using standard algorithms. In the interest of keeping the complexity of the on-line computations and code low, we note that a simple suboptimal computation of the updates can also be used [58].

We conclude this section with some general remarks regarding the application of this algorithm. In its present form, the algorithm requires u to have energy at all the frequency points to avoid an ill-posed objective. Division by zero can be prevented either by a simple numerical fix, or by combining some of the outputs of the filter-bank. As an adaptive algorithm, it also has a stationary point at $\theta = 0$, where $u = 0$. However, these cases do not represent the intended use of this adaptive algorithm. Adequate input excitation, e.g., by injecting external excitation at the reference or at plant input, is essential in order to ensure reliable tuning. Although the normalization of the $\mathscr{H}_\infty$/FB estimator signals removes some of the sensitivity of the estimates to the spectral properties of the specific input, an appropriate excitation signal design is essential to obtain a model relevant for controller tuning.

In most cases, a random binary (two-level) signal that has sufficient energy content around the desired closed-loop bandwidth is appropriate. The signal level should be high enough for good signal-to-noise ratio, but low enough for the system to be approximately linear around the selected operating point. On the other hand, this signal normalization can also introduce significant bias in the case of disturbances that have energy in a few frequencies where they mask the useful signals. For this reason, it is beneficial to augment the algorithm with suitable bandpass filters that can improve the signal-to-noise ratio. Their drawback is that they alter somewhat the optimization objective, but this is usually minor. In counteracting the disturbance effects, an appropriate selection of the filter-bank may be helpful, and, of course, limiting the adaptation to intervals where the effects of external bounded disturbances are low is a commonly followed sensible practice.

For practical implementation, the possibly non-zero offsets for the input and output (e.g., linearization steady-state) should be taken into account. This can be achieved by introducing an additional constant term in the linear model or, in a quick fix, filtering the input–output pairs by a high pass filter; for the latter, its time constant should be slow enough so that it does not affect the optimization result.

Last, but not least, this algorithm enjoys the highly desirable property that it is unaffected by input saturations, as long as their level is known. Of course, in this case, the implemented PID should contain an integrator anti-windup mechanism, e.g., as in [59, 60].

10.5.2 $\mathscr{H}_\infty$/FB PID Tuning Examples

The $\mathscr{H}_\infty$/**FB** adaptive algorithm was successfully simulated in a variety of problems. The examples presented here represent some of the difficult cases found in applications, including plant integrators and right half-plane zeros near the tuning bandwidth. The former may often result in the adaptation shutting down the integral action. The latter contains a non-invertible element making the target loop harder to approximate.

We consider different transfer functions to demonstrate various aspects of the tuning process and emphasize the differences between the $\mathscr{H}_\infty$/**FB** and standard least-squares estimation

$$P_1(s) = \frac{-0.5s+1}{s(s+1)}, \qquad P_2(s) = \frac{e^{-s}}{s+1}.$$

Our objective is to tune a PID to a crossover frequency around $\omega_c = 1$ rad/s. We select target loops as

$$L_1(s) = \frac{s+0.5}{s^2}, \qquad L_2(s) = \frac{1}{s}$$

for each of the plants. This choice represents a relatively high bandwidth loop for the plant right half-plane zero. Using the FLS technique and a sampling time of $\tau = 0.05$ s, the off-line tunings for each of these cases are found to be

$$[K_p, K_i, K_d]_1 = [1.02, 0.29, 0.70]; \quad \text{approx. error is } 0.42;$$
$$[K_p, K_i, K_d]_2 = [0.67, 0.53, 0.20]; \quad \text{approx. error is } 0.48.$$

We construct a tuning scenario where a square wave excitation of frequency 0.1 rad/s and/or a random excitation of amplitude 5 are injected at the reference input for 200 s. The same square wave continues as a reference signal for the period 200–500 s. After time 300 s, a square wave disturbance of amplitude 1 and frequency 0.1 rad/s is injected at the plant input to verify the disturbance rejection properties. In a half of the cases, we introduce a slow sinusoidal disturbance at the plant output, with amplitude 2 and frequency 0.015 rad/s. This disturbance is active during identification and serves to expose limitations of the identification caused by drifting disturbances. Also, we use an optional bandpass filter to emphasize the input/output content around the crossover. This filter is composed of two second order-order filters, one low-pass and one high-pass, with cutoff frequencies one decade above and one below the desired crossover frequency.

We use this scenario to evaluate the performance of the $\mathscr{H}_\infty$/FB PID tuner that utilizes a bank of 20 band-pass filters of the form:

$$F_1 = \frac{\omega_1}{s+\omega_1}; \qquad F_{20} = \frac{s}{s+\omega_{19}},$$
$$F_i = \frac{(\omega_{i+1}-\omega_i)s}{(s+\omega_i)(s+\omega_{i+1})}, \quad i = 2, \dots, 19$$

(their corner frequencies are logarithmically spaced between 0.1–10). For comparison, we also construct an estimator using standard fading memory Least Squares (LS), with a small forgetting factor corresponding to 1000-sample time-constant. The simulations with the LS estimator verify that the increased complexity of the $\mathscr{H}_\infty$/FB is justified in the cases of large approximation errors due to plant-target mismatch.

Our general observation is that both estimators are successful when the target loop is near the feasible set (distance measured in terms of the loop sensitivities), and there are no external disturbances. With large disturbances, successful tuning requires that the disturbance has low energy relative to the signals used for parameter estimation. This can be enhanced if there is a separation between disturbance frequency content and the loop crossover frequencies. Typical examples include the natural frequency weighting introduced by a robust stability optimization objective, the use of bandpass filters, and the careful selection of the injected excitation. Apart from these ordinary signal-to-noise related issues, there is an important difference between estimation algorithms in their behavior with infeasible targets and in the efficient use of the available excitation. The $\mathscr{H}_\infty$/FB estimation produces consistent near-optimal tunings with adequate excitation and relatively small external disturbances. The observed failures of the $\mathscr{H}_\infty$/FB estimation were in cases where a large drift disturbance was present and a bandpass filter was not used. On the other hand, the Least Squares (LS) estimation produces results that are more sensitive to the excitation, especially when the target loop is far from the feasible set. Failures with this estimator were observed even in the absence of disturbances, with or without bandpass filters.

Figures 10.12 and 10.13 show the responses for the plants P_1 and P_2 with high excitation and bandpass filtering and a slow sinusoidal disturbance. This is the baseline case where both estimators yield successful tunings. Notice how the partial costs (10.13) of $\mathscr{H}_\infty$/FB approximate the off-line maximum approximation error, providing an on-line measure of the confidence in the tuning.

Figures 10.14 and 10.15 shows that, under the same conditions as before, a change in the excitation can cause the LS estimation to fail while the $\mathscr{H}_\infty$/FB estimation is still successful. Here, the excitation is a random input that is masked by the large mismatch between the feasible loops and the target loop. It should be mentioned that in this case it is not the output disturbance causing the problem since the results remain largely unchanged if the disturbance is removed. The $\mathscr{H}_\infty$/FB overcomes this problem by separating the spectral components and applying an optimization that approximates the off-line one.

10.6 Conclusions and Future Perspectives

In this study, we discussed several aspects of PID tuning from the perspective of system identification and its dual counterpart controller order reduction. Simple PID tuning procedures like Ziegler–Nichols utilize only minimal information (one frequency point) about the plant and have been very successful by restricting the class

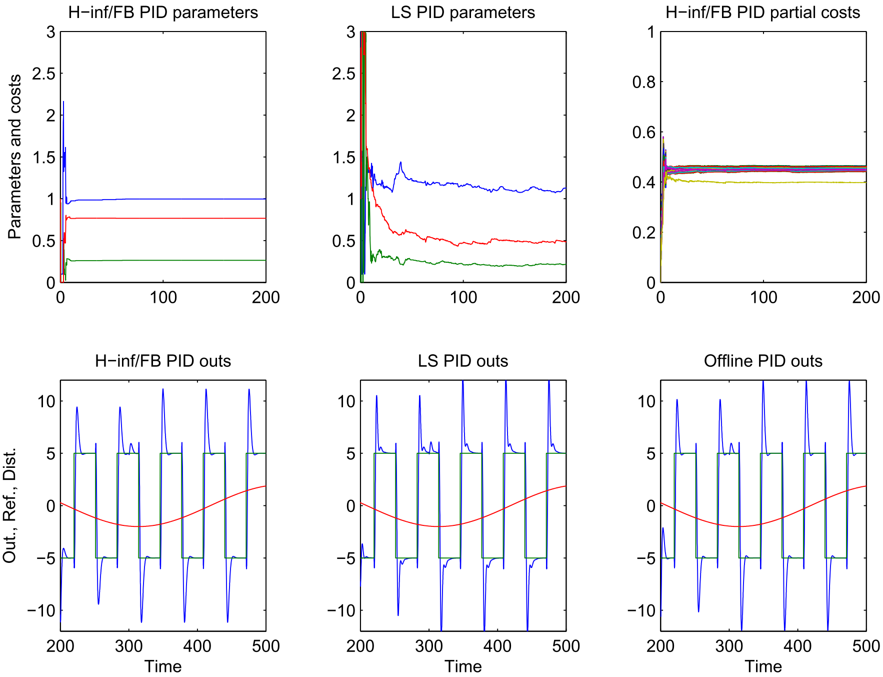


Fig. 10.12 Successful PID tuning for the plant P_1. The excitation is random-plus-square-wave at the reference input, a drift disturbance is present and bandpass filtering is active. Both estimators result in a successful tuning, with parameters close to the off-line estimates. *Legend*: Plots clockwise from top-left: Parameter estimates with the $\mathscr{H}_\infty$/FB estimator. Parameter estimates with the LS estimator. Partial costs of optimization objective of the $\mathscr{H}_\infty$/FB estimator. Output, reference, and disturbance with the $\mathscr{H}_\infty$/FB estimator. Output, reference, and disturbance with the LS estimator. Output, reference, and disturbance with the off-line $\mathscr{H}_\infty$ estimator

of applications to systems where this information is sufficient for controller design. Seeking to expand the applicability and reliability of tuning procedures for PID controllers, we employ general system identification methods to obtain a general plant model together with an associated description of uncertainty. This translates the PID tuning problem as the reduction of a general full-order controller to the reduced degree-of-freedom PID. The use of such a heavy machinery is an overhead to be paid in order to improve the reliability of the tuning and avoid costly trial-and-error iterations. It offers the ability to include, not only uncertainty estimates in the PID tuning, but also a multiple model description of the plant that is to be controlled by a single PID. While it is certainly not a “back-of-the-envelope” computation, the full order controller design is relatively straightforward and well established so that it can be automated and supported by all modern computer platforms, tablets, etc.

The special linear-in-the-parameters form of the PID also allows for direct identification of its parameters and on-line tuning. The algorithm employs a filter-bank and a convex optimization procedure to emulate the off-line tuning by FLS,

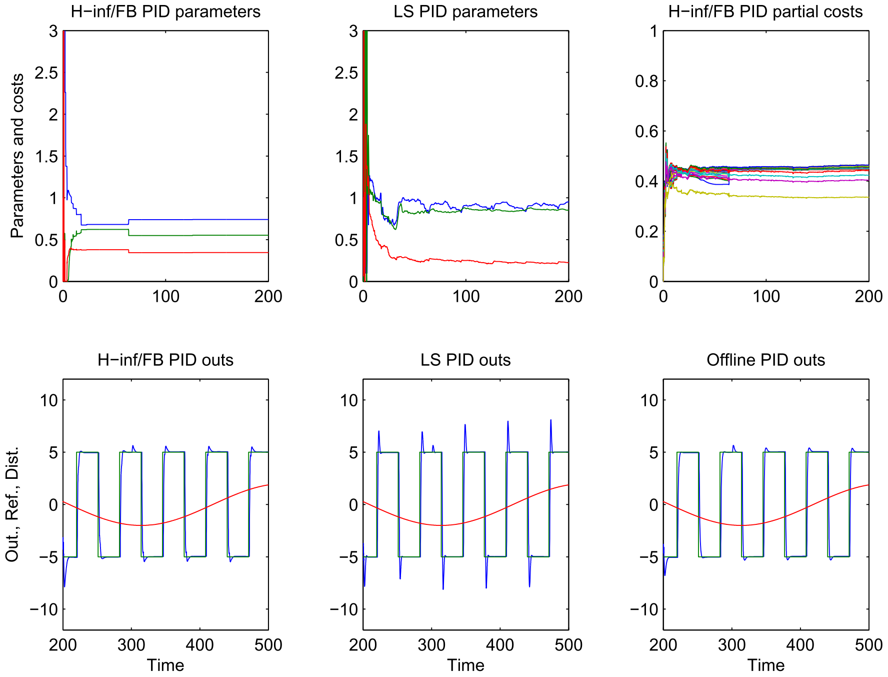


Fig. 10.13 Successful PID tuning for the plant P_2. The excitation is random-plus-square-wave at the reference input, a drift disturbance is present and bandpass filtering is active. Even though the LS estimates result in a more underdamped loop, both estimators produce reasonable results. (Legend as in Fig. 10.12)

which is an $\mathscr{H}_\infty$ optimization. The complexity of the $\mathscr{H}_\infty$/FB estimator is justified by the robustness of the tunings with respect to the spectrum of the excitation, even for large mismatch errors between the set of feasible loops and the target loop. This is not the case for the much simpler LS estimators that, when the mismatch error is large, tend to be sensitive to the spectral properties of the excitation and can produce unacceptable tuning results, even in the absence of disturbances.

This adaptive controller is suitable for use in loops where excitation is present or can be injected in order to perform the controller tuning, and its magnitude is sufficiently large to yield a good signal-to-noise ratio. It is susceptible to narrowband disturbances, but their effect can be mitigated with bandpass filters, as long as they do not appear around the target closed-loop crossover. An important by-product of our analysis is that the objective function of the $\mathscr{H}_\infty$/FB optimization can be used for assessment of the tuning performance, as well as continuous monitoring of the controller. This can have interesting practical implications even with other types of controllers.

Finally, an ever-important consideration, common to all tuners, is that an update of the target loop (control objective) may be required if the plant is considerably different from its nominal value. This problem is beyond the scope of the present

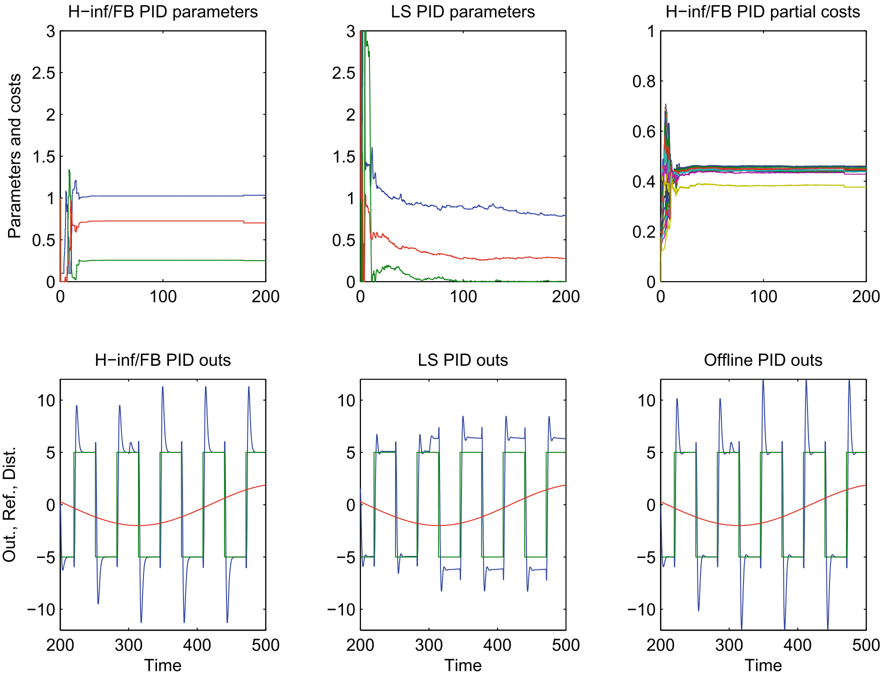


Fig. 10.14 For the plant P_1, random-only excitation at the reference results in failure for the LS PID tuning, driving the integral gain to zero. The $\mathscr{H}_\infty$/FB estimation still converges approximately to the correct (off-line) result. A drift disturbance is present and bandpass filtering is used. A similar result is obtained without a disturbance, indicating that the LS tuning failure is caused by the mismatch between target loop and feasible loops. (Legend as in Fig. 10.12)

study, but a quick remedy may be provided by identifying the plant transfer function when necessary and then determining a suitable target loop. Future work along this direction would aim to incorporate the possibility of adjusting the target loop (or the control objective) according to the observed model structure (non invertible elements/fundamental performance limitations) and the estimated uncertainty. Such a development can be beneficial in applications by reducing the conservatism of the controller tuning and increasing its reliability. A different application of the same basic concepts can also lead to the development of a general tool that continuously monitors the controller performance and detects any degradation that requires retuning. Further down the line, there would be a significant practical interest from the generalization of these principles to more complex systems, e.g., fractional order systems or classes of nonlinear systems, for which there have been several studies reported in the literature. Since these require nontrivial extensions in parametric modeling, uncertainty characterization and estimation, a detailed survey of results extends beyond the scope of this document.

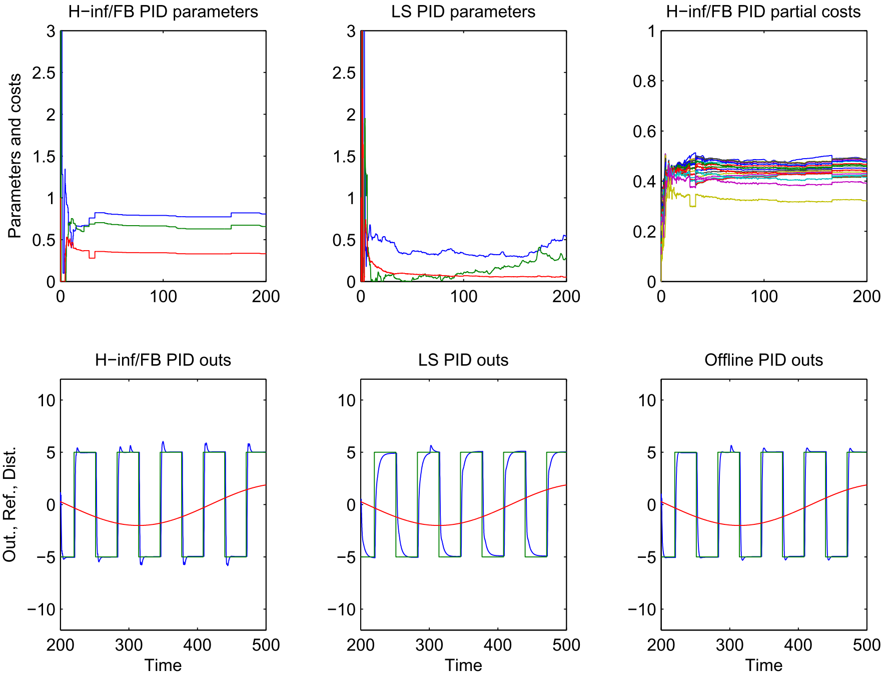


Fig. 10.15 For the plant P_2, changing the excitation at the reference to only random, results in poor LS PID tuning with significantly lower bandwidth than desired. The $\mathscr{H}_\infty$/FB estimation converges approximately to the correct result. A drift disturbance is present and bandpass filtering is used. (Legend as in Fig. 10.12)

References

1. Astrom, K.J., Hagglund, T.: PID Controllers: Theory, Design, and Tuning, 2nd edn. ISA (1995)
2. Voda, A.A., Landau, I.D.: A method for the auto-calibration of PID controllers. Automatica **31**(1), 41–53 (1995)
3. Harris, S., Mellichamp, D.: Controller tuning using optimization to meet multiple closed-loop criteria. AIChE J. **31**(3), 484–487 (1985)
4. Schei, T.S.: Automatic tuning of PID controllers based on transfer function estimation. Automatica **30**(12), 1983–1989 (1994)
5. Nishikawa, Y., Sannomiya, N., Ohta, T., Tanaka, H.: A method for auto-tuning of PID control parameters. Automatica **20**(3), 321–332 (1984)
6. Tan, W., Liu, J., Chen, T., Marquez, H.: Comparison of some well-known PID tuning formulas. Comput. Chem. Eng. **30**, 1416–1423 (2006)
7. Kristansson, B., Lennartson, B.: Evaluation and simple tuning of PID controllers with high-frequency robustness. J. Process Control **16**, 91–102 (2006)
8. de Callafon, R.A., Van den Hof, P.M.J.: Control relevant identification for H_∞-norm based performance specifications. In: Proc. 34th CDC, pp. 3498–3503 (1995)
9. Gevers, M., Anderson, B.D.O., Codrons, B.: Issues in modeling for control. In: Proc. 1998 ACC, pp. 1615–1619 (1998)
10. Reinelt, W., Garulli, A., Ljung, L.: Comparing different approaches to model error modeling in robust identification. Automatica **38**, 787–803 (2002)

11. Bombois, X., Gevers, M., Scorletti, G., Anderson, B.: Robustness analysis tools for an uncertainty set obtained by prediction error identification. Automatica **37**(10), 1629–1636 (2001)
12. Gevers, M., Bombois, X., Codrons, B., Scorletti, G., Anderson, B.: Model validation for control and controller validation in a prediction error identification framework. Automatica **39**(3), 403–427 (2003)
13. Rivera, D.E., Gaikwad, S.V.: Digital PID controller tuning using prefiltered ARX estimation. In: Proc. 31st CDC, pp. 68–69 (1992)
14. Grassi, E., Tsakalis, K.S.: PID controller tuning by frequency loop shaping. In: Proc. 35th CDC (1996)
15. Adusumilli, S., Rivera, D.E., Dash, S., Tsakalis, K.: Integrated MIMO identification and robust PID controller design through loop shaping. In: Proc. 1998 ACC, pp. 1230–1234 (1998)
16. Grimble, M.J.: $\mathscr{H}_\infty$ controllers with a PID structure. J. Dyn. Syst. Meas. Control **112**, 325–336 (1990)
17. Panagopoulos, H., Astrom, K.J., Hagglund, T.: Design of PID controllers based on constrained optimization. IEE Proc., Control Theory Appl. **149**(1), 32–40 (2002)
18. Astrom, K.J., Panagopoulos, H., Hagglund, T.: Design of PI controllers based on nonconvex optimization. Automatica **34**(5), 585–601 (1998)
19. Ender, D.: Process control performance: Not as good as you think. In: Control Engineering, pp. 180–190 (1993)
20. Gaikwad, S.V., Dash, S.K., Tsakalis, K.S., Stein, G.: Auto-tuning controller using loop-shaping. U.S. Patent No. 7,024,253, 4 April 2006
21. Grassi, E., Tsakalis, K., Dash, S., Gaikwad, S.V., MacArthur, W., Stein, G.: Integrated identification and PID controller tuning by frequency loop-shaping. IEEE Trans. Control Syst. Technol. **9**(2), 285–294 (2001)
22. Gevers, M.: Identification for control: from early achievements to the revival of experiment design. Eur. J. Control **11**, 1–18 (2005)
23. Hjalmarsson, H.: From experiment to closed-loop control. Automatica **41**, 393–438 (2005)
24. Lee, W.S., Anderson, B.D.O., Kosut, R.L., Mareels, I.M.Y.: A new approach to adaptive robust control. Int. J. Adapt. Control Signal Process. **7**(3), 183–211 (1993)
25. Lee, W.S., Anderson, B.D.O., Mareels, I.M.Y., Kosut, R.L.: On some key issues in the windsurfer approach to adaptive robust control. Automatica **31**(11), 1619–1636 (1995)
26. Ziegler, J.G., Nichols, N.B.: Optimum settings for automatic controllers. Trans. Am. Soc. Mech. Eng. **64**(11), 759–768 (1942)
27. Astrom, K.J., Hagglund, T.: A frequency domain method for automatic tuning of simple feedback loops. In: Proc. 23rd IEEE Conf. Dec. Contr., Las Vegas, NV, December 1984, pp. 299–304 (1984)
28. Gaikwad, S., Dash, S., Stein, G.: Auto-tuning PID using loop-shaping ideas. In: Proc. 1999 IEEE Intl. Conf. on Control Applications, Hawaii, August 1999, pp. 589–593 (1999)
29. Grassi, E., Tsakalis, K.S., Dash, S., Gaikwad, S.V., Stein, G.: Adaptive/self-tuning PID control by frequency loop-shaping. In: Proc. 39th IEEE Conf. on Decision and Control, Sydney, December 2000, pp. 1099–1101 (2000)
30. Jun, M., Safonov, M.G.: Automatic PID tuning: An application of unfalsified control. In: Proc. IEEE CCA/CACSD, Kohala Coast Island of Hawaii, 22–27 August 1999, pp. 328–333 (1999)
31. Morse, A.S., Mayne, D.Q., Goodwin, G.C.: Application of hysteresis switching in parameter adaptive control. IEEE Trans. Autom. Control **37**(9), 1343–1354 (1992)
32. Safonov, M.G., Tsao, T.: The unfalsified control concept and learning. IEEE Trans. Autom. Control **42**(6), 843–847 (1997)
33. Hespanha, J.P., Liberzon, D., Morse, A.S.: Hysteresis-based switching algorithms for supervisory control of uncertain systems. Automatica **39**(2), 263–272 (2003)
34. Dehghani, A., Anderson, B.D.O., Lanzon, A.: Unfalsified adaptive control: a new controller implementation and some remarks. In: Proc. of the European Control Conference 2007, Kos, Greece, 2–5 July 2007, pp. 709–716 (2007)
35. Stefanovic, M., Safonov, M.G.: Safe adaptive switching control: Stability and convergence. IEEE Trans. Autom. Control **53**(9), 2012–2021 (2008)

36. Khalil, H.: Nonlinear Systems, 3rd edn. Prentice Hall, Upper Saddle River (2002)
37. Francis, B.: A Course in H_∞ Control Theory. Springer, Berlin (1987)
38. Barnes, T.J.D., Wang, L., Cluett, W.R.: A frequency domain design method for PID controllers. In: Proc. Amer. Contr. Conf., San Francisco, pp. 890–894 (1993)
39. Wang, L., Barnes, T.J.D., Cluett, W.R.: A new frequency domain design method for PID controllers. IEE Proc., Control Theory Appl. **142**(4), 265–271 (1995)
40. Rivera, D.E., Morari, M., Skogestad, S.: Internal model control 4. PID controller design. Ind. Eng. Chem. Process Des. Dev. **25**, 252–265 (1986)
41. Grassi, E., Tsakalis, K.: PID controller tuning by frequency loop-shaping: Application to diffusion furnace temperature control. IEEE Trans. Control Syst. Technol. **8**(5), 842–847 (2000)
42. Ljung, L.: System Identification. Prentice Hall, New York (1987)
43. Vidyasagar, M.: Nonlinear Systems Analysis, 2nd edn. Prentice Hall, New York (1987)
44. Safonov, M.G.: Stability margins of diagonally perturbed multivariable feedback systems. Proc. IEEE **129**(6), 251–256 (1982)
45. Adusumilli, S., Rivera, D.E., Dash, S., Tsakalis, K.: Integrated identification and robust PID controller design through loop shaping for multi-input multi-output processes. In: Proc. ACC, June 1998, pp. 1230–1234 (1998)
46. McFarlane, D.C., Glover, K.: A loop shaping design procedure using H_∞ synthesis. IEEE Trans. Autom. Control **37**(6), 759–769 (1992)
47. Alexander, C., Tsakalis, K.S.: Control of an inverted pendulum: A classical experiment revisited. In: Proc. of 1995 Soc. for Comp. Sim. Western Multi-Conference, Las Vegas, NV, January 1995
48. Papadopoulos, A., Tsakalis, K.: Adaptive control of an inverted pendulum. In: Proc. 23rd IASTED int'l conf. on modeling, identification and control, Grindelwald, Switzerland, February 2004, vol. 412-051, pp. 461–466 (2004)
49. Limanond, S., Si, J., Tsakalis, K.S.: Monitoring and control of semiconductor manufacturing processes. IEEE Control Syst. **18**(6), 46–58 (1998)
50. Tsakalis, K.S., Stoddard, K.D.: Integrated identification and control for diffusion/CVD furnaces. In: Proc. 6th IEEE Int. Conference on Emerging Technologies and Factory Automation, Los Angeles, September 1997, pp. 514–519 (1997)
51. Tsakalis, K., Flores-Godoy, J.-J., Stoddard, K.: Temperature control of diffusion/CVD furnaces using robust multivariable loop-shaping techniques. In: Proc. 38th Conf. Decision and Contr., Phoenix, AZ, December 1999, vol. 4, pp. 4192–4197 (1999)
52. McCormack, A.S., Godfrey, K.R.: Rule-Based autotuning based on frequency domain identification. IEEE Trans. Control Syst. Technol. **6**(1), 43–61 (1998)
53. Fox, P.D., Godfrey, K.R.: Multiharmonic perturbations for nonparametric autotuning. IEE Proc., Control Theory Appl. **146**, 1–8 (1999)
54. Zhu, Y.-C.: Identification and control of MIMO industrial processes: An integration approach. Ph.D. dissertation. Eindhoven Univ. of Technology, Dept. of Electr. Engr., The Netherlands (1990)
55. Ioannou, P.A., Sun, J.: Robust Adaptive Control. Prentice Hall, New York (1996)
56. Grassi, E., Tsakalis, K., Dash, S., Gaikwad, S., Stein, G.: Adaptive/self-tuning PID control by frequency loop-shaping. In: Proc. IEEE 39th Conf. Decision and Control, Sydney, 12–15 December 2000, vol. 2, pp. 1099–1101 (2000)
57. Krause, J.M., Khargonekar, P.P.: Robust parameter adjustment. In: Proc. 1988 ACC, Atlanta, GA, pp. 331–336 (1988)
58. Tsakalis, K., Dash, S.: Adaptive PID control using Filter-Banks and frequency loop shaping. In: European Control Conference 2007, Kos, Greece, 2–5 July 2007, pp. 1340–1347 (2007)
59. Astrom, K.J., Rundkwist, L.: Integrator wind up and how to avoid it. In: Proc. 1989 ACC, pp. 1693–1698 (1989)
60. Teel, A.R., Kapoor, N.: The L-2 anti-windup problem: its definition and solution. In: Proc. of Europ. Contr. Conf. (1997)

Chapter 11
Modern PID Control: Stabilizing Sets and Multiple Performance Specifications

L.H. Keel and S.P. Bhattacharyya

11.1 Introduction

Since its introduction [1], PID control has been one of the most widely used methodologies in industrial control systems [2]. Despite its popularity, the main focus of the research in this area is largely concentrated in developing methodologies of tuning three PID control parameters that include manual tuning, Ziegler–Nichols [3], Cohen–Coon [4], and tuning by using software tools [2]. On the other hand, the new approach is based on characterizing the complete set of stabilizing controllers [5–9]. Once the entire set of stabilizing controllers is found, the control design method proceeds to find a subset satisfying additional design constraints. It is shown that the problem of determining the entire set of stabilizing controllers is reduced to finding the set of solutions for a set of linear inequalities. It is further shown that some important performance measures of the control system such as gain margin, phase margin, and H_∞ margin requirements are also implemented as a sequence of linear programming problems. In addition, time domain specifications such as overshoot and rise time can also be searched for within this set. The stabilizing set also provides the information regarding the robustness and/or fragility of the controller selected [10].

In later sections of this chapter, it is shown that the PID synthesis described can be carried out directly from frequency response measurements on the plant without constructing a state space or transfer function model [6, 7]. The results show that complete sets achieving stability and various performance specifications can be obtained from Nyquist/Bode data without constructing an identified model. A notable

L.H. Keel (✉)
Tennessee State University, Nashville, TN 37209, USA
e-mail: keel@gauss.tsuniv.edu

S.P. Bhattacharyya
Texas A&M University, College Station, TX 77843, USA
e-mail: bhatt@ece.tamu.edu

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_11, © Springer-Verlag London Limited 2012

feature is that the solution does not require knowledge of the order of the system. It will be seen in the sequel that the solution specifies an exact "low frequency band" where the plant frequency response must be known accurately and beyond which rough data or measurements suffice. These features have important implications in real world control engineering, where models are often unavailable, measurements can be made only over a restricted range of frequencies and where guarantees of various performance specifications must still be made.

Before giving the concluding remarks, we devote a section giving a discussion of some sharp distinctions between model-based and data-based designs for high order systems with an example. This discussion clearly illustrates that the data-based approach given here is a significant alternative to the traditional model-based design of controllers.

11.2 Signature, Root Counting, and Nyquist–Bode Equivalence

In this section, we develop some notation and technical results which will be used later.

11.2.1 Signature Formula for Real Polynomials

Let $p(s)$ denote a polynomial of degree n with real coefficients and without zeros on the $j\omega$ axis. Write

$$p(s) := \underbrace{p_0 + p_2 s^2 + \cdots}_{p_{\text{even}}(s^2)} + s\underbrace{\left(p_1 + p_3 s^2 + \cdots\right)}_{p_{\text{odd}}(s^2)} \tag{11.1}$$

so that

$$p(j\omega) = p_r(\omega) + jp_i(\omega) \tag{11.2}$$

where $p_r(\omega)$, $p_i(\omega)$ are polynomials in ω with real coefficients with

$$p_r(\omega) = p_{\text{even}}\left(-\omega^2\right), \qquad p_i(\omega) = \omega p_{\text{odd}}\left(-\omega^2\right). \tag{11.3}$$

Consider the standard signum function $\text{sgn} : \mathcal{R} \to \{-1, 0, 1\}$ is defined by

$$\text{sgn}[x] = \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases} \tag{11.4}$$

Let $p(s)$ be a given polynomial of degree n with real coefficients and without zeros on the imaginary axis. Let $\mathcal{C}^-$ denote the open left-half plane (LHP), $\mathcal{C}^+$ the open right-half plane (RHP), and l and r the numbers of roots of $p(s)$ in $\mathcal{C}^-$ and $\mathcal{C}^+$, respectively. Let $\angle p(j\omega)$ denote the angle of $p(j\omega)$ and $\Delta_{\omega_1}^{\omega_2} \angle p(j\omega)$ the net change, in radians, in the phase or angle of $p(j\omega)$ as ω runs from ω_1 to ω_2.

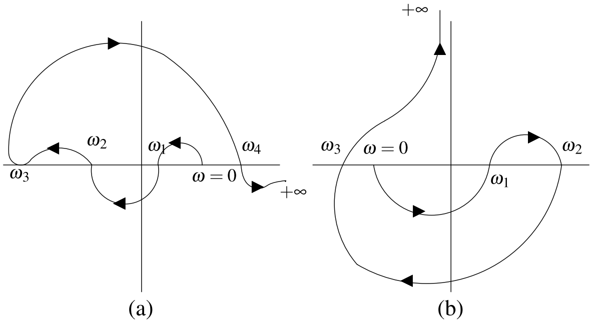


Fig. 11.1 (**a**) Plot of $p(j\omega)$ for $p(s)$ of even degree. (**b**) Plot of $p(j\omega)$ for $p(s)$ of odd degree

Lemma 11.1

$$\Delta_0^\infty \angle p(j\omega) = \frac{\pi}{2}(l - r). \tag{11.5}$$

Proof Each LHP root contributes π and each RHP root contributes $-\pi$ to the net change in phase of $p(j\omega)$ as ω runs from $-\infty$ to ∞, and (11.5) follows from the symmetry about the real axis of the roots since $p(s)$ has real coefficients. □

We call $l - r$, the Hurwitz *signature* of $p(s)$, and denote it as:

$$\sigma(p) := l - r. \tag{11.6}$$

Computation of $\sigma(p)$ By Lemma 11.1, the computation of $\sigma(p)$ amounts to a determination of the total phase change of $p(j\omega)$. To see how the total phase change may be calculated, consider typical plots of $p(j\omega)$ where ω runs from 0 to $+\infty$ as in Fig. 11.1. We note that the frequencies $0, \omega_1, \omega_2, \omega_3, \omega_4$ are the points where the plot cuts or touches the real axis. In Fig. 11.1(a), ω_3 is a point where the plot *touches* but does not cut the real axis.

In Fig. 11.1(a), we have

$$\Delta_0^\infty \angle p(j\omega) = \underbrace{\Delta_0^{\omega_1} \angle p(j\omega)}_{0} + \underbrace{\Delta_{\omega_1}^{\omega_2} \angle p(j\omega)}_{-\pi} + \underbrace{\Delta_{\omega_2}^{\omega_3} \angle p(j\omega)}_{0} + \cdots . \tag{11.7}$$

Observe that

$$\begin{aligned}
\Delta_0^{\omega_1} \angle p(j\omega) &= \operatorname{sgn}\big[p_i\big(0^+\big)\big]\big(\operatorname{sgn}\big[p_r(0) - \operatorname{sgn}\big[p_r(\omega_1)\big]\big)\frac{\pi}{2}, \\
\Delta_{\omega_1}^{\omega_2} \angle p(j\omega) &= \operatorname{sgn}\big[p_i\big(\omega_1^+\big)\big]\big(\operatorname{sgn}\big[p_r(\omega_1) - \operatorname{sgn}\big[p_r(\omega_2)\big]\big)\frac{\pi}{2}, \\
\Delta_{\omega_2}^{\omega_3} \angle p(j\omega) &= \operatorname{sgn}\big[p_i\big(\omega_2^+\big)\big]\big(\operatorname{sgn}\big[p_r(\omega_2) - \operatorname{sgn}\big[p_r(\omega_3)\big]\big)\frac{\pi}{2}, \\
&\vdots
\end{aligned} \tag{11.8}$$

and

$$\begin{aligned}
\operatorname{sgn}\big[p_i\big(\omega_1^+\big)\big] &= -\operatorname{sgn}\big[p_i\big(0^+\big)\big],\\
\operatorname{sgn}\big[p_i\big(\omega_2^+\big)\big] &= -\operatorname{sgn}\big[p_i\big(\omega_1^+\big)\big] = +\operatorname{sgn}\big[p_i\big(0^+\big)\big],\\
\operatorname{sgn}\big[p_i\big(\omega_3^+\big)\big] &= +\operatorname{sgn}\big[p_i\big(\omega_2^+\big)\big] = +\operatorname{sgn}\big[p_i\big(0^+\big)\big],\\
\operatorname{sgn}\big[p_i\big(\omega_4^+\big)\big] &= -\operatorname{sgn}\big[p_i\big(\omega_3^+\big)\big] = -\operatorname{sgn}\big[p_i\big(0^+\big)\big],
\end{aligned} \tag{11.9}$$

and note also that 0, ω_1, ω_2, ω_4 are the real zeros of $p_i(\omega)$ of *odd* multiplicity whereas ω_3 is a real zero of *even* multiplicity. From these relations, it is evident that (11.7) may be rewritten, skipping the terms involving ω_3, the root of even multiplicity, so that

$$\begin{aligned}
\Delta_0^\infty \angle p(j\omega) &= \Delta_0^{\omega_1}\angle p(j\omega) + \Delta_{\omega_1}^{\omega_2}\angle p(j\omega) + \Delta_{\omega_2}^{\omega_4}\angle p(j\omega) + \Delta_{\omega_4}^{\infty}\angle p(j\omega)\\
&= \frac{\pi}{2}\Big(\operatorname{sgn}\big[p_i\big(0^+\big)\big]\big(\operatorname{sgn}\big[p_r(0)\big] - \operatorname{sgn}\big[p_r(\omega_1)\big]\big)\\
&\quad - \operatorname{sgn}\big[p_i\big(0^+\big)\big]\big(\operatorname{sgn}\big[p_r(\omega_1)\big]\\
&\quad - \operatorname{sgn}\big[p_r(\omega_2)\big]\big) + \operatorname{sgn}\big[p_i\big(0^+\big)\big]\big(\operatorname{sgn}\big[p_r(\omega_2)\big] - \operatorname{sgn}\big[p_r(\omega_4)\big]\big)\\
&\quad - \operatorname{sgn}\big[p_i\big(0^+\big)\big]\big(\operatorname{sgn}\big[p_r(\omega_4)\big] - \operatorname{sgn}\big[p_r(\infty)\big]\big)\Big).
\end{aligned} \tag{11.10}$$

Equation (11.10) can be rewritten as

$$\begin{aligned}
\Delta_0^\infty \angle p(j\omega) &= \frac{\pi}{2}\operatorname{sgn}\big[p_i\big(0^+\big)\big]\big(\operatorname{sgn}\big[p_r(0)\big] - 2\operatorname{sgn}\big[p_r(\omega_1)\big] + 2\operatorname{sgn}\big[p_r(\omega_2)\big]\\
&\quad - 2\operatorname{sgn}\big[p_r(\omega_4)\big] + \operatorname{sgn}\big[p_r(\infty)\big]\big).
\end{aligned} \tag{11.11}$$

In the case of Fig. 11.1(b), that is, when $p(s)$ is of odd degree, we have

$$\Delta_0^\infty \angle p(j\omega) = \underbrace{\Delta_0^{\omega_1}\angle p(j\omega)}_{+\pi} + \underbrace{\Delta_{\omega_1}^{\omega_2}\angle p(j\omega)}_{0} + \underbrace{\Delta_{\omega_2}^{\omega_3}\angle p(j\omega)}_{-\pi} + \underbrace{\Delta_{\omega_3}^{+\infty}\angle p(j\omega)}_{-\frac{\pi}{2}}, \tag{11.12}$$

and $\Delta_0^{\omega_1}\angle p(j\omega)$, $\Delta_{\omega_1}^{\omega_2}\angle p(j\omega)$, $\Delta_{\omega_2}^{\omega_3}\angle p(j\omega)$ are as before whereas

$$\Delta_{\omega_3}^{\infty}\angle p(j\omega) = \frac{\pi}{2}\operatorname{sgn}\big[p_i\big(\omega_3^+\big)\big]\operatorname{sgn}\big[p_r(\omega_3)\big]. \tag{11.13}$$

We also have, as before,

$$\operatorname{sgn}\big[p_i\big(\omega_j^+\big)\big] = (-1)^j\operatorname{sgn}\big[p_i\big(0^+\big)\big], \quad j = 1, 2, 3. \tag{11.14}$$

Combining (11.12)–(11.14), we have, finally, for Fig. 11.1(b),

$$\begin{aligned}
\Delta_0^\infty \angle p(j\omega) &= \frac{\pi}{2}\operatorname{sgn}\big[p_i\big(0^+\big)\big]\big(\operatorname{sgn}\big[p_r(0)\big] - 2\operatorname{sgn}\big[p_r(\omega_1)\big] + 2\operatorname{sgn}\big[p_r(\omega_2)\big]\\
&\quad - 2\operatorname{sgn}\big[p_r(\omega_3)\big]\big).
\end{aligned} \tag{11.15}$$

We can now easily generalize the above formulas for the signature, based on Lemma 11.1.

Theorem 11.1 *Let $p(s)$ be a polynomial of degree n with real coefficients, without zeros on the imaginary axis. Write $p(j\omega) = p_r(\omega) + jp_i(\omega)$ and let $\omega_0, \omega_1, \omega_3, \dots, \omega_{l-1}$ denote the real nonnegative zeros of $p_i(\omega)$ with odd multiplicities with $\omega_0 = 0$. Then*

- *If n is even,*

$$\sigma(p) = \operatorname{sgn}\bigl[p_i\bigl(0^+\bigr)\bigr]\Biggl(\operatorname{sgn}\bigl[p_r(0)\bigr] + 2\sum_{j=1}^{l-1}(-1)^j \operatorname{sgn}\bigl[p_r(\omega_j)\bigr] \\ + (-1)^l \operatorname{sgn}\bigl[p_r(\infty)\bigr]\Biggr).$$

- *If n is odd,*

$$\sigma(p) = \operatorname{sgn}\bigl[p_i\bigl(0^+\bigr)\bigr]\Biggl(\operatorname{sgn}\bigl[p_r(0)\bigr] + 2\sum_{j=1}^{l-1}(-1)^j \operatorname{sgn}\bigl[p_r(\omega_j)\bigr]\Biggr).$$

11.2.2 Signature Formulas for Complex Polynomials

Consider the polynomial $c(s)$ with complex coefficients and suppose that $c(s)$ has no $j\omega$ axis zeros and has l and r open LHP and open RHP roots. As before, we have the total phase change

$$\Delta_{-\infty}^{+\infty}\angle c(j\omega) = \pi(l - r) \tag{11.16}$$

and we define the signature

$$\sigma(c) := l - r. \tag{11.17}$$

To compute the signature, write

$$c(j\omega) = p(\omega) + jq(\omega) \tag{11.18}$$

where $p(\omega)$ and $q(\omega)$ are polynomials with real coefficients. Let $\omega_1, \omega_2, \dots, \omega_{l-1}$ denote the real zeros of $q(\omega)$ of odd multiplicities, $\omega_0 = -\infty$, $\omega_l = +\infty$ and $\omega_0 < \omega_1 < \cdots < \omega_{l-1} < \omega_l$. Write

$$j_- = \operatorname{sgn}\bigl[q(-\infty)\bigr], \qquad j_+ = \operatorname{sgn}\bigl[q(\infty)\bigr], \tag{11.19}$$

and

$$i_k = \operatorname{sgn}\bigl[p(\omega_k)\bigr], \quad \text{for } k = 0, 1, \dots, l. \tag{11.20}$$

Theorem 11.2

(a) *If* $\deg[p] > \deg[q]$,

$$\sigma(c) = \frac{1}{2} j_- \{i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1} 2i_{l-1} + (-1)^l i_l\} \tag{11.21}$$

$$= \frac{1}{2} j_+ (-1)^{l-1} \{i_0 - 2i_1 + \cdots + (-1)^{l-1} 2i_{l-1} + (-1)^l i_l\}. \tag{11.22}$$

(b) *If* $\deg[q] \geq \deg[p]$,

$$\sigma(c) = j_- \{-i_1 + i_2 - i_3 + \cdots + (-1)^{l-1} i_{l-1}\} \tag{11.23}$$

$$= j_+ (-1)^{l-1} \{-i_1 + i_2 - i_3 + \cdots + (-1)^{l-1} i_{l-1}\}. \tag{11.24}$$

Proof In case (a), the complex plane plot of $c(j\omega)$ approaches the real axis as $|\omega| \to \infty$. Thus,

$$\Delta_{-\infty}^{+\infty} \angle c(j\omega) = \Delta_{-\infty}^{\omega_1} \angle c(j\omega) + \Delta_{\omega_1}^{\omega_2} \angle c(j\omega) + \cdots + \Delta_{\omega_{l-1}}^{+\infty} \angle c(j\omega) \tag{11.25}$$

and

$$\Delta_{-\infty}^{\omega_1} \angle c(j\omega) = \frac{\pi}{2} j_- (i_0 - i_1), \tag{11.26}$$

$$\Delta_{\omega_k}^{\omega_{k+1}} \angle c(j\omega) = \frac{\pi}{2} j_- (-1)^k (i_k - i_{k+1}), \quad k = 0, 1, \ldots, l-1. \tag{11.27}$$

Substituting (11.26), (11.27) into (11.25) and using (11.16), (11.17), we get (11.21); (11.22) follows from

$$j_- = j_+ (-1)^{l-1}. \tag{11.28}$$

In case (b),

$$\begin{aligned} \Delta_{\omega_1}^{\omega_2} \angle c(j\omega) &= -j_- (i_1 - i_2) \frac{\pi}{2}, \\ \Delta_{\omega_2}^{\omega_3} \angle c(j\omega) &= +j_- (i_2 - i_3) \frac{\pi}{2}, \\ &\vdots \\ \Delta_{\omega_{l-2}}^{\omega_{l-1}} \angle c(j\omega) &= (-1)^{l-2} j_- (i_{l-2} - i_{l-1}) \frac{\pi}{2}, \end{aligned} \tag{11.29}$$

whereas

$$\Delta_{-\infty}^{\omega_1} \angle c(j\omega) + \Delta_{\omega_{l-1}}^{+\infty} \angle c(j\omega) = j_- \big(-i_1 + (-1)^{l-1} i_{l-1}\big) \frac{\pi}{2}. \tag{11.30}$$

Combining (11.29) and (11.30) with (11.16) and (11.17), we obtain (11.23); (11.24) again follows from

$$j_- = j_+ (-1)^{l-1}. \tag{11.31}$$

□

11.2.3 Signature Formula for Rational Functions

Consider a real rational function

$$R(s) = \frac{A(s)}{B(s)} \tag{11.32}$$

where $A(s)$ and $B(s)$ are polynomials with real coefficients and of degrees m and n, respectively. We assume that $A(s)$ and $B(s)$ have no zeros on the $j\omega$ axis. Let z_R^+, p_R^+ (z_R^-, p_R^-) denote the numbers of open right half-plane (RHP) and open left half-plane (LHP) zeros and poles of $R(s)$. Also let $\Delta_0^\infty \angle R(j\omega)$ denote the net change in phase of $R(j\omega)$ as ω runs from 0 to $+\infty$. Then we have

$$\Delta_0^\infty \angle R(j\omega) = \frac{\pi}{2}\left[z_R^- - z_R^+ - \left(p_R^- - p_R^+\right)\right]. \tag{11.33}$$

This formula follows from the fact that each LHP zero and each RHP pole contribute $+\frac{\pi}{2}$ to the net phase change whereas each RHP zero and LHP pole contribute $-\frac{\pi}{2}$ to the net phase change.

For convenience, we define the (Hurwitz) signature of $R(s)$ as

$$\sigma(R) := z_R^- - z_R^+ - \left(p_R^- - p_R^+\right). \tag{11.34}$$

Write

$$R(j\omega) = R_r(\omega) + jR_i(\omega) \tag{11.35}$$

where $R_r(\omega)$ and $R_i(\omega)$ are rational functions in ω with real coefficients. It is easy to see that $R_r(\omega)$ and $R_i(\omega)$ have no real poles for $\omega \in (-\infty, +\infty)$ since $R(s)$ has no imaginary axis poles. To compute the net change in phase, that is, the left-hand side of (11.33), it is convenient to develop formulas in terms of $R_r(\omega)$ and $R_i(\omega)$. Note that $\omega_0 = 0$ is always a zero of $R_i(\omega)$ since $R(s)$ is real. Let

$$0 = \omega_0 < \omega_1 < \omega_2 < \cdots < \omega_{l-1} \tag{11.36}$$

denote the real, finite non-negative zeros of $R_i(\omega) = 0$ of odd multiplicities, and define $\omega_l = \infty^-$. Define, for a real function $f(t)$,

$$f\left(t_0^-\right) := \lim_{t \to t_0, t < t_0} f(t), \qquad f\left(t_0^+\right) := \lim_{t \to t_0, t > t_0} f(t). \tag{11.37}$$

Lemma 11.2 (Real Hurwitz Signature Lemma) *For $n - m$ even,*

$$\sigma(R) = \left(\operatorname{sgn}\left[R_r(\omega_0)\right] + 2\sum_{j=1}^{l-1}(-1)^j \operatorname{sgn}\left[R_r(\omega_j)\right] + (-1)^l \operatorname{sgn}\left[R_r(\omega_l)\right]\right) \cdot (-1)^{l-1}\operatorname{sgn}\left[R_i\left(\infty^-\right)\right].$$

For $n-m$ odd,

$$\sigma(R)=\left(\operatorname{sgn}\left[R_r(\omega_0)\right]+2\sum_{j=1}^{l-1}(-1)^j\operatorname{sgn}\left[R_r(\omega_j)\right]\right)(-1)^{l-1}\operatorname{sgn}\left[R_i\left(\infty^-\right)\right].$$

The proof of this formula is found in [7]. Now consider a complex rational function

$$Q(s)=\frac{D(s)}{E(s)} \tag{11.38}$$

where $D(s)$ and $E(s)$ are polynomials with complex coefficients of degrees n and m, respectively. As before, we assume that $D(s)$ and $E(s)$ do not have zeros on the $j\omega$ axis. Let $\Delta_{-\infty}^{+\infty}\angle Q(j\omega)$ denote the net change in phase of $Q(j\omega)$ as ω runs from $-\infty$ to $+\infty$. Also let z_Q^+, p_Q^+ (z_Q^-, p_Q^-) denote the numbers of RHP (LHP) zeros and poles of $Q(s)$. Then we have

$$\Delta_{-\infty}^{+\infty}\angle Q(j\omega)=\pi\left[z_Q^- - z_Q^+ - \left(p_Q^- - p_Q^+\right)\right]. \tag{11.39}$$

This easily follows from the fact that each LHP zero (RHP zero) and each RHP pole (LHP pole) contribute $+\pi$ $(-\pi)$ to the net phase change. Summing over all poles and zeros we obtain the formula given. Analogous to the real case, we define the signature of the complex rational function $Q(s)$:

$$\sigma(Q):=z_Q^- - z_Q^+ - \left(p_Q^- - p_Q^+\right). \tag{11.40}$$

Write

$$Q(j\omega)=Q_r(\omega)+jQ_i(\omega) \tag{11.41}$$

where $Q_r(\omega)$ and $Q_i(\omega)$ are rational functions with real coefficients. Moreover, $Q_r(\omega)$ and $Q_i(\omega)$ have no real poles for $\omega\in(-\infty,+\infty)$. It is easy to show that the numerators of $Q_r(\omega)$ and $Q_i(\omega)$ are generically of the same degree. Indeed, if this is not so, multiplying $Q(s)$ by an arbitrary complex number $\alpha+j\beta$ will render this condition to be true without changing poles, zeros or signature of $Q(s)$. Therefore, we henceforth assume this to be true. Let $\omega_0,\dots,\omega_{l-1}$ ordered as

$$-\infty<\omega_0<\omega_1<\cdots<\omega_{l-1}<+\infty:=\omega_l$$

denote the real, distinct, finite zeros of $Q_i(\omega)=0$ with odd multiplicities.

Lemma 11.3 (Complex Hurwitz Signature Lemma)

$$\sigma(Q)=\left(\sum_{j=1}^{l-1}(-1)^{l-1-j}\operatorname{sgn}\left[Q_r(\omega_j)\right]\right)\operatorname{sgn}\left[Q_i\left(\infty^-\right)\right].$$

The proof is found in [7].

11.3 Characterization of All Stabilizing PID Controllers

11.3.1 Computation of the PID Stabilizing Set

Consider the plant, with rational transfer function

$$P(s)=\frac{N(s)}{D(s)}$$

with the PID feedback controller

$$C(s)=\frac{k_p s+k_i+k_d s^2}{s(1+sT)},\quad T>0.$$

The closed-loop characteristic polynomial is

$$\delta(s)=sD(s)(1+sT)+\big(k_p s+k_i+k_d s^2\big)N(s). \tag{11.42}$$

We form the new polynomial

$$\nu(s):=\delta(s)N(-s) \tag{11.43}$$

and note that the even-odd decomposition of $\nu(s)$ is of the form:

$$\nu(s)=\nu_{\text{even}}\big(s^2,k_i,k_d\big)+s\nu_{\text{odd}}\big(s^2,k_p\big). \tag{11.44}$$

The polynomial $\nu(s)$ exhibits the *parameter separation property*, namely, that k_p appears only in the odd part and k_i, k_d only in the even part. This will facilitate the computation of the stabilizing set using signature concepts.

Let $\deg[D(s)]=n$, $\deg[N(s)]=m\le n$, and let z^+ and z^- denote the number of RHP and LHP zeros of the plant, respectively, that is, zeros of $N(s)$. We assume, as a convenient technical assumption, that the plant has no $j\omega$ axis zeros.

Theorem 11.3 *The closed-loop system is stable if and only if*

$$\sigma(\nu)=n-m+2+2z^+. \tag{11.45}$$

Proof Closed-loop stability is equivalent to the requirement that the $n+2$ zeros of $\delta(s)$ lie in the open LHP. This is equivalent to $\sigma(\delta)=n+2$ and to

$$\sigma(\nu)=n+2+z^+-z^-=n+2+z^+-\big(m-z^+\big)=(n-m)+2+2z^+.\quad\square$$

Based on this, we can develop the following procedure to calculate $\mathcal{S}^0$, the stabilizing set:

(a) First, fix $k_p=k_p^*$ and let $0<\omega_1<\omega_2<\cdots<\omega_{l-1}$ denote the real, positive, finite frequencies which are zeros of

$$\nu_{\text{odd}}\big(-\omega^2,k_p^*\big)=0 \tag{11.46}$$

of odd multiplicities. Let $\omega_0:=0$ and $\omega_l:=\infty$.

(b) Write

$$j = \operatorname{sgn}\bigl[v_{\text{odd}}\bigl(0^+, k_p^*\bigr)\bigr]$$

and determine strings of integers, $i_0, i_1, \dots$ such that:

- If $n+m$ is even,

$$j\bigl(i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1} 2i_{l-1} + (-1)^l i_l\bigr) = n - m + 2 + 2z^+. \quad (11.47)$$

- If $n+m$ is odd,

$$j\bigl(i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1} 2i_{l-1}\bigr) = n - m + 2 + 2z^+. \quad (11.48)$$

(c) Let $I_1, I_2, I_3, \dots$ denote distinct strings $\{i_0, i_1, \dots\}$ satisfying (11.47) or (11.48). Then the stabilizing sets in k_i, k_d space, for $k_p = k_p^*$ are given by the linear inequalities

$$\nu_{\text{even}}\bigl(-\omega_t^2, k_i, k_d\bigr) i_t > 0 \quad (11.49)$$

where the i_t range over each of the strings $I_1, I_2, \dots$.

(d) For each string I_j, (11.49) generates a convex stability set $\mathcal{S}_j(k_p^*)$ and the complete stabilizing set for fixed k_p^* is the union of these convex sets

$$\mathcal{S}\bigl(k_p^*\bigr) = \bigcup_j \mathcal{S}_j\bigl(k_p^*\bigr). \quad (11.50)$$

(e) The complete stabilizing set in (k_p, k_i, k_d) space can be found by sweeping k_p over the real axis and repeating the calculations (11.46)–(11.50). From (11.47) and (11.48), we can see that the range of sweeping can be restricted to those values such that the number of roots $l-1$ can satisfy (11.47) or (11.48) in the most favorable case. For $n+m$ even, this requires that

$$2 + 2(l-1) \geq n - m + 2 + 2z^+ \quad \text{or} \quad l - 1 \geq \frac{n - m + 2z^+}{2}, \quad (11.51)$$

and for $n+m$ odd, we need

$$1 + 2(l-1) \geq n - m + 2 + 2z^+ \quad \text{or} \quad l - 1 \geq \frac{n - m + 1 + 2z^+}{2}. \quad (11.52)$$

Thus, k_p needs to be swept over those ranges where (11.46) is satisfied with $l-1$ given by (11.51) or (11.52).

Remark 11.1 If the PID controller with pure derivative action is used ($T = 0$) it is easy to see that the signature requirement for stability becomes

$$\sigma(\nu) = n - m + 1 + 2z^+.$$

The following example illustrates the detailed calculations involved in determining the stabilizing (k_p, k_i, k_d) gain values.

Example 11.1 Consider the problem of determining stabilizing PID gains for the plant $P(s) = \frac{N(s)}{D(s)}$ where

$$N(s) = s^3 - 2s^2 - s - 1, \qquad D(s) = s^6 + 2s^5 + 32s^4 + 26s^3 + 65s^2 - 8s + 1.$$

In this example, we use the PID controller with $T = 0$. The closed-loop characteristic polynomial is

$$\delta(s, k_p, k_i, k_d) = sD(s) + \big(k_i + k_d s^2\big)N(s) + k_p s N(s).$$

Here $n = 6$ and $m = 3$

$$\begin{aligned} N_e\big(s^2\big) &= -2s^2 - 1, & N_o\big(s^2\big) &= s^2 - 1, \\ D_e\big(s^2\big) &= s^6 + 32s^4 + 65s^2 + 1, & D_o\big(s^2\big) &= 2s^4 + 26s^2 - 8, \end{aligned}$$

and

$$N(-s) = \big(-2s^2 - 1\big) - s\big(s^2 - 1\big).$$

Therefore, we obtain

$$\begin{aligned} \nu(s) &= \delta(s, k_p, k_i, k_d)N(-s) \\ &= \big[s^2\big(-s^8 - 35s^6 - 87s^4 + 54s^2 + 9\big) + \big(k_i + k_d s^2\big)\big(-s^6 + 6s^4 + 3s^2 + 1\big)\big] \\ &\quad + s\big[\big(-4s^8 - 89s^6 - 128s^4 - 75s^2 - 1\big) + k_p\big(-s^6 + 6s^4 + 3s^2 + 1\big)\big] \end{aligned}$$

so that

$$\nu(j\omega, k_p, k_i, k_d) = \big[p_1(\omega) + \big(k_i - k_d\omega^2\big)p_2(\omega)\big] + j\big[q_1(\omega) + k_p q_2(\omega)\big]$$

where

$$\begin{aligned} p_1(\omega) &= \omega^{10} - 35\omega^8 + 87\omega^6 + 54\omega^4 - 9\omega^2, \\ p_2(\omega) &= \omega^6 + 6\omega^4 - 3\omega^2 + 1, \\ q_1(\omega) &= -4\omega^9 + 89\omega^7 - 128\omega^5 + 75\omega^3 - \omega, \\ q_2(\omega) &= \omega^7 + 6\omega^5 - 3\omega^3 + \omega. \end{aligned}$$

We find that $z^+ = 1$ so that the signature requirement on $\nu(s)$ for stability is

$$\sigma(\nu) = n - m + 1 + 2z^+ = 6.$$

Since the degree of $\nu(s)$ is even, we see from the signature formulas that $q(\omega)$ must have at least two positive real roots of odd multiplicity. The range of k_p such that $q(\omega, k_p)$ has at least 2 real, positive, distinct, finite zeros with odd multiplicities was determined to be $(-24.7513, 1)$ which is the *allowable* range for k_p. For a fixed $k_p \in (-24.7513, 1)$, for instance, $k_p = -18$, we have

$$q(\omega, -18) = q_1(\omega) - 18q_2(\omega) = -4\omega^9 + 71\omega^7 - 236\omega^5 + 129\omega^3 - 19\omega.$$

The real, nonnegative, distinct finite zeros of $q(\omega, -18)$ with odd multiplicities are

$$\omega_0 = 0, \qquad \omega_1 = 0.5195, \qquad \omega_2 = 0.6055, \qquad \omega_3 = 1.8804, \qquad \omega_4 = 3.6848.$$

Also define $\omega_5 = \infty$. Since $\operatorname{sgn}[q(0, -18)] = -1$, it follows that every *admissible* string $\mathcal{I} = \{i_0, i_1, i_2, i_3, i_4, i_5\}$ must satisfy

$$\{i_0 - 2i_1 + 2i_2 - 2i_3 + 2i_4 - i_5\} \cdot (-1) = 6.$$

Hence, the admissible strings are

$$\begin{aligned} \mathcal{I}_1 &= \{-1, -1, -1, 1, -1, 1\}, & \mathcal{I}_2 &= \{-1, 1, 1, 1, -1, 1\}, \\ \mathcal{I}_3 &= \{-1, 1, -1, -1, -1, 1\}, & \mathcal{I}_4 &= \{-1, 1, -1, 1, 1, 1\}, \\ \mathcal{I}_5 &= \{1, 1, -1, 1, -1, -1\}. \end{aligned}$$

For $\mathcal{I}_1$ it follows that the stabilizing (k_i, k_d) values corresponding to $k_p = -18$ must satisfy the string of inequalities:

$$\begin{aligned} p_1(\omega_0) + \big(k_i - k_d\omega_0^2\big)p_2(\omega_0) &< 0, & p_1(\omega_1) + \big(k_i - k_d\omega_1^2\big)p_2(\omega_1) &< 0, \\ p_1(\omega_2) + \big(k_i - k_d\omega_2^2\big)p_2(\omega_2) &< 0, & p_1(\omega_3) + \big(k_i - k_d\omega_3^2\big)p_2(\omega_3) &> 0, \\ p_1(\omega_4) + \big(k_i - k_d\omega_4^2\big)p_2(\omega_4) &< 0, & p_1(\omega_5) + \big(k_i - k_d\omega_5^2\big)p_2(\omega_5) &> 0. \end{aligned}$$

Substituting for ω_0, ω_1, ω_2, ω_3, ω_4, and ω_5 in the above expressions, we obtain

$$\begin{aligned} &k_i < 0, \quad k_i - 0.2699k_d < -4.6836, \quad k_i - 0.3666k_d < -10.0797, \\ &k_i - 3.5358k_d > 3.912, \quad k_i - 13.5777k_d < 140.2055. \end{aligned} \tag{11.53}$$

The set of values of (k_i, k_d) for which (11.53) holds can be solved by linear programming and is denoted by $\mathcal{S}_1$. For $\mathcal{I}_2$, we have

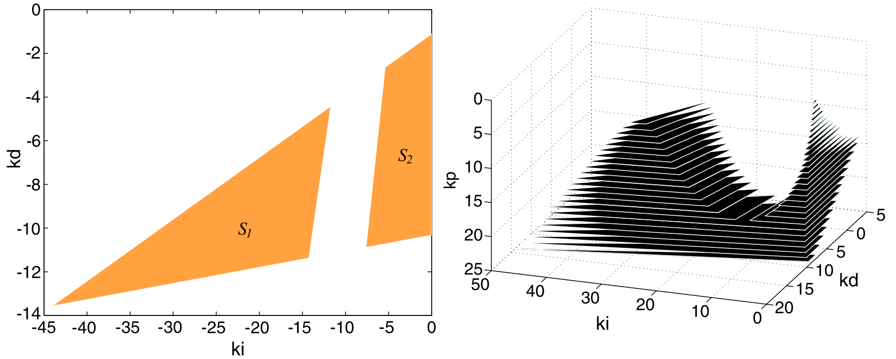
$$\begin{aligned} &k_i < 0, \quad k_i - 0.2699k_d > -4.6836, \quad k_i - 0.3666k_d > -10.0797, \\ &k_i - 3.5358k_d > 3.912, \quad k_i - 13.5777k_d < 140.2055. \end{aligned} \tag{11.54}$$

The set of values of (k_i, k_d) for which (11.54) holds can also be solved by linear programming and is denoted by $\mathcal{S}_2$. Similarly, we obtain

$$\mathcal{S}_3 = \emptyset, \qquad \mathcal{S}_4 = \emptyset, \qquad \mathcal{S}_5 = \emptyset.$$

Then, the stabilizing set of (k_i, k_d) values when $k_p = -18$ is given by

$$\mathcal{S}_{(-18)} = \bigcup_{x=1,2,\ldots,5} \mathcal{S}_x = \mathcal{S}_1 \cup \mathcal{S}_2.$$



(a) The stabilizing set of (k_i, k_d) values when $k_p = -18$.

(b) The stabilizing set of (k_p, k_i, k_d) values.

Fig. 11.2 Example 11.1

The set $\mathcal{S}_{(-18)}$ and the corresponding $\mathcal{S}_1$ and $\mathcal{S}_2$ are shown in Fig. 11.2(a). By sweeping over different k_p values within the interval $(-24.7513, 1)$ and repeating the above procedure at each stage, we can generate the set of stabilizing (k_p, k_i, k_d) values. This set is shown in Fig. 11.2(b).

11.4 Direct PID Synthesis from Frequency Response Data

11.4.1 Phase, Signature, Poles, Zeros, and Bode Plots

Let P denote a LTI plant and $P(s)$ its rational transfer function with z^+, p^+ (z^-, p^-) denoting the numbers of RHP (LHP) zeros and poles, $n(m)$ the denominator (numerator) degrees (see Fig. 11.3). Let the relative degree be denoted r_P:

$$r_P := n - m.$$

As defined earlier the signature of P is

$$\sigma(P) = \left(z^- - z^+\right) - \left(p^- - p^+\right). \tag{11.55}$$

Lemma 11.4

$$r_P = -\frac{1}{20} \cdot \frac{dP_{\mathrm{db}}(\omega)}{d(\log_{10}\omega)}\bigg|_{\omega\to\infty}, \tag{11.56}$$

$$\sigma(P) = \frac{2}{\pi}\Delta_0^{\infty}\angle P(j\omega) \tag{11.57}$$

where

$$P_{\mathrm{db}}(\omega) := 20\log_{10}\left|P(j\omega)\right|.$$

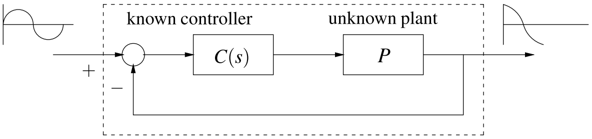


Fig. 11.3 Frequency response measurement on an unstable plant

Proof Equation (11.56) states that the relative degree is the high frequency slope of the Bode magnitude plot, and (11.57) states that the signature can be found from the net change in phase from the phase plot. □

Assuming that $P(s)$ has no $j\omega$ axis poles and zeros, we can also write

$$\sigma(P) = -(n-m) - 2\big(z^+ - p^+\big) = -r_P - 2\big(z^+ - p^+\big). \tag{11.58}$$

Therefore, $z^+ - p^+$ can be determined from the Bode plot of P. In particular if $P(s)$ is stable the Bode plot can often be obtained experimentally by measuring the frequency response of the system. Then the above relations with $p^+ = 0$ determine z^+ from the Bode plot data.

Now suppose that P is an *unstable* LTI plant with a rational transfer function *unknown* to us and assume that P does not have imaginary axis poles and zeros. We assume, however, that a known feedback controller $C(s)$ stabilizes P and the closed-loop frequency response can be *measured* and is denoted by $G(j\omega)$ for $\omega \in [0, \infty)$.

Then

$$P(j\omega) = \frac{G(j\omega)}{C(j\omega)(1 - G(j\omega))} \tag{11.59}$$

is the *computed* "frequency response" of the unstable plant. The next result shows that knowledge of $C(s)$ and $G(j\omega)$ is sufficient to determine the numbers z^+ and p^+, that is the numbers of RHP zeros and poles of the plant. Let z_c^+ denote the number of RHP zeros of $C(s)$.

Theorem 11.4

$$z^+ = \frac{1}{2}\big[-r_P - r_C - 2z_c^+ - \sigma(G)\big], \tag{11.60}$$

$$p^+ = \frac{1}{2}\big[\sigma(P) - \sigma(G) - r_C\big] - 2z_c^+. \tag{11.61}$$

Proof We have

$$G(s) = \frac{P(s)C(s)}{1 + P(s)C(s)}$$

and, since $G(s)$ is stable,

$$\sigma(G) = \big(z^- + z_c^-\big) - \big(z^+ + z_c^+\big) - (n + n_c) = -r_P - r_C - 2z_c^+ - 2z^+$$

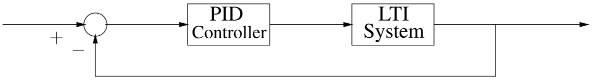


Fig. 11.4 A unity feedback system with a PID controller

which implies (11.60). From (11.55) applied to $P(s)$, we have

$$p^{+} = z^{+} + \frac{\sigma(P)}{2} + \frac{r_P}{2}. \tag{11.62}$$

Substituting (11.60) in (11.62), we have (11.61). □

Remark 11.2 In the above theorem, we assume that $C(s)$, a stabilizing controller, is known, and the corresponding closed-loop frequency response $G(j\omega)$ for $\omega \in [0, \infty)$ can be *measured*. Thus, $P(j\omega)$ can be computed from (11.59). Now r_P and $\sigma(G)$ can be found by applying the results of Lemma 11.4 to $P(j\omega)$ and $G(j\omega)$, respectively. r_C and z_c^+ are known as $C(s)$ is known. Therefore, z^+ and p^+ can be found.

Remark 11.3 In the above analysis, we have assumed for simplicity that the plant is devoid of imaginary axis poles and zeros. When such poles and zeros are present, their numbers may be known from physical considerations or their numbers and locations may be ascertained from the experimentally determined or computed Bode plot. Once identified we can lump these poles and zeros with the controller and proceed with the design procedure. At an imaginary axis zero (pole) of multiplicity k, away from the origin, the magnitude plot is zero (infinity) and the phase plot undergoes an instantaneous change of phase $k\pi$. If such poles or zeros occur at the origin, there is a corresponding phase shift of $\frac{k\pi}{2}$ at zero frequency. It is straightforward to establish that, in this case, the relations (11.60) and (11.61) need to be modified to the following:

$$z^{+} = \frac{1}{2}\big[-r_P - r_C - 2z_c^{+} - z_c^{i} - \sigma(G)\big], \tag{11.63}$$

$$p^{+} = \frac{1}{2}\big[\sigma(P) - \sigma(G) - r_C\big] - 2z_c^{+} - z_c^{i} + z^{i} - p^{i} \tag{11.64}$$

where z^i, z_c^i, p^i, p_c^i denote the numbers of imaginary axis zeros and poles of plant and controller.

11.4.2 PID Synthesis for Delay Free Continuous-Time Systems

In this section, we consider the synthesis and design of PID controllers for a continuous-time LTI plant, with underlying transfer function $P(s)$ with $n(m)$ poles (zeros) (see Fig. 11.4).

We assume that the *only* information available to the designer is:

1. Knowledge of the frequency response magnitude and phase, equivalently, $P(j\omega)$, $\omega \in [0, \infty)$ if the plant is stable.
2. Knowledge of a known stabilizing controller and the corresponding closed loop frequency response $G(j\omega)$.

Such assumptions are reasonable for most systems. We also make the technical assumption that the plant has no $j\omega$ poles or zeros so that the magnitude, its inverse, and phase are well-defined for all $\omega \geq 0$. As we have seen from the discussion in the last section, the numbers and locations of RHP poles and zeros can be found from the above data for any LTI plant and either "divided out" or lumped with the controller. Write

$$P(j\omega) = \left|P(j\omega)\right| e^{j\phi(\omega)} = P_r(\omega) + jP_i(\omega) \tag{11.65}$$

where $|P(j\omega)|$ denotes the *magnitude* and $\phi(\omega)$ the *phase* of the plant, at the frequency ω.

Let the PID controller be of the form

$$C(s) = \frac{K_i + K_p s + K_d s^2}{s(1+sT)} \quad \text{for } T > 0 \tag{11.66}$$

where T is assumed to be fixed and small. We now present results for developing our procedure for determining the stabilizing set.

Lemma 11.5 *Let*

$$F(s) := s(1+sT) + \left(K_i + K_p s + K_d s^2\right)P(s)$$

and

$$\bar{F}(s) = F(s)P(-s).$$

Then closed-loop stability is equivalent to

$$\sigma\left(\bar{F}(s)\right) = n - m + 2z^+ + 2. \tag{11.67}$$

Proof Closed-loop stability is equivalent to the condition that all zeros of $F(s)$ lie in the LHP. This, in turn is, equivalent to the condition

$$\sigma\left(F(s)\right) = n + 2 - \left(p^- - p^+\right).$$

Now consider the rational function

$$\bar{F}(s) = F(s)P(-s).$$

Note that

$$\sigma\left(\bar{F}(s)\right) = \sigma\left(F(s)\right) + \sigma\left(P(-s)\right).$$

Therefore, the stability condition becomes

$$\begin{aligned}\sigma\big(\bar{F}(s)\big) &= n+2-\big(p^- - p^+\big)+\big(z^+ - z^-\big)-\big(p^+ - p^-\big)\\ &= n+2+z^+ - z^- = n-m+2z^+ +2.\end{aligned}$$

□

Write

$$\begin{aligned}\bar{F}(j\omega) &= j\omega(1+j\omega T)P(-j\omega)+\big(K_i+j\omega K_p-\omega^2K_d\big)P(j\omega)P(-j\omega)\\ &= \underbrace{\big(K_i-K_d\omega^2\big)\big|P(j\omega)\big|^2-\omega^2TP_r(\omega)+\omega P_i(\omega)}_{\bar{F}_r(\omega,K_i,K_d)}\\ &\quad + j\omega\underbrace{\big(K_p\big|P(j\omega)\big|^2+P_r(\omega)+\omega TP_i(\omega)\big)}_{\bar{F}_i(\omega,K_p)}\\ &= \bar{F}_r(\omega,K_i,K_d)+j\omega\bar{F}_i(\omega,K_p).\end{aligned}$$

Theorem 11.5 *Let $\omega_1 < \omega_2 < \cdots < \omega_{l-1}$ denote the distinct frequencies of odd multiplicities which are solutions of*

$$\bar{F}_i\big(\omega, K_p^*\big) = 0, \tag{11.68}$$

or

$$K_p^* = -\frac{P_r(\omega)+\omega TP_i(\omega)}{|P(j\omega)|^2} = -\frac{\cos\phi(\omega)+\omega T\sin\phi(\omega)}{|P(j\omega)|} =: g(\omega) \tag{11.69}$$

for fixed $K_p = K_p^*$. Let $\omega_0 = 0$ and $\omega_l = \infty$, and $j := \operatorname{sgn}[\bar{F}_i(\infty^-, K_p^*)]$. Determine strings of integers

$$I = [i_0, i_1, i_2, \dots, i_l]$$

with $i_t \in \{+1, -1\}$ such that

- *For $n-m$ even,*

$$\big[i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1}2i_{l-1} + (-1)^l i_l\big](-1)^{l-1}j = n-m+2z^+ +2. \tag{11.70}$$

- *For $n-m$ odd,*

$$\big[i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1}2i_{l-1}\big](-1)^{l-1}j = n-m+2z^+ +2. \tag{11.71}$$

Then for the fixed $K_p = K_p^*$, the (K_i, K_d) values corresponding to closed-loop stability are given by

$$\bar{F}_r(\omega_t, K_i, K_d)i_t > 0, \tag{11.72}$$

where the i_t's are taken from strings satisfying (11.70) *and* (11.71), *and ω_t's are taken from the solutions of* (11.68).

Proof By Lemma 11.5, the stability condition has been reduced to the signature condition in (11.67). The theorem follows from applying Lemma 11.2 to compute the signature of $\bar{F}(s)$. □

The following result clarifies what range the parameters K_p must be swept over.

Theorem 11.6 *For the given function $g(\omega)$ in* (11.69) *determined completely by the plant data $P(j\omega)$ and T:*

(i) *A necessary condition for PID stabilization is that there exists K_p such that the function*

$$K_p = g(\omega) \tag{11.73}$$

has at least k distinct roots of odd multiplicities where

$$k \geq \frac{n-m+2z^+ +2}{2} - 1 \quad \textit{if } n-m \textit{ even},$$

$$k \geq \frac{n-m+2z^+ +3}{2} - 1 \quad \textit{if } n-m \textit{ odd}.$$

(ii) *There exist unique ranges of ω over which condition* (i1) *occurs and this determines the corresponding ranges of K_p to be swept.*

Remark 11.4 The function $g(\omega)$ is well defined due to the assumption that the plant either has no $j\omega$ zeros or these have been lumped with the controller.

The computation of the stabilizing set implied by the above results is summarized as the following procedure.

Computation of PID Stabilizing Set from Nyquist/Bode Data The complete set of stabilizing PID gains can be computed by the following procedure:

For stable systems: The available data is the frequency response of the plant $P(j\omega)$.

0.1 Determine the relative degree of the plant $r_P = n - m$ from the high frequency slope of the Bode magnitude plot of $P(j\omega)$.

0.2 Let $\Delta_0^\infty[\phi(\omega)]$ denote the net change of phase of $P(j\omega)$ for $\omega \in [0,\infty)$. Determine z^+ from

$$\Delta_0^\infty\big[\phi(\omega)\big] = -\big[(n-m)+2z^+\big]\frac{\pi}{2}$$

which follows from (11.58) with $p^+ = 0$.

For unstable systems: The available data are a stabilizing controller transfer function $C(s)$ and the frequency response of the corresponding stable closed-loop system $G(j\omega)$.

0.1 Compute the frequency response $P(j\omega)$ from (11.59).
0.2 Determine relative degree of the plant r_P from the high frequency slope of the Bode magnitude plot of $P(j\omega)$.
0.3 Determine z_c^+ and r_C from $C(s)$.
0.4 Compute $\sigma(G)$ from (11.57) applied to $G(j\omega)$.
0.5 Compute z^+ using (11.60) in Theorem 11.4.
0.6 Compute $g(\omega)$ using (11.69) and the data.

1. Fix $K_p = K_p^*$, solve (11.69), and let $\omega_1 < \omega_2 < \cdots < \omega_{l-1}$ denote the distinct frequencies of odd multiplicities which are solutions of (11.69).
2. Set $\omega_0 = 0$, $\omega_l = \infty$ and $j = \operatorname{sgn}\bar{F}_i(-\infty^-, K_p^*)$. Determine all strings of integers $i_t \in \{+1, -1\}$ such that:

   – For $n - m$ even,

$$\left[i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{i-1} 2i_{l-1} + (-1)^l i_l\right](-1)^{l-1} j = n - m + 2z^+ + 2. \tag{11.74}$$

   – For $n - m$ odd,

$$\left[i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{i-1} 2i_{l-1}\right](-1)^{l-1} j = n - m + 2z^+ + 2. \tag{11.75}$$

3. For the fixed $K_p = K_p^*$ chosen in Step 1, solve for the stabilizing (K_i, K_d) values from

$$\left[K_i - K_d\omega_t^2 + \frac{\omega_t \sin\phi(\omega_t) - \omega_t^2 T \cos\phi(\omega_t)}{|P(j\omega_t)|^2}\right] i_t > 0, \tag{11.76}$$

   for $t = 0, 1, \ldots, l$.
4. Repeat the previous three steps by updating K_p over prescribed ranges. The ranges over which K_p must be swept is determined from the requirements that (11.74) or (11.75) is satisfied for at least one string of integers as in Theorem 11.6.

We emphasize that all computations are based on the data $P(j\omega)$ and knowledge of the transfer function $P(s)$ is not required. For well-posedness of the loop, it is necessary that

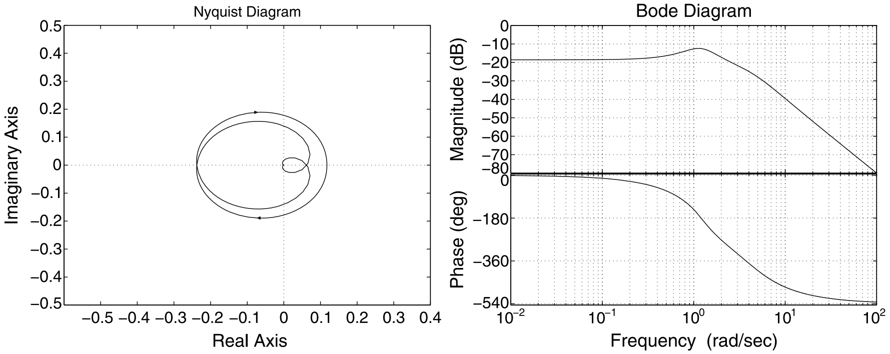
$$K_d \neq -\frac{T}{P(\infty)}.$$

For strictly proper plants, $P(\infty) = 0$ and this constraint vanishes.

11.4.3 Illustrative Example

To illustrate the main result, we take a set of frequency data points for the *stable* plant used in [5]. Let the data points be

$$\mathbf{P}(j\omega) := \left\{P(j\omega), \omega \in (0, 10) \text{ sampled every } 0.01\right\}.$$



(a) Nyquist plot of the plant (b) Bode plot of the plant

Fig. 11.5 Frequency response data

The Nyquist and Bode plot are shown in Fig. 11.5(b).

The high frequency slope of the Bode magnitude plot is −40 db/decade and thus $n-m=2$. The total change of phase is −540 degrees and so

$$-6\frac{\pi}{2} = -\big((n-m) - 2\big(p^+ - z^+\big)\big)\frac{\pi}{2},$$

and since the plant is stable, $p^+ = 0$, giving $z^+ = 2$. The required signature for stability can now be determined and is

$$\sigma\big(\bar{F}(s)\big) = (n-m) + 2z^+ + 2 = (2) + 2(2) + 2 = 8.$$

Since $n-m$ is even, we have

$$\big[i_0 - 2i_1 + 2i_2 - 2i_3 + 2i_4 - \cdots + (-1)^l i_l\big](-1)^{l-1} j = 8$$

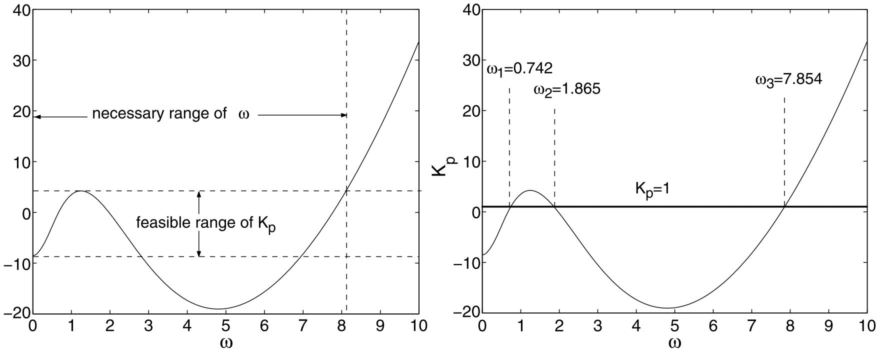
where

$$j = \operatorname{sgn}\big[\bar{F}_i\big(\infty^-, K_p\big)\big] = -\operatorname{sgn}\Big[\lim_{\omega\to\infty} g(\omega)\Big] = -1,$$

and it is clear that at least four terms are required to satisfy the above. In other words, $l \geq 4$. From Fig. 11.6(a), it is easy to see that (11.69) has at most three positive frequencies as solutions, and therefore we have $i_0 - 2i_1 + 2i_2 - 2i_3 + i_4 = 8$. Also $i_4 = \operatorname{sgn}[\bar{F}_r(\infty^-, K_i, K_d)] = 1$ independent of K_i and K_d. This means that K_p must be chosen so that $\bar{F}_i(\omega, K_p^*) = 0$ has exactly three positive real zeros. This gives the feasible range of K_p values as shown in Fig. 11.6(a) which depicts the function

$$g(\omega) = -\frac{\cos\phi(\omega) + \omega T \sin\phi(\omega)}{|P(j\omega)|}. \tag{11.77}$$

The feasible range of K_p is such that K_p intersects the graph of $g(\omega)$ three times. This feasible range is shown in Fig. 11.6(a).



(a) Graph of the function $g(\omega)$ in (77).

(b) Finding the set of ω's satisfying (69) with $K_p = 1$.

Fig. 11.6 Selecting K_p

In Fig. 11.6(a), we also observe that the *frequency range over which plant data must accurately be known for PID control* is [0, 8.2] radians/s. We now fix $K_p^* = 1$ and compute the set of ω's that satisfies

$$-\frac{\cos\phi(\omega) + \omega T \sin\phi(\omega)}{|P(j\omega)|} = 1.$$

To find the set of ω's satisfying the above, we plot the function $g(\omega)$ as shown in Fig. 11.6(b). The three frequencies $\omega_1, \omega_2, \omega_3$ are required to compute the stability set in (K_i, K_d) space.

From this we found the solutions $\{\omega_1, \omega_2, \omega_3\} = \{0,\ 0.742,\ 1.865,\ 7.854\}$. This leads to the requirement $i_0 - 2i_1 + 2i_2 - 2i_3 = 7$, giving the only feasible string

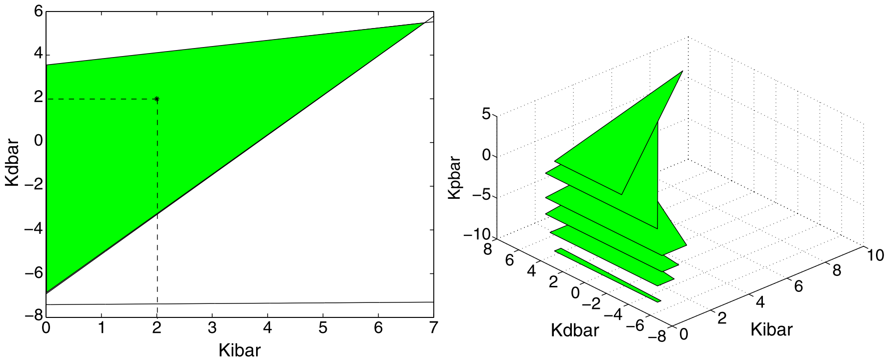
$$\mathcal{F} = \{i_0, i_1, i_2, i_3\} = \{1, -1, 1, -1\}.$$

Thus, we have the following set of linear inequalities for stability:

$$\begin{aligned} K_i > 0, &\quad -3.8114 + K_i - 0.5506K_d < 0, \\ 12.2106 + K_i - 3.4782K_d > 0, &\quad -457.0235 + K_i - 61.6853K_d < 0. \end{aligned}$$

The complete set of stabilizing PID gains for $K_p^* = 1$ is given in Fig. 11.7(a). This set is determined by finding the string of integers $\{i_0, i_1, i_2, i_3\}$ satisfying the stability condition (11.74) and (11.75). The corresponding linear inequalities (11.76) determine the stabilizing set shown in the (K_i, K_d) space. The point marked with "*" was used as a test point to verify stability. The region obtained here was verified to be identical to that obtained in [5].

By sweeping over K_p, we have the entire set of stabilizing PID gains as shown in Fig. 11.7(b). This set is determined by sweeping over the feasible range of K_p determined in Fig. 11.6(b), and solving the corresponding linear inequalities given by (11.76) in (K_i, K_d) space for a fixed K_p^* in this range.

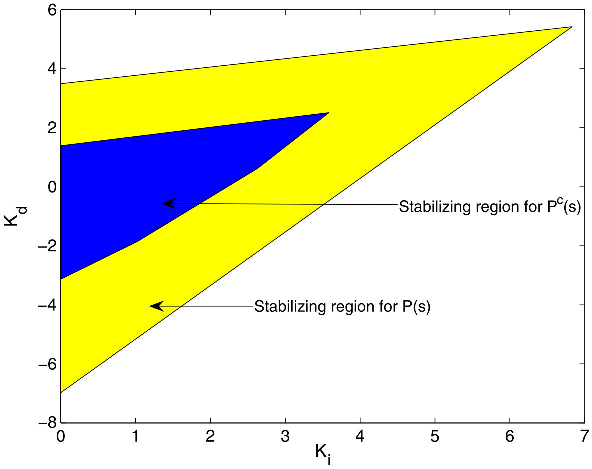


(a) The complete set of stabilizing PID gains in (K_i, K_d) space when $K_p = 1$.

(b) Entire set of stabilizing PID gains.

Fig. 11.7 Stabilizing set

Fig. 11.8 The complete set of stabilizing PID gains satisfying H_∞ norm specification when $K_p = 1$



Note that the range of K_p over which the search needs to be carried out is also obvious from Fig. 11.6(a) as discussed and it is $K_p \in [-8.5, 4.2]$.

We now consider the problem of achieving an H_∞ norm specification on the complementary sensitivity function $T(s)$, that is,

$$\left\| W(s)T(s) \right\|_\infty < 1 \quad \text{where } W(s) = \frac{s+0.1}{s+1}.$$

This set is obtained by solving the PID stabilization problem for the family of complex plants for fixed $K_p = 1$, corresponding to the H_∞ norm specification, and intersecting the resulting stabilizing set with that of the stabilizing set for the real plant (see Fig. 11.8).

By sweeping K_p over the feasible ranges and generating the linear inequalities (11.86) in (K_i, K_d) space corresponding to stability and performance, we have the

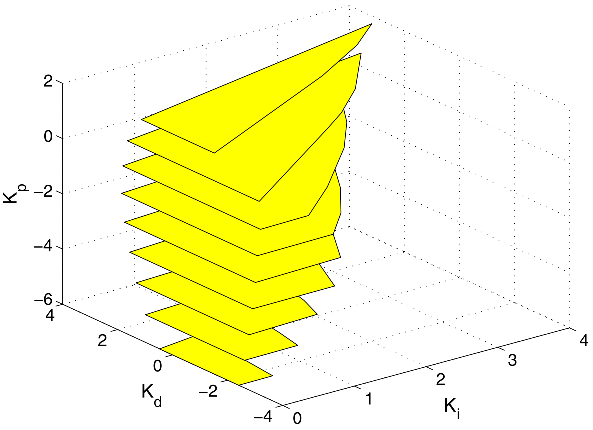


Fig. 11.9 The entire PID parameter set satisfying the prescribed H_∞ norm specification

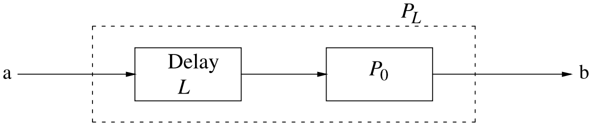


Fig. 11.10 A plant with cascaded delay

entire stabilizing PID gains that satisfy the given H_∞ specification as shown in Fig. 11.9.

11.4.4 PID Synthesis for Systems with Time-Delay

In this section, we show how the previous results can be extended to systems with delay. Consider the finite dimensional LTI plant P_L with a cascaded time-delay in Fig. 11.10.

Here P_0 represents an LTI delay free system with a proper transfer function. The transfer functions of P_0 and P_L are denoted $P_0(s)$ and $P_L(s)$, respectively. We assume that frequency response measurements can be made at terminals "a" and "b", that is, on the delay system P_L. Thus, the data we have is:

$$P_L(j\omega) = e^{-j\omega L} P_0(j\omega) = m_L(\omega) e^{j\phi_L(\omega)}, \quad 0 \le \omega < \infty.$$

Write

$$P_0(j\omega) = m_0(\omega) e^{j\phi_0(\omega)}.$$

Therefore,

$$m_0(\omega) = m_L(\omega) \quad \text{and} \quad \phi_L(\omega) = \phi_0(\omega) - \omega L, \quad 0 \le \omega < \infty. \tag{11.78}$$

It is clear that $m_0(\omega)$ and $\phi_0(\omega)$ can be determined from the data $m_L(\omega)$ and $\phi_L(\omega)$ on the system with embedded delay L when L is known.

Now let $C(s, \mathbf{k})$ denote the PID controller

$$C(s,\mathbf{k}) = \frac{K_i + K_p s + K_d s^2}{s(1+sT)}, \quad \mathbf{k} = [K_i, K_p, K_d].$$

Let $\mathcal{S}_0$ denote the set of stabilizing PID controllers for the delay free plant

$$\mathcal{S}_0 = \{\mathbf{k} : C(s,\mathbf{k}) \text{ stabilizes } P_0\}.$$

We have seen how $\mathcal{S}_0$ can be found in the previous section when $P_0(j\omega)$ is known. It follows from (11.78) that $\mathcal{S}_0$ can be determined from $P_L(j\omega)$ the data for the embedded delay system. We denote by $\mathcal{S}_L$ the set of PID controllers stabilizing the plant P_0 with cascaded delay ranging from 0 to L seconds.

Following the results in [8], introduce the sets

$$\mathcal{S}_\infty = \{\mathbf{k} : |C(s,\mathbf{k})P_0(s)|_{s=\infty} \geq 1\},$$
$$\mathcal{S}_\mathcal{B} = \left\{\mathbf{k} : C(j\omega,\mathbf{k}) = \frac{-e^{j\omega\underline{L}}}{P_0(j\omega)}, \omega \in [0,\infty), 0 \leq \underline{L} \leq L\right\}.$$

The main result of [8] is the following theorem.

Theorem 11.7 ([8]) *The set $\mathcal{S}_L$ can be found from*

$$\mathcal{S}_L = \mathcal{S}_0 \backslash (\mathcal{S}_\infty \cup \mathcal{S}_\mathcal{B}). \tag{11.79}$$

The proof of this theorem is omitted and the reader is referred to Chap. 11 in [9] for a complete proof. The set $\mathcal{S}_\infty$ consists of those PID gains for which the Nyquist plot approaches points outside the unit circle as $s \to \infty$. The set $\mathcal{S}_\mathcal{B}$ consists of those PID gains for which an imaginary axis characteristic root occurs for delays less than L. The relationship (11.79) states that the exclusion of $\mathcal{S}_\infty$ and $\mathcal{S}_\mathcal{B}$ from $\mathcal{S}_0$ determines the stabilizing set $\mathcal{S}_L$ for the system with cascaded delay up to L seconds.

The computation of $\mathcal{S}_0$ has been described in the previous section. The set $\mathcal{S}_\infty$ is easy to calculate. In fact,

$$\mathcal{S}_\infty = \left\{\mathbf{k} : |K_d| \geq \frac{T}{|P_0(\infty)|}\right\}.$$

To determine $\mathcal{S}_\mathcal{B}$, let

$$\theta(\omega,\mathbf{k}) := \angle\left(\frac{K_i - K_d\omega^2 + j\omega K_p}{j\omega(1+j\omega T)}\right).$$

Then the conditions defining $\mathcal{S}_\mathcal{B}$ can be written as the magnitude and phase conditions

$$K_i - K_d\omega^2 = \pm\sqrt{\frac{\omega^2(1+T^2)}{m_0^2(\omega)} - \omega^2 K_p^2}, \tag{11.80}$$

$$\theta(\omega, \mathbf{k}) \geq \pi - \phi_0(\omega) - \omega L, \quad \omega \in [0, \infty), \tag{11.81}$$

so that

$$\mathcal{S}_{\mathcal{B}} = \big\{ \mathbf{k} : \mathbf{k} \text{ satisfies (11.80) and (11.81) for some } \omega \in [0, \infty) \big\}.$$

Note that (11.80) represents a straight line in (K_i, K_d) space for each fixed K_p and ω. The calculation of the set $\mathcal{S}_{\mathcal{B}}$ is tedious but straight-forward if one sweeps over the frequency variable.

11.5 PID Synthesis for Performance

Many performance attainment problems can be cast as simultaneous stabilization of the plant $P(s)$, and families of real and complex plants. For example,

1. The problem of achieving a gain margin is equivalent to simultaneously stabilizing the plant $P(s)$ and the family of *real* plants
$$\mathcal{P}^c(s) = \big\{ K P(s) : K \in [K_{\min}, K_{\max}] \big\}.$$
2. The problem of achieving prescribed phase margin θ_m is equivalent to simultaneously stabilizing the plant $P(s)$ and the family of *complex* plants
$$\mathcal{P}^c(s) = \big\{ e^{-j\theta} P(s) : \theta \in [0, \theta_m] \big\}.$$
3. The problem of achieving an H_∞ norm specification on the sensitivity function $S(s)$, that is, $\|W(s)S(s)\|_\infty < \gamma$ is equivalent to simultaneously stabilizing the plant $P(s)$ and the family of *complex* plants
$$\mathcal{P}^c(s) = \left\{ \left[\frac{1}{1 + \frac{1}{\gamma} e^{j\theta} W(s)} \right] P(s) : \theta \in [0, 2\pi] \right\}.$$
4. The problem of achieving an H_∞ norm specification on the complementary sensitivity function $T(s)$, that is, $\|W(s)T(s)\|_\infty < \gamma$, is equivalent to simultaneously stabilizing the plant $P(s)$ and the family of *complex* plants
$$\mathcal{P}^c(s) = \left\{ P(s) \left[1 + \frac{1}{\gamma} e^{j\theta} W(s) \right] : \theta \in [0, 2\pi] \right\}.$$

Based on the above, we consider the problem of stabilizing a complex LTI plant with transfer function $P^c(s)$ using PID control. Let n_c and m_c denote the numerator and denominator degrees of $P^c(s)$ and $z_c^+ (p_c^+)$ the number of RHP zeros and poles. Note that in each of the above cases $z_c^+ (p_c^+)$ are uniquely determined by the numbers of RHP zeros and poles of the real plant $P(s)$. As in the real case, we assume that the only information available to the designer is "Knowledge of the frequency response magnitude and phase of the real plant, equivalently, $P(j\omega)$, $\omega \in (0, \infty)$."

Suppose the complex plant with transfer function $P^c(s)$ is in a closed-loop with the PID controller

$$C(s) = \frac{K_i + K_p s + K_d s^2}{s(1+sT)}, \quad T > 0.$$

Consider the complex rational function

$$F^c(s) := s(1+sT) + \big(K_i + K_p s + K_d s^2\big)P^c(s).$$

Closed-loop stability is equivalent to the condition that the zeros of $F^c(s)$ lie in the LHP. This, in turn, is equivalent to the condition

$$\sigma\big(F^c(s)\big) = n_c + 2 - \big(p_c^- - p_c^+\big).$$

Now consider the rational function

$$\bar{F}^c(s) = F^c(s)P^{c*}(-s),$$

where $P^{c*}(-s)$ is obtained by replacing the numerator and denominator coefficients of $P^c(s)$ by their conjugates and replacing "s" by "$-s$". Note that

$$\sigma\big(\bar{F}^c(s)\big) = \sigma\big(F^c(s)\big) + \sigma\big(P^{c*}(-s)\big).$$

It is easy to verify that

$$\sigma\big(P^{c*}(-s)\big) = \sigma\big(P^c(-s)\big).$$

Therefore, the stability condition becomes

$$\sigma\big(\bar{F}^c(s)\big) = n_c - m_c + 2z_c^+ + 2.$$

To compute $\sigma(\bar{F}^c(s))$, we write

$$\bar{F}^c(j\omega) = j\omega(1+j\omega T)P^{c*}(-j\omega) + \big(K_i + j\omega K_p - \omega^2 K_d\big)P^c(j\omega)P^{c*}(-j\omega).$$

Write

$$P^c(j\omega) = P_r^c(\omega) + jP_i^c(\omega),$$

where $P_r^c(\omega)$ and $P_i^c(\omega)$ are real rational functions, then

$$P^{c*}(-j\omega) = P_r^c(\omega) - jP_i^c(\omega).$$

Then

$$\begin{aligned}\bar{F}^c(j\omega) &= \underbrace{\big(K_i - K_d\omega^2\big)P^c(j\omega)P^{c*}(-j\omega) - \omega^2 T P_r^c(\omega) + \omega P_i^c(\omega)}_{\bar{F}_r^c(\omega, K_i, K_d)} \\ &\quad + j\underbrace{\big(K_p P^c(j\omega)P^{c*}(-j\omega) + P_r^c(\omega) + \omega T P_i^c(\omega)\big)}_{\bar{F}_i^c(\omega, K_p)} \\ &= \bar{F}_r^c(\omega, K_i, K_d) + j\bar{F}_i^c(\omega, K_p).\end{aligned}$$

Theorem 11.8 *Let $\omega_1 < \cdots < \omega_{l-1}$ denote the distinct frequencies of odd multiplicities which are solutions of*

$$\bar{F}_i^c\big(\omega, K_p^*\big) = 0, \tag{11.82}$$

for $K_p = K_p^*$. Let $\omega_0 = -\infty$ and $\omega_l = \infty$ and $j = \operatorname{sgn}[\bar{F}_i^c(\infty^-, K_p^*)]$. Define strings of integers $\{i_1, i_2, \dots, i_{l-1}\}$ with $i_t \in \{+1, -1\}$ such that

$$\sum_{r=1}^{l-1}(-1)^{l-1-r} i_r \cdot j = n_c - m_c + 2z_c^+ + 2. \tag{11.83}$$

For the fixed $K_p = K_p^*$, the (K_i, K_d) stabilizing values for the complex plant P^c are those satisfying

$$\bar{F}_r^c(\omega_t, K_i, K_d) i_t > 0, \tag{11.84}$$

where i_t's are taken from the strings satisfying (11.83) and ω_t's are the roots of (11.82).

The complete set of stabilizing PID gains for a given complex LTI plant can be found from the frequency response data $P^c(j\omega)$ and the knowledge of the number p_c^+ of RHP poles of $P^c(s)$. These, in turn, can be found from knowledge of the real plant data $P(j\omega)$. This leads to the procedure parallel to the real stabilization case. Note that (11.82) can be written as

$$K_p^* = -\frac{P_r^c(\omega) + \omega T P_i^c(\omega)}{P^c(j\omega)P^{c*}(-j\omega)} = -\frac{P_r^c(\omega) + \omega T P_i^c(\omega)}{|P^c(j\omega)|^2} =: g^c(\omega). \tag{11.85}$$

For the fixed $K_p = K_p^*$, the stabilizing (K_i, K_d) are determined by the following expression:

$$\left[K_i - K_d\omega_t^2 + \frac{\omega_t \cos\phi_c(\omega_t) - \omega_t^2 T \sin\phi_c(\omega_t)}{|P^c(j\omega_t)|^2}\right] i_t > 0, \tag{11.86}$$

for $t = 0, 1, \dots$.

Note that for the class of performance problems mentioned, $P^c(s)$ is of the form

$$P^c(s) = W^c(s)P(s),$$

where $W^c(s)$ is a weight chosen by the designer. Therefore,

$$P^{c*}(-s) = W^{c*}(-s)P(-s),$$

and thus $P^{c*}(-j\omega)$ is known from the knowledge of $P(j\omega)$. It is also important to note that all calculations above use only the data $P(j\omega)$ of the real plant. Finally, we need to intersect the stabilizing set with the performance set. For a fixed K_p, this amounts to generating linear inequalities corresponding to stability from the previous section and corresponding to performance described above, and solving them simultaneously. In general, these sets are disconnected and a union of convex components. Multiple performance specifications can be handled in a similar manner.

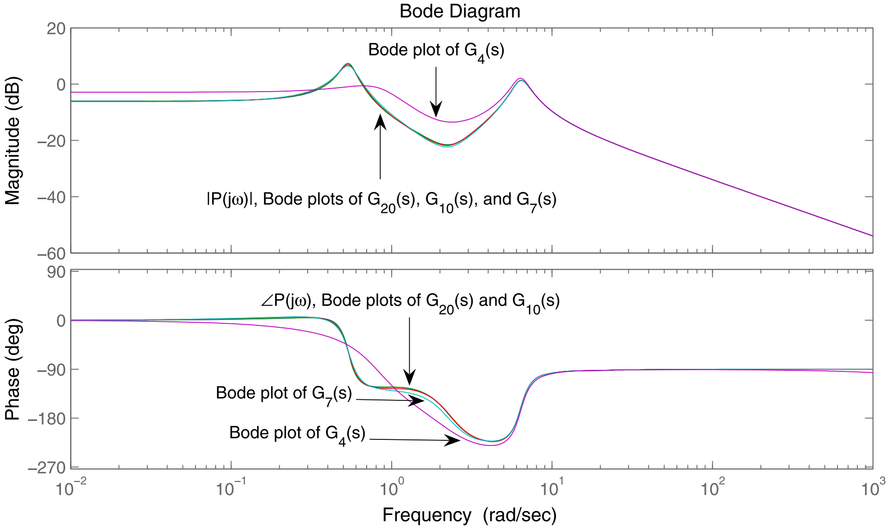


Fig. 11.11 Frequency domain data and the Bode plots of 20th, 10th, 7th, and 4th order identified models

11.6 Data Based Design vs. Model Based Design

In this section, we discuss some differences between model based design and the data based designs described here. In model based design, mathematical models are obtained from the laws of physics that describes the dynamic behavior of a system to be controlled. On the other hand, the most common way of obtaining mathematical models in engineering systems is through a system identification process. Let us assume that the frequency domain data is obtained by exciting a plant with a unknown rational transfer function, by sinusoidal signals. In theory, a system identification procedure should exactly be able to determine the unknown rational transfer function. In this ideal situation, there should be no distinction between model based and data based synthesis methods. However, typical system identification procedures can fail to find an exact rational function even if exact (or perfect) data is available. This is especially true when the order of the plant is high. The following example illustrates that this, in turn, can lead to drastic differences in control design.

Example 11.2 Let us assume that the frequency domain data $P(j\omega)$ shown below is obtained from a 20th order plant. Note that the plant is unstable with 2 RHP poles. Then mathematical models of 20th, 10th, 7th, and 4th orders were obtained by a system identification process applied to $P(j\omega)$. For convenience, let us denote by $G_{20}(s)$, $G_{10}(s)$, $G_7(s)$, and $G_4(s)$ the 20th, 10th, 7th, and 4th order models identified, respectively. Figure 11.11 shows that the Bode plots of the four identified models along with the frequency domain data collected from the 20th order plant considered here. It is observed that the Bode plots of these are almost identical except for the fact that the 4th order identified model is relatively crude.

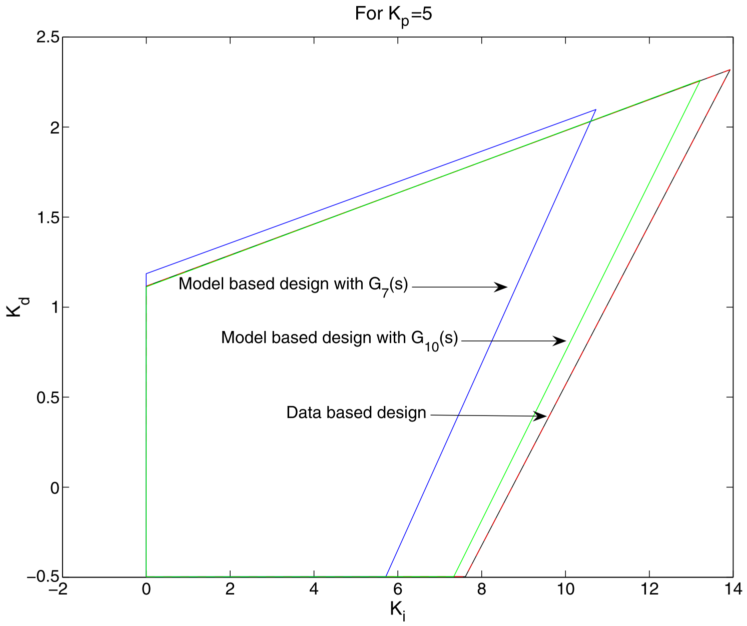


Fig. 11.12 Stabilizing regions

We now compute the stabilizing PID parameter regions of each of these systems. Figure 11.12 shows the stabilizing regions in the PID controller parameter space for each of these systems for $K_p = 5$. Note that the PID stabilizing parameter regions for G_{20} and $G_4(s)$ are empty and not shown in the figure.

We can make the following observations.

1. The model $G_4(s)$ is found to be not PID stabilizable for the chosen $K_p = 5$. This is not unexpected for the model $G_4(s)$ since there is some difference between the Bode plot of identified model $G_4(s)$ and the data $P(j\omega)$.
2. We have verified that $G_{20}(s)$ is also not PID-stabilizable. This may seem surprising since the Bode magnitude and phase plot of $G_{20}(s)$ is indistinguishably close to the data $P(j\omega)$. In fact, $G_{20}(s)$ has additional RHP poles and zeros over those in the original plant model. For this reason, the model is not PID-stabilizable.
3. The stabilizing regions are found for $G_{10}(s)$ and $G_7(s)$. These regions overlap, but are not the same. It suggests that selection of controllers should be done inside the intersection of the stabilizing regions for $G_{10}(s)$ and $G_7(s)$.
4. The stabilizing region obtained from the data based method given here differs from those for $G_{10}(s)$ and $G_7(s)$. Thus, a reasonable selection of controller may be done inside the intersection of the stabilizing regions for $G_{10}(s)$, $G_7(s)$, and the region obtained from the data based method.

In practice, experimentally obtained frequency domain data always contains noise and measurement errors. As discussed above, the stability regions determined by the model based design and the data based design will generally be different. Although the accuracy of the regions depends on each particular case, the data based design gives useful alternatives to model based design methods and in general the two complement each other. In particular it gives new guidelines for identification when it is to be used for controller design.

11.7 Conclusions and Future Perspectives

The extension of the results presented here to multivariable and nonlinear systems are challenging and open important problems. Some preliminary results are reported in [11, 12]. There is some progress on translating the results of this paper to efficient software but much work remains to be done. The application of these results to specific problems in the process control, motion control, power systems and aerospace industries offer many opportunities.

Acknowledgements This research was supported in part by NSF Grants CMMI-0927664 and CMMI-0927652, and DOD Grant W911NF-08-0514.

References

1. Bennett, S.: Nicholas Minorsky and the automatic steering of ships. IEEE Control Syst. Mag. **4**(4), 10–15 (1984)
2. Special Issue on PID. IEEE Control Syst. Mag. **26**(1), February 2006
3. Ziegler, J.G., Nichols, N.B.: Optimum setting for automatic controllers. Trans. Am. Soc. Mech. Eng. **64**, 759–768 (1942)
4. Cohen, G.H., Coon, G.A.: Theoretical consideration of retarded control. Trans. Am. Soc. Mech. Eng. **76**, 827–834 (1953)
5. Datta, A., Ho, M.-Z., Bhattacharyya, S.P.: Structure and Synthesis of PID Controllers. Springer, London (2000)
6. Bhattacharyya, S.P., Datta, A., Keel, L.H.: Linear Control Theory: Structure, Robustness, and Optimization. CRC Press, New York (2009)
7. Keel, L.H., Bhattacharyya, S.P.: Controller synthesis free of analytical models: three term controllers. Modern PID control: stabilizing sets and multiple performance specifications. IEEE Trans. Autom. Control **53**(6), 1353–1369 (2008)
8. Xu, H., Datta, A., Bhattacharyya, S.P.: PID stabilization of LTI plants with time-delay. In: Proc. 42nd IEEE Conf. Decision Control, Maui, HI, pp. 4038–4040 (2003)
9. Silva, G.J., Datta, A., Bhattacharyya, S.P.: PID Controllers for Time-delay Systems. Birkhäuser, Boston (2005)
10. Keel, L.H., Bhattacharyya, S.P.: Robust, fragile, or optimal? IEEE Trans. Autom. Control **42**(8), 1098–1105 (1997)
11. Keel, L.H., Bhattacharyya, S.P.: A Bode plot characterization of all stabilizing controllers. IEEE Trans. Autom. Control **55**(11), 2650–2654 (2010)
12. Keel, L.H., Bhattacharyya, S.P.: Robust stability via sign-definite decomposition. IEEE Trans. Autom. Control **56**(1), 140–145 (2011)

Chapter 12
Fragility Evaluation of PI and PID Controllers Tuning Rules

Víctor M. Alfaro and Ramon Vilanova

12.1 Introduction

The introduction in 1940 of the first commercially available *Proportional Integral Derivative* (PID) controller, the Taylor Fulscope 100 [18, 19], motivated Ziegler and Nichols to present in 1942 their well-known tuning rules [73]. Since that date, a great number of other tuning procedures have been developed for the PID controller and its variations, as revealed in O'Dwyer's handbook [54].

At the beginning, only the control system *performance* was taken into account in the controller design, considering a step change either in the set-point, *servo-control* operation, or in the load-disturbance, *regulatory control* operation, as in the classic tuning rules of Cohen and Coon [23], López et al. [48], and Rovira et al. [58], among others [21, 22, 38, 50, 57, 65], for One-Degree-of-Freedom (1DoF) PI and PID controllers.

Later, the consideration of the control system relative stability, its *robustness* to the changes in the controlled process characteristics, was introduced into the controller design, considering first the control-loop gain and phase margins (A_m, ϕ_m) as in [11, 27, 35, 36, 44, 47, 72]. More recently, these classic indicators of robustness have been replaced by a single value given by the maximum of the magnitude of the sensitivity function, denoted by M_S. This approach has been used in [5, 7, 12, 13, 17, 31, 46, 56, 65, 66].

The implementation of the Two-Degree-of-Freedom (2DoF) PID controllers, proposed by Araki [9, 10], allowed the separation of the control system design in

V.M. Alfaro (✉)
Departamento de Automática, Escuela de Ingeniería Eléctrica, Universidad de Costa Rica, 11501-2060 San José, Costa Rica
e-mail: Victor.Alfaro@ucr.ac.cr

R. Vilanova
Departament de Telecomunicació i d'Enginyeria de Sistemes, Escola d'Enginyeria, Universitat Autònoma de Barcelona, Bellaterra, 08193 Barcelona, Spain
e-mail: Ramon.Vilanova@uab.cat

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_12, © Springer-Verlag London Limited 2012

two steps, considering in the first step, the control system stability and the regulatory control performance and in the second, the servo-control performance. See [3, 6, 16, 17, 30, 32, 64] and the references therein.

Up to this point, it is clear that a design procedure of a control system with PID controllers must make use of the capabilities provided by the 2DoF controllers, which must take into consideration several conflicting specifications: on the one hand, *performance*, or the response to the set-point and load-disturbance changes; and on the other hand, the system relative stability, or the *robustness* to the changes in the controlled process dynamics. The control signal variation and extreme values must also be taken into account. Therefore, the controller design is really a multi-objective problem [33, 41].

There is, however, another consideration that must be taken into account in the control system design process: the effect of the variation of the controller parameters over the control system stability and performance, known as the *controller fragility*. If the control system robustness is an indication of the margin of variation of the process characteristics with a fixed controller, then the controller fragility has a similar meaning but in terms of the variation of the controller parameters considering a fixed controlled process.

The fragility of certain controllers was documented by Keel and Battacharyya [45]. They found that many modern design techniques for optimum and robust controllers under the H_2, H_∞, and l_1 norms would produce extremely fragile, high-order controllers. They observed that in some cases, minimum variations of the parameters of these controllers would make the system unstable. A fragility analysis was included in the PID controller design by Datta et al. [25], Ho [34], and Silva et al. [62].

Although, in control system designs, the assumption is often made that the controller can be implemented exactly, a certain degree of uncertainty inevitably exists in the controller implementation. The controller fragility is affected by the tolerances of its analog components. In its digital version, there are inaccuracies because of the use of fixed-length words and rounded errors of numerical calculations [40, 69]. In addition, the controller must allow variations of its parameters around their design values, making it easy to *fine-tune* the controller when the control loop is placed in service. The latter is the most probable cause of major variations in the controller parameters from their design, or nominal, values. Effectively, most of the tuning approaches, either based on tuning rules or on optimization methods, provide precise values for the controller parameters, but due to the inaccuracies associated with the controlled process model used as part of the tuning procedure, normally these parameters should be taken only as a first approximation, and such final fine-tuning of the controller is normally required.

Considering the above, modern tuning rules for PI and PID controllers must take into account issues such as, the closed-loop servo- and regulatory control *performance*, the *control effort* requirements, the control system *robustness*, and the controller *fragility*.

The chapter is organized as follows. Section 12.2 introduces the controller fragility concept and its treatment in the control systems literature. Section 12.3

presents the framework used for its analysis. Section 12.4 states the fragility indices. In Sect. 12.5, the use of fragility graphic tools is introduced, and the fragility of several tuning rules is evaluated. The chapter ends with some conclusions and suggestions for future work.

12.2 Early Work on Controller Fragility

As indicated above, the fragility of high-order controllers designed with optimal and robust techniques was pointed out by Kell and Battacharyya [45]. As a measure of the controller fragility, they utilized the ratio of the l_2-norm of the perturbation vector of the controller transfer function coefficients that made the system unstable to the l_2-norm of its nominal parameter vector. If the parameter perturbations making the system unstable are "small", then the controller is considered *fragile*. Their fragility index is really a parametric stability margin measure around the nominal parameters of the designed controller. However, and as indicated by Mäkilä [49] and Paattilammi and Mäkilä [55] (see references therein), the high-order controllers fragility is not only a result of the optimization procedure used for their synthesis but also a result of the controllers implementation.

The "controller fragility" term is related more to high-order controller design [1, 26, 43] and realization [51, 52, 69] than with PID controllers. However, there are problems with the digital implementation of PID controllers with fixed-point arithmetic due to the Finite-Word-Length (FWL) effect, addressed in [41, 70].

The stability radii as measurements of the controller fragility were introduced by Whidborne [69]. If the controller nominal (design) parameters are $\bar{\theta}_c^o$ and $\Delta\bar{\theta}_c$ are the controller parameter perturbations, the l_2-norm controller *stability radius*, r_s, is the radius of the largest ball centered at $\bar{\theta}_c^o$ inside the controller parametric stability space and is given by the following:

$$r_s \doteq \max_{\Delta}\bigl\{\bigl\|\Delta\bar{\theta}_c\bigr\|_2 : \text{closed-loop control system is stable}\bigr\}. \tag{12.1}$$

Based on the results in [25], Ho [34], Ho et al. [37], and Xu [71] suggested a design method to obtain a robust ("non-fragile") PI (PID) controller by choosing the controller nominal parameters, $\bar{\theta}_c^o$, as the center of the circle (sphere) with the largest radius inscribed inside the controller stabilizing plane (three-dimensional space) for the particular controlled process model under consideration. This radius is the maximum l_2-norm parametric stability margin with respect to the perturbations of the controller parameters and the same as stated by (12.1). The particular case of the First-Order-Plus-Dead-Time (FOPDT) models is considered by Silva et al. [60]. It is important to note that this design method does not take into account any other design criteria, such as the control system performance with changes in the set-point or load-disturbances, the controller output requirements, or the control system robustness to changes in the controlled process characteristics.

To include some type of time-domain performance specifications into the design of non-fragile controllers, Silva et al. [62] considered only the controller parameter

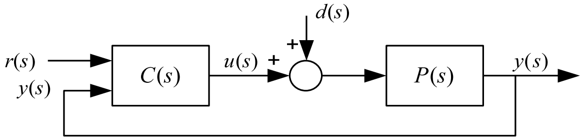


Fig. 12.1 Closed-loop control system

sets that lie inside a box of an arbitrary size defined in the parameter stabilizing space to mitigate to some extent the controller fragility problem and to perform a search inside this box analyzing the servo-control transient responses to select the controller parameter set that meets, or approximately meets, the performance specification. However, they do not provide any criteria for the selection of the box size, the allowed range of variation of the controller parameters, or a quantitative measure to establish when a controller can be considered non-fragile.

The concept of fragility of the PID controllers in the above-cited references is related to the minor variation of its parameters that make the control system unstable; therefore, it is more a stability margin measure. If we consider that a modern PID controller design method must take into account the closed-loop performance to changes in its inputs, set-point and load-disturbance, and its robustness to changes in the controlled process characteristics, then it is evident that from the designer point of view, it is very important that these characteristics be preserved when fine-tuning the controller. In addition, if this is not possible, then there should be at least some sort of information of how such changes in the controller parameters affect the control system robustness and performance. Taking this into account, the use of the PID controller fragility definition and tools introduced by Alfaro et al. [2, 4] are considered more suitable for the fragility analysis of the PI and PID controller tuning rules, as a measure of the control system loss of robustness and/or performance when the controller parameters change.

12.3 Problem Formulation and Framework

Consider the closed-loop control system of Fig. 12.1, where $P(s)$ and $C(s)$ are the *controlled process model* and the *controller* transfer function, respectively. In this system, $r(s)$ is the *set-point*, $u(s)$ is the *controller output signal*, $d(s)$ is the *load-disturbance*, and $y(s)$ is the *controlled process variable*. The parameters of the controlled process model transfer function, $P(s)$, will be considered constant for the fragility analysis.

We will consider the controlled processes represented by the First- or Second-Order-Plus-Dead-Time (FOPDT, SOPDT) models given by the transfer function:

$$P(s) = \frac{K\mathrm{e}^{-Ls}}{(Ts+1)(aTs+1)}, \qquad \tau_o = L/T, \tag{12.2}$$

were K is the process static *gain*, T is the main *time constant*, a the *ratio of its two time constants* ($0 \leq a \leq 1.0$), L is the *dead-time*, and τ_o is the *normalized dead-time*.

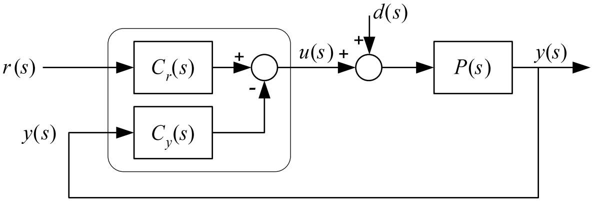


Fig. 12.2 Control system with a two-degree-of-freedom controller

Without the loss of generality, it is supposed that the controller is a Standard Two-Degree-of-Freedom PID (PID_2) controller [68] whose output is as follows:

$$u(t) = K_p\left\{\beta r(t) - y(t) + \frac{1}{T_i}\int_0^t \big[r(\tau) - y(\tau)\big]\,\mathrm{d}\tau + T_d\frac{\mathrm{d}[\gamma r(t) - y(t)]}{\mathrm{d}t}\right\}, \quad (12.3)$$

or

$$u(s) = K_p\left\{\beta r(s) - y(s) + \frac{1}{T_i s}\big[r(s) - y(s)\big] + \frac{T_d s}{\alpha T_d s + 1}\big[\gamma r(s) - y(s)\big]\right\}, \quad (12.4)$$

where K_p is the controller *proportional gain*, T_i is the *integral time constant*, T_d is the *derivative time constant*, β is the *proportional set-point weight*, and γ is the *derivative set-point weight*. In (12.4), α is the *derivative filter constant*, usually $\alpha = 0.10$ [24].

For the analysis, not for the implementation, the PID_2 controller output (12.4) will be rewritten as follows:

$$u(s) = K_p\left(\beta + \frac{1}{T_i s} + \frac{\gamma T_d s}{\alpha T_d s + 1}\right)r(s) - K_p\left(1 + \frac{1}{T_i s} + \frac{T_d s}{\alpha T_d s + 1}\right)y(s), \quad (12.5)$$

and in the compact form shown in Fig. 12.2 as follows:

$$u(s) = C_r(s)r(s) - C_y(s)y(s), \quad (12.6)$$

where

$$C_r(s) = K_p\left(\beta + \frac{1}{T_i s} + \frac{\gamma T_d s}{\alpha T_d s + 1}\right) \quad (12.7)$$

is the part of the PID_2 controller applied to r, the *set-point controller* transfer function, and

$$C_y(s) = K_p\left(1 + \frac{1}{T_i s} + \frac{T_d s}{\alpha T_d s + 1}\right) \quad (12.8)$$

is the part of the PID_2 controller applied to y, the *feedback controller* transfer function.

The closed-loop control system output, $y(s)$, to a change in its inputs, $r(s)$ and $d(s)$, is given by the following:

$$y(s) = \frac{C_r(s)P(s)}{1 + C_y(s)P(s)}r(s) + \frac{P(s)}{1 + C_y(s)P(s)}d(s), \quad (12.9)$$

and the closed-loop characteristic polynomial is as follows:

$$p(s) \doteq 1 + L(s) = 1 + C_y(s)P(s). \quad (12.10)$$

The control system stability depends on the controlled process model $P(s)$, with parameters $\bar{\theta}_p$, and on the feedback controller $C_y(s)$, with parameters $\bar{\theta}_{cy} = \{K_p, T_i, T_d\}$; hence, it is not affected by the controller set-point weights, β and γ.

12.3.1 Control System Robustness Evaluation

There are several quantitative measures of the control system *relative stability* that may be used for the robustness fragility definition, such as the classical *Gain Margin* and *Phase Margin* (A_m, ϕ_m) [29], that provide an indication of the distance from the open-loop transfer function, $L(j\omega)$, frequency response, or Nyquist curve, to the critical point $(-1, 0)$ on the open-loop polar graph. There is also the parametric *Gain Ratio* and the *Delay Ratio* of the *robustness plot* of Gerry and Hansen [28], which defines a parametric robustness region.

Another way to express the system robustness is by using the *Stability Margin*, which is the shortest distance from the Nyquist curve to the critical point [15]. This distance is the reciprocal of the maximum peak of the sensitivity function, or *Maximum Sensitivity* (M_S) [12], defined as follows:

$$M_S \doteq \max_{\omega} \left| S(j\omega) \right| = \max_{\omega} \left| \frac{1}{1 + C_y(j\omega)P(j\omega)} \right|. \tag{12.11}$$

The use of the maximum sensitivity as a robustness measure has the advantage that lower bounds to the gain and phase margins can be assured according to the following [12]:

$$A_m > \frac{M_S}{M_S - 1}, \qquad \phi_m > 2\sin^{-1}\left(\frac{1}{2M_S}\right). \tag{12.12}$$

The relations in (12.12) can be obtained from Fig. 12.3.

For the controller robustness fragility definitions, we will use the maximum sensitivity, M_S, as the indication of the closed-loop control system robustness.

12.3.2 Control System Performance Evaluation

The performance of the closed-loop control system may be evaluated with diverse indices such as those related to the integrated error, the difference between the controlled variable set-point and its real value, IAE, ITAE, ISE, and ITSE, and is given in general by the following:

$$J_{eg} \doteq \int_0^{\infty} t^m \left| e(t) \right|^n \mathrm{d}t, \tag{12.13}$$

or with other characteristics of the time response to a set-point or a load-disturbance change such as the overshoot, rise- or settling-time, and peak error, or decay ratio.

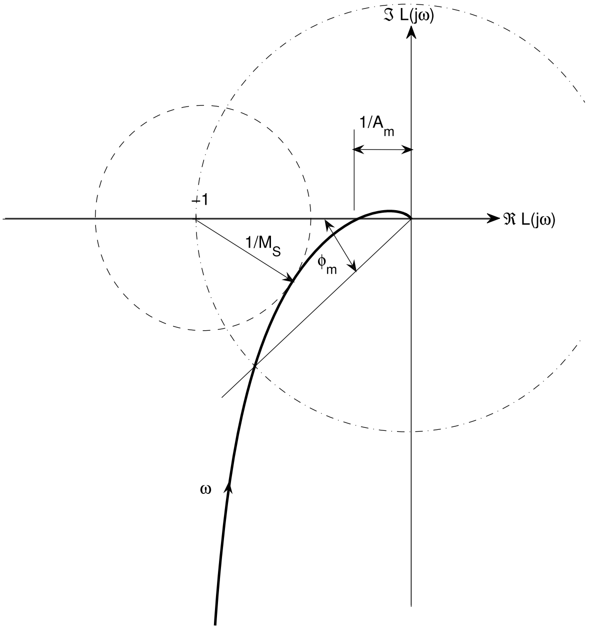


Fig. 12.3 Definition of the control system relative stability margins

For the controller performance fragility evaluation, it is desirable to use a performance indicator that takes into account economic considerations, or an economic performance measure, as does the integrated error (IE) [59]. Taking this into account, to avoid the cancellation of positive and negative errors, we will select, as the control system performance measure, the integrated absolute error, $m = 0$, $n = 1$ in (12.13); given by the following:

$$J_e \doteq \int_0^\infty \left|e(t)\right| \mathrm{d}t = \int_0^\infty \left|r(t) - y(t)\right| \mathrm{d}t. \tag{12.14}$$

12.4 Delta Epsilon Fragility Indices

The concept of PID controllers fragility, the Delta-Epsilon-Fragility Index $\mathrm{FI}_{\Delta\varepsilon}$ and their application to define when a controller is considered *fragile*, *non-fragile* or *resilient*, introduced in [2], are related to the closed-loop control system loss of robustness when the controller parameters are perturbed. These will be extended here to include the control system loss of performance. Then, in our context, the *PID controller fragility* is an indication of the reduction of the closed-loop control system robustness and/or performance when the controller parameters are perturbed, and not a measure of the system stability margin in the controller parameters space.

For the fragility analysis, the controlled process will be represented by a nominal model of the fixed parameters, $\bar{\theta}_p^o$, obtained at the control system normal operation point. This model is used for tuning the controller; then, the controller nominal parameters are $\bar{\theta}_c^o$ and their delta epsilon perturbations, $\delta\varepsilon$. In the following, $\delta\varepsilon$ denotes the variation of each individual controller parameter and $\Delta\varepsilon$ that all controller parameters are perturbed.

12.4.1 Robustness Fragility

The controller *Delta-Epsilon-Robustness-Fragility Index* relates the control system loss of robustness to its nominal robustness and is given by the following:

$$\mathrm{RFI}_{\Delta\varepsilon} \doteq \frac{M^m_{S\Delta\varepsilon}}{M^o_S} - 1 = \frac{\max\{M_S((1 \pm \delta\varepsilon)\bar{\theta}^o_c)\}}{M_S(\bar{\theta}^o_c)} - 1, \tag{12.15}$$

where $M^m_{S\Delta\varepsilon}$ and M^o_S are the control system extreme and the nominal maximum sensitivity, respectively.

The *extreme maximum sensitivity*, $M^m_{S\Delta\varepsilon}$, represents the highest loss of robustness of the control system when all the parameters, $\bar{\theta}_c$, of the controller have been perturbed by the same $\delta\varepsilon$ amount from their nominal values, $\bar{\theta}^o_c$, considering all the possible combinations of the perturbed parameters.

In the ideal case, for a completely delta epsilon robustness-resilient (or absolutely robustness-non-fragile) controller, $\mathrm{RFI}_{\Delta\varepsilon} = 0$, the controller would not lose robustness when its nominal parameters, $\bar{\theta}^o_c$, are perturbed by $\delta\varepsilon$.

The relative influence of a $\delta\varepsilon$ change in the controller parameter p_i over its robustness fragility can be obtained with the *Parametric-Delta-Epsilon-Robustness-Fragility Index* given by the following:

$$\mathrm{RFI}^{p_i}_{\delta\varepsilon} \doteq \frac{M^{p_i}_{S\delta\varepsilon}}{M^o_S} - 1 = \frac{\max\{M_S((1 \pm \delta\varepsilon)p_i, \bar{\theta}^o_c)\}}{M_S(\bar{\theta}^o_c)} - 1. \tag{12.16}$$

The final *fine-tuning* of the control-loop is considered the most probable cause of major variations in the controller parameters, for example, it is possible to see commissioning changes up to 10% or 20% in their values. Considering this, the *Delta 20 Robustness-Fragility Index* can be defined to measure the maximum loss of the control system robustness when a change of up to 20% occurs in one or more of the nominal controller parameters values and is given by the following:

$$\mathrm{RFI}_{\Delta 20} \doteq \frac{M^m_{S\Delta 20}}{M^o_S} - 1. \tag{12.17}$$

Based on the $\mathrm{RFI}_{\Delta 20}$, the controller robustness fragility degree is defined as follows:

Controller Robustness Fragility Definitions

- *Robustness Fragile PID controller*: a PID controller is robustness-fragile if its delta 20 robustness fragility index is higher than 0.50, $\mathrm{RFI}_{\Delta 20} > 0.50$.
- *Robustness Non-Fragile PID controller*: a PID controller is robustness-non-fragile if its delta 20 robustness fragility index is less than or equal to 0.50, $\mathrm{RFI}_{\Delta 20} \leq 0.50$.
- *Robustness Resilient PID controller*: a PID controller is robustness-resilient if its delta 20 robustness fragility index is less than or equal to 0.10, $\mathrm{RFI}_{\Delta 20} \leq 0.10$.

A controller will be considered *robustness-fragile* if the control system loses more than 50% of its robustness when all its parameters change up to 20%; otherwise, it is *robustness-non-fragile*. In addition, a controller will be *robustness-resilient* if the control system does not lose more than 10% of its robustness when its parameters change up to 20%. A controller with a low robustness-fragility will allow final fine-tuning without a significant reduction in the control system robustness.

The selection of a ±20% ($\Delta 20$) change in the controller parameters for the robustness fragility definition above considers a 10% reduction in the control system robustness as marginal and a 50% reduction as the maximum allowed limit because it will turn a highly robust system, with M_S lower than 1.4, into one with a minimally acceptable robustness, M_S of approximately 2.0. Although, using (12.15) and (12.16) it is possible to evaluate the effect of any other particular $\delta\varepsilon$ perturbation in one or more controller parameters.

12.4.2 Performance Fragility

The controller *Delta-Epsilon-Performance-Fragility Index* relates the control system loss of performance to its nominal performance and is given by the following:

$$\mathrm{PFI}_{\Delta\varepsilon} \doteq \frac{J^m_{e\Delta\varepsilon}}{J^o_e} - 1 = \frac{\max\{J_e(1 \pm \delta\varepsilon)\bar{\theta}^o_c\}}{J_e(\bar{\theta}^o_c)} - 1, \tag{12.18}$$

were $J^m_{e\Delta\varepsilon}$ and J^o_e are the extreme and the nominal performance, respectively, measured by the integrated absolute error (12.14).

The relative influence of a $\delta\varepsilon$ change in the controller parameter p_i over its performance fragility can be obtained with the *Parametric Delta-Epsilon-Performance-Fragility Index*, given by the following:

$$\mathrm{PFI}^{p_i}_{\delta\varepsilon} \doteq \frac{J^{p_i}_{e\delta\varepsilon}}{J^o_e} - 1 = \frac{\max\{J_e((1 \pm \delta\varepsilon)p_i, \bar{\theta}^o_c)\}}{J_e(\bar{\theta}^o_c)} - 1. \tag{12.19}$$

Considering the same 20% change in the controller parameters used in the robustness fragility definitions above, the *Delta 20 Performance-Fragility Index* could define the maximum loss of the control system performance when a change of up to 20% occurs in one or more of the nominal controller parameters values given by the following:

$$\mathrm{PFI}_{\Delta 20} \doteq \frac{J^{m}_{e\Delta 20}}{J^{o}_{e}} - 1. \tag{12.20}$$

Based on the $\mathrm{PFI}_{\Delta 20}$, the controller performance fragility degree is defined as follows:

Controller Performance Fragility Definitions

- *Performance Fragile PID controller*: a PID controller is performance-fragile if its delta 20 performance fragility index is higher than 0.50, $\mathrm{PFI}_{\Delta 20} > 0.50$.
- *Performance Non-Fragile PID controller*: a PID controller is performance-non-fragile if its delta 20 performance fragility index is less than or equal to 0.50, $\mathrm{PFI}_{\Delta 20} \leq 0.50$.
- *Performance Resilient PID controller*: a PID controller is performance-resilient if its delta 20 performance fragility index is less than or equal to 0.10, $\mathrm{PFI}_{\Delta 20} \leq 0.10$.

A controller will be considered *performance-fragile* if the control system loses more than 50% of its performance when all its parameters change up to 20%; otherwise, it is *performance-non-fragile*. In addition, a controller will be *performance-resilient* if the control system does not lose more than 10% of its performance when its parameters change up to 20%.

The controller performance fragility must be evaluated for the servo-control response, $\mathrm{PFI}_{r\Delta 20}$, and for the regulatory control response, $\mathrm{PFI}_{d\Delta 20}$.

In a similar way as was indicated for the robustness-fragility evaluation, using now (12.18) and (12.19), the controller performance-fragility may be evaluated for any other $\delta\varepsilon$ perturbation in one or more controller parameters.

12.4.3 Fragility Balance

To define when a controller is or is not a robustness- or performance-fragility-balanced controller, we must obtain first its *average parametric delta-epsilon-robustness-fragility index*:

$$\mathrm{RFI}^{a}_{\delta\varepsilon} \doteq \frac{1}{n}\sum_{i=1}^{n}\mathrm{RFI}^{p_i}_{\delta\varepsilon}, \tag{12.21}$$

and its *average parametric delta-epsilon-performance-fragility index*:

$$\mathrm{PFI}^{a}_{\delta\varepsilon} \doteq \frac{1}{n}\sum_{i=1}^{n} \mathrm{PFI}^{p_i}_{\delta\varepsilon}, \tag{12.22}$$

where the number of parameters is two for a PI ($n = 2$), and three for a PID ($n = 3$).

Based on the parametric delta-epsilon-fragility indices, the controller fragility balance is defined as follows:

Controller Fragility Balance Definitions

- *Robustness-Fragility-Balanced PID controller*: a robustness-fragility-balanced PID controller is one in which all its parametric robustness delta-epsilon-fragility indices are within a selected $\pm\sigma\%$ band (usually $\pm 25\%$) centered on its average parametric delta-epsilon-robustness-fragility index, $\mathrm{RFI}^{a}_{\delta\varepsilon}$; otherwise, it is a robustness-fragility-unbalanced controller.
- *Performance-Fragility-Balanced PID controller*: a performance-fragility-balanced PID controller is one in which all its parametric performance delta-epsilon-fragility indices are within a selected $\pm\sigma\%$ band (usually $\pm 25\%$) centered on its average parametric delta-epsilon-performance-fragility index, $\mathrm{PFI}^{a}_{\delta\varepsilon}$; otherwise, it is a performance-fragility-unbalanced controller.

The robustness- or performance-fragility unbalance of a controller is caused by the controller parameter with the highest parametric robustness- or performance-fragility index.

12.5 Fragility of the PI and PID Tuning Rules

We will review the "fragility" evaluation of several tuning rules available in the control system literature before analyzing the robustness and performance fragility of several tuning rules with the fragility criteria stated in Sect. 12.4.

12.5.1 Assuring a Controller Stability Margin

Using the characterization of all the stabilizing PID controllers for First-Order-Plus-Dead-Time models, Silva et al. [60–62] analyzed several tuning rules for the controllers robustness with respect to small perturbations in their parameters. To obtain a good parametric stability margin, the controller integral gain value was forced to lie inside a box located 20% from the stabilizing integral gain boundaries for the fixed proportional and derivative gains provided by the tuning rule. As a result, the range of the model normalized dead-time that ensures the above controller robustness criteria for a tuning rule are as follows:

- Ziegler–Nichols [73]: $0 < \tau_o < 1.07$.
- Chien–Hrones–Reswick [22]: $0.37 < \tau_o$.
- Cohen–Coon [23]: $0 < \tau_o < 8.53$.
- Morari–Zafiriou IMC [53]: $0.37 < \tau_o$ $(\lambda/L = 0.25)$.

The controller robustness in this analysis is understood as a good l_2-norm parametric stability margin in the controller parametric space.

12.5.2 Tuning Rules for the Test

For the evaluation of the fragility of the PI and PID controllers tuning rules, a set of methods was selected ranging from the classic performance optimized rules to modern methods that allow dealing with the control system performance/robustness trade-off.

The selected tuning methods are the following:

1. Performance Optimization:
   - López et al. [48] [L_{IAE}, L_{ITAE}]—controllers: 1DoF PI and PID; design criteria: optimize the integrated absolute error (IAE) or the integrated time weighted absolute error (ITAE) to a load-disturbance step change; controlled process information: FOPDT model; application range: $0.1 \le \tau_o \le 1.0$.
   - Rovira et al. [58] [R_{IAE}, R_{ITAE}]—controllers: 1DoF PI and PID; design criteria: optimize the IAE or the ITAE error criteria to a set-point step change; controlled process information: FOPDT model; application range: $0.1 \le \tau_o \le 1.0$.
   - Taguchi and Araki [64] [*T&A*]—controllers: Two-Degree-Of-Freedom (2DoF) PI and PID; design criteria: two-step optimization of a weighted integrated error (1. d response, 2. r response); controlled process information: FOPDT model; application range: $0.1 \le \tau_o \le 1.0$.
2. From a Test Batch of Processes:
   - Åström and Hägglund Kappa-Tau [12] [*KT*]—controllers: 2DoF PI and PID; design criteria: dominant pole design with two robustness levels, $M_S \in \{1.4, 2.0\}$; controlled process information: K, T, L; application range: $0.1 \le \tau_o \le 6$.
   - Åström and Hägglund Approximated M constrained Integral Gain Optimization [14] [*AMIGO*]—controllers: 2DoF PI and PID; design criteria: maximize the controller integral gain subject to a constraint in the robustness, $M_S = 1.4$; controlled process information: K, T, L; application range: $0.001 \le \tau_o \le 50$.
3. Internal Model Control (IMC)-Based:
   - Skogestad Simple Control [63] [*SIMC*]—controllers: 1DoF PI and PID; design criteria: modified servo-control direct synthesis (IMC) for fast servo- and

regulatory control responses with a robustness $M_S = 1.59$; controlled process information: FOPDT model for PI tuning and SOPDT model for PID tuning; application range: $\tau_o \geq 0.1$.

- Ali and Majhi Percent Overshoot [8] [POS_S, POS_T]—controllers: 1DoF PI and PID; design criteria: set-point response percent overshoot specification, POS_S smooth control 0% OS ($M_S = 1.38$), POS_T tight control 10% OS ($M_S = 1.71$); controlled process information: FOPDT model for PI tuning and SOPDT model for PID tuning; application range: $\tau_o \geq 0.1$.

4. Multi-Objective Optimization:

- Tavakoli et al. [67] [MOO]—controllers: 2DoF PI; design criteria: two-step optimization considering servo- and regulatory control performance (IAE), control effort smoothness (TV_u), and robustness ($M_S \leq 2.0$); controlled process information: FOPDT model; application range: $0.1 \leq \tau_o \leq 2.0$.

5. Model Reference Design:

- Alfaro et al. [7] [PI$_{2M_S}$]—controllers: 2DoF PI; design criteria: two-step non-oscillatory servo- and regulatory control model reference optimization with a robustness restriction, $M_S \in \{1.4, 1.6, 1.8, 2.0\}$; controlled process information: FOPDT model; application range: $0.1 \leq \tau_o \leq 2.0$.

The above selected controller tuning rules will be used to illustrate the different robustness- and performance-fragility analysis that can be conducted. Although the intention of this chapter is not to present an exhaustive fragility analysis of all these tuning rules, the results allow evaluating and comparing their fragility issues.

12.5.3 Nominal Robustness and Performance of Tuning Rules

For the fragility analysis, a unit gain normalized controlled process model will be used, obtained by applying the transformation, $\hat{s} = Ts$, in (12.2), given by the following:

$$\hat{P}(\hat{s}) = \frac{\mathrm{e}^{-\tau_o \hat{s}}}{(\hat{s}+1)(a\hat{s}+1)}, \quad 0 \leq a \leq 1.0. \tag{12.23}$$

With each tuning rule under evaluation, the controller nominal normalized parameters, $\hat{\theta}_c^o$, were found for the normalized dead-times, τ_o, in the range of application of the rule. For each one of these controllers, the nominal control system robustness, M_S^o, the nominal servo-control performance, J_{er}^o, and the nominal regulatory control performance, J_{ed}^o, were computed.

In all the evaluations, the PID controller derivative mode will be applied only to the feedback signal y (i.e., $\gamma = 0$), which will only affect the servo-control performance. In addition, the PID derivative mode will include a derivative filter with $\alpha = 0.1$, even for testing such tuning methods obtained using a non-proper, “ideal” PID controller.

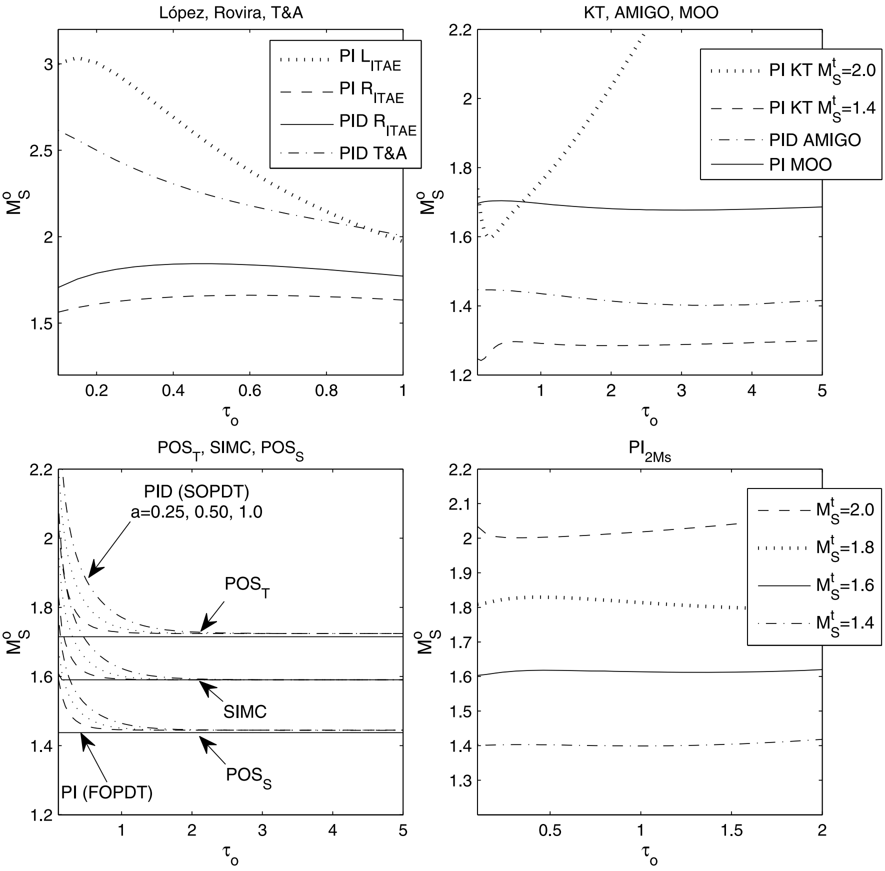


Fig. 12.4 Nominal robustness of the tuning rules

Nominal Robustness The nominal robustness, M_S^o, of the tuning rules under evaluation are shown in Fig. 12.4. As can be seen, the robustness of the control systems with the regulatory control optimized-performance controllers, PI L_{ITAE} and PID *T&A*, is very poor; in no case was the minimum robustness level of $M_S \leq 2.0$ obtained. The PI/PID L_{IAE}, PID L_{ITAE} and PI *T&A*, not shown, have even lower robustness. In the servo-control optimized-performance controller case, the PI/PID R_{ITAE} have middle-range robustness, but the R_{IAE} has robustness of up to 20% lower.

From the Kappa-Tau and *AMIGO* group of controllers, only the *KT* PI for high robustness with $M_S^o \approx 1.3$ and the *AMIGO* PID with $M_S^o \approx 1.42$ are near to their robustness target design level of $M_S^t = 1.4$. As has been reported elsewhere, the *KT* and *AMIGO* methods for minimum robustness ($M_S^t = 2.0$) did not achieve their design robustness criteria.

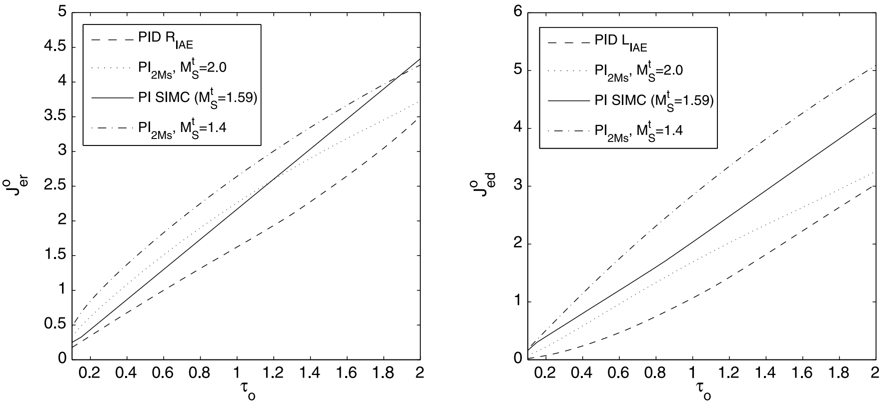


Fig. 12.5 Nominal servo- and regulatory control performance of the tuning rules

It is also noted from this figure that the Multi-Objective-Optimization Method (MOO) provides a nearly constant robustness level of $M_S^o \approx 1.7$ for models with normalized dead-times, even further than its range of application.

The IMC-based *SIMC*, POS_T and POS_S methods are designed to provide specific target robustness levels: *SIMC* $M_S^t = 1.59$, POS_T $M_S^t = 1.71$, and POS_S $M_S^t = 1.38$. The *SIMC* and POS_T target robustness are achieved without problems with a PI controller for FOPDT models, but the POS_S nominal robustness is $M_S^o = 1.44$ for FOPDT models. In the PID controller case tuned with the SOPDT models and due to these methods obtained using zero/pole cancellations with an ideal PID controller (without derivative filter), when they are applied to a "real" proper PID controller, the zero/pole cancellation does not take place, affecting the robustness of the control system. This is particularly severe for time-constant-dominant ($\tau_o < 1$) SOPDT models with time constants ratios of $a > 0.25$.

As seen in Fig. 12.4 with the PI_{2Ms} tuning method, it is possible to select four different robustness levels that are all accomplished successfully.

Nominal Performance For future use, the servo-control and regulatory control nominal performance of several of the evaluated tuning methods are shown in Fig. 12.5. As expected, at the performance evaluation side are the performance-optimized tunings of Rovira (servo-control) and López (regulatory control), which the methods showing the best performance to step changes in the set-point or the load-disturbance. Also noted in this figure is the constantly presented performance/robustness trade-off when the development of the tuning rule takes into account the control system robustness.

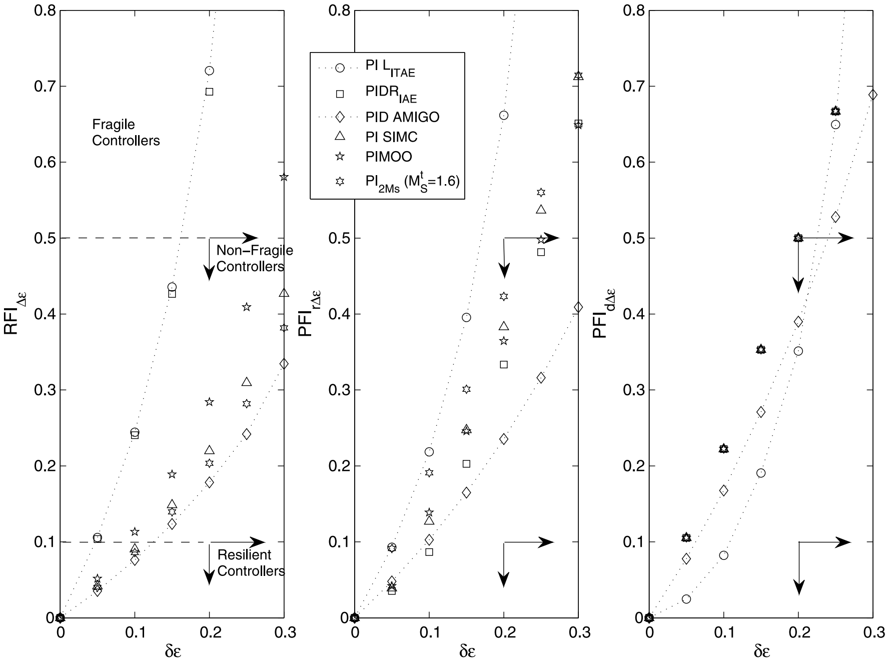


Fig. 12.6 Delta-epsilon-robustness- and performance-fragility

12.5.4 Delta-Epsilon and Parametric Robustness Fragility

The Delta-Epsilon-Fragility Indices, (12.15) and (12.18), can be used to analyze the effect of a change in the controller nominal parameters over the control system robustness and performance.

12.5.4.1 Robustness and Performance Delta-Epsilon Fragility

The Delta-Epsilon-Robustness- and Performance-Fragility Index Plots of Fig. 12.6 show the effect of the $\delta\varepsilon$ changes on the controller parameters and allow a comparison of the fragility of the tuning methods. In this particular case, the comparison was made using a normalized controlled process model (12.23) with $\tau_o = 0.50$.

As can be seen, the performance-optimized controllers (PI L_{ITAE}, PID R_{IAE}) lose robustness very quickly when the controller parameter are perturbed. Up to 25% of their robustness is lost with a 10% change in the controller parameters. At the other extreme, the controller tuned with the *AMIGO* method, with a high nominal robustness $M_S^o \approx 1.3$, loses only 8% of its robustness with the same 10% change in its parameters. For the particular case of the model considered, with the exception of the López and Rovira controllers, all are robustness non-fragile controllers, but none are robustness-resilient.

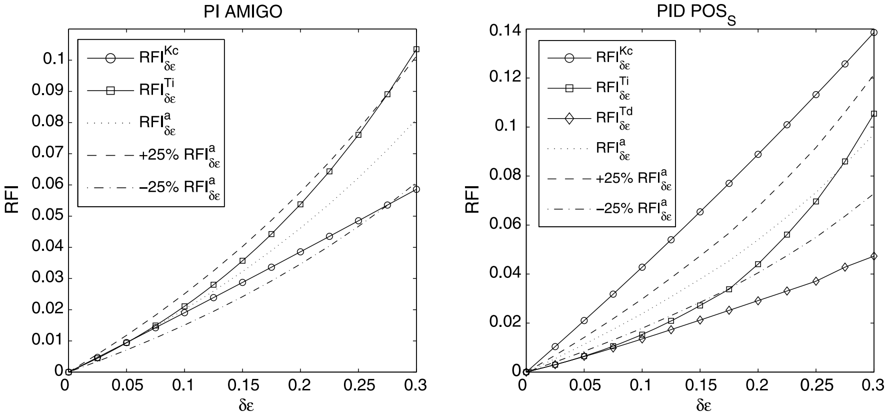


Fig. 12.7 Controller robustness-fragility balance

It is also noted that the servo-control performance fragility has the same behavior as shown by the robustness fragility. In addition, and excluding the PI L_{ITAE} and the PID *AMIGO* controllers, all the controllers have the same regulatory control performance fragility proportional to the $\delta\varepsilon$ change in the controller parameters.

12.5.4.2 Controller Fragility Balance

The parametric delta-epsilon-robustness-fragility indices can be used to evaluate the balance of controller fragility [4].

As can be seen in Fig. 12.7, the PI *AMIGO* is a robustness-fragility-balanced controller, while the PID POS_S is an unbalanced controller. It is also noted that in the PI *AMIGO* case, the control system robustness is affected more by a change in the controller integral time constant, T_i, than in its gain, K_p. In the PID POS_S case, the controller proportional gain, K_p, is the parameter that deteriorates more of the control system robustness and causes its fragility unbalance.

The controller final fine-tuning will be safer if the controller is fragility-balanced (robustness and performance); in such a case, each controller parameter variation will have a similar effect over the control system robustness and/or performance.

12.5.5 Robustness and Performance Fragility of the Tuning Rules

The Delta-Epsilon-Fragility Indices in the above sections showed the control system loss of robustness and performance for $\delta\varepsilon$ changes on the controller parameters from very small values up to values greater than the 20% change used to define when a controller is considered robustness- or performance *fragile*, *non-fragile*, or *resilient*.

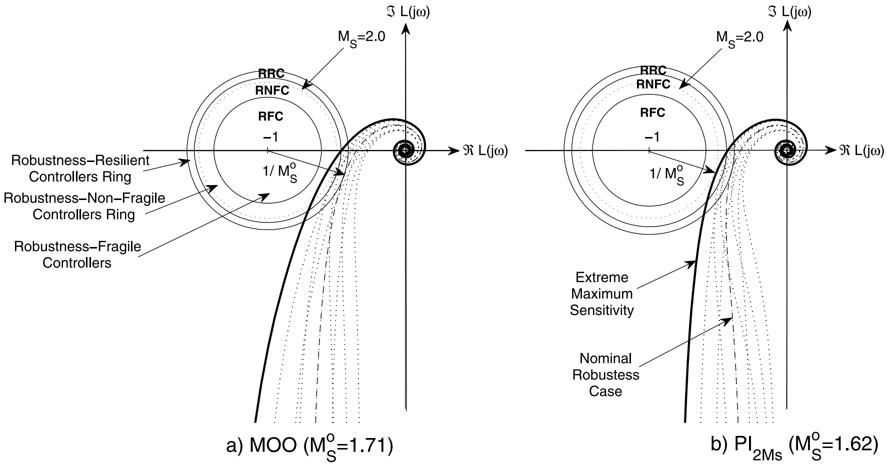


Fig. 12.8 Controller delta 20 robustness-fragility rings

In the following sections, the Delta 20 fragility is considered first to visually show the robustness fragility of a tuning rule for a specific controlled process model and, afterward, to analyze the robustness and performance fragility of a tuning rule as a whole. In the latter case, the evaluation will show if a tuning rule can be considered as *globally robustness-* and/or *performance-resilient*, *non-fragile* or *fragile*.

12.5.5.1 Delta 20 Robustness-Fragility Rings

We will introduce here the *Delta 20 Robustness-Fragility Rings*, which is a simple tool that uses the open-loop transfer function, $L(j\omega)$, Nyquist curve of the nominal, and Delta 20 perturbed controllers to provide an indication of the control system robustness-fragility. The plot includes the robustness-fragility rings to show the areas in the $L(j\omega)$ plane that define when the controller is a robustness-resilient controller (RRC), a robustness-non-fragile controller (RNFC), or a robustness-fragile controller (RFC).

Figure 12.8 shows the Delta 20 fragility rings of the *MOO* ($M_S^o = 1.71$) and PI_{2Ms} ($M_S^o = 1.62$) PI controllers for the $\tau_o = 0.5$ model case. As can be seen, both are robustness-non-fragile controllers; their perturbed open-loop Nyquist curves enter the robustness-non-fragile controller (RNFC) ring area. However, the *MOO* PI controller is more robustness-fragile ($\mathrm{RFI}_{\Delta 20} = 0.284$) than the PI_{2Ms} controller ($\mathrm{RFI}_{\Delta 20} = 0.202$), with an extreme maximum sensitivity of $M_{S\Delta 20}^m = 2.20$, which is higher than the normal minimal robustness level of $M_S = 2.0$. The extreme maximum sensitivity of the PI_{2Ms} controller is $M_{S\Delta 20}^m = 1.95$ in this case.

12.5.5.2 Robustness- and Performance-Fragility Plots of the Tuning Rules

In the following figures, the nominal robustness, M_S^o, the Delta 20 Robustness-Fragility Index $\mathrm{RFI}_{\Delta 20}$, and the Parametric Delta 20 Robustness-Fragility Indices, $\mathrm{RFI}_{\delta 20}^{K_p}$, $\mathrm{RFI}_{\delta 20}^{T_i}$, and $\mathrm{RFI}_{\delta 20}^{T_d}$, of two of the selected tuning rules are presented.

To provide useful information to the control system designer, the tuning rule-fragility evaluation must be conducted within the application range of the rule, covering the model normalized dead-time range for which the rule was designed. This will provide a picture of the global fragility of the tuning rule.

Simple Control—*SIMC* Although the IMC-based *SIMC* tuning rule does not impose any specific restriction to the model normalized dead-time, it will be evaluated in the $0.1 \leq \tau_o \leq 2$ range covering the time-constant-dominant and dead-time-dominant models.

The *SIMC* robustness-fragility indices are shown in Fig. 12.9. As can be seen, although the PI controller (FOPDT model) produces control systems with a constant nominal robustness of $M_S^o = 1.59$, its robustness-fragility is affected by the model normalized dead-time. The PI controller became more robustness-fragile as the normalized dead-time increased due to the raising integral time parametric robustness-fragility, while its gain parametric robustness-fragility remained constant.

For the PID controller (SOPDT model), it can be seen that the control system nominal robustness and the controller fragility depend not only on the model normalized dead-time, τ_o, but also on the model time-constants ratio, a. This is especially evident for lower values of τ_o and higher values of a.

The servo-control and the regulatory control nominal performance and the performance fragility of the *SIMC* method are shown in Figs. 12.10 and 12.11, respectively. The higher servo-control performance is obtained with the PI controller, but it is more performance-fragile. However, the servo-control Delta 20 and parametric performance fragility are almost constant; therefore, the expected adverse effect of a 20% change in the controller parameters over the servo-control performance could be predicted.

Regarding the regulatory control nominal performance, there is no significant difference between the PI and the PID controllers, and it decreased (J_{ed} increase) linearly with τ_o. It is also noted that, for $\tau_o \leq 0.8$, the regulatory control performance-fragility is constant: $\mathrm{PFI}_{d\Delta 20} = 0.50$, $\mathrm{PFI}_{d\delta 20}^{K_p} = 0.25$, $\mathrm{PFI}_{d\delta 20}^{T_i} = 0.20$, $\mathrm{PFI}_{d\delta 20}^{T_d} \approx 0$; these characteristics will be analyzed in more detail later. For higher τ_o values, the controller performance-fragility decreased.

Robust Tuning—$\mathrm{PI}2_{Ms}$ The $\mathrm{PI}2_{Ms}$ tuning addressed the performance/robustness trade-off, providing rules for four target robustness levels. Three are shown in Fig. 12.12: the normal minimum robustness level ($M_S^t = 2.0$), an intermediate level ($M_S^t = 1.6$), and a high robustness level ($M_S^t = 1.4$).

Although all the achieved robustness levels are nearly constant for $0.1 \leq \tau_o \leq 2.0$, the controller fragility increases with the model normalized dead-time. In addition, the low robustness controllers are more robustness-fragile than the high

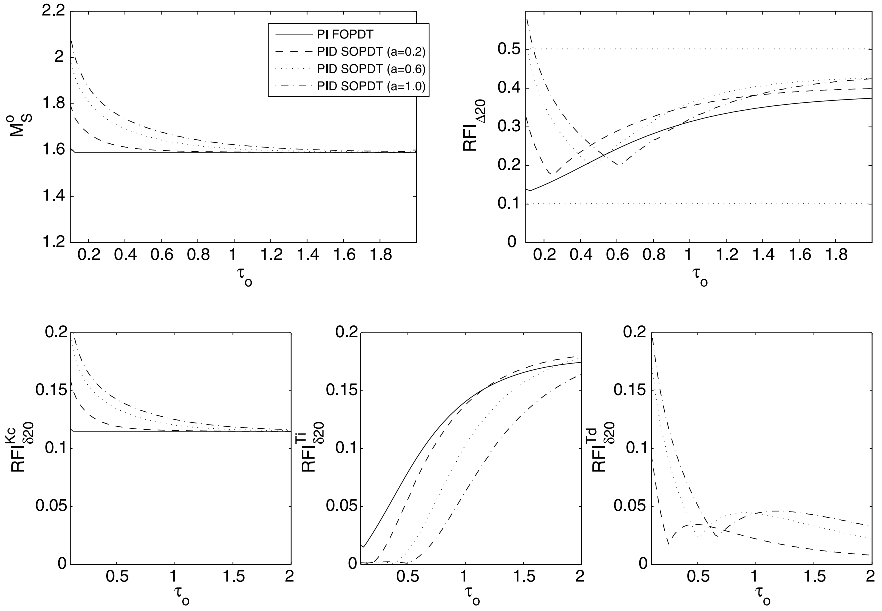


Fig. 12.9 *SIMC* nominal robustness and fragility

robustness ones. All the controllers are robustness-non-fragile, except for $M_S^t = 2.0$ and $\tau_o > 1.6$. It is also noted that the controller gain parametric fragility is nearly constant, except for $M_S^t = 2.0$, and the integral time constant parametric fragility increases linearly with τ_o.

The performance/robustness trade-off can be seen in Fig. 12.13 (servo-control) and Fig. 12.14 (regulatory control). If the control system nominal robustness increases, lower M_S^o, the system nominal performances decreases, higher J_{er}^o and higher J_{ed}^o.

In addition to the robustness-fragility, the servo-control performance-fragility increases with the model normalized dead-time, but in this case, the fragility/ robustness relation is inverted. The high robustness systems are more performance-fragile than the systems with low robustness. This rise in the performance-fragility is due to an increment in the gain parametric performance-fragility when the normalized dead-time increases. The integral time constant servo-control performance fragility is constant for all τ_o and M_S target levels, except for $M_S^t = 2.0$ and $\tau_o > 1.7$.

As noted in Fig. 12.14, the regulatory control performance is affected not only by the model normalized dead-time but also by the control system design robustness level. In this case, more robust control systems have a higher regulatory control performance, which is more evident for higher normalized dead-times.

With the exception of the controllers for $\tau_o \geq 1.5$ tuned for low robustness ($M_S^t = 2.0$), the regulatory control performance-fragility index of all the PI_{2Ms} controllers is $\mathrm{PFI}_{d\Delta 20} = 0.50$, and their parametric performance fragility indices are $\mathrm{PFI}_{d\delta 20}^{K_p} = 0.25$ and $\mathrm{PFI}_{d\delta 20}^{T_i} = 0.20$. For the PI_{2Ms} tuning rule, the regulatory per-

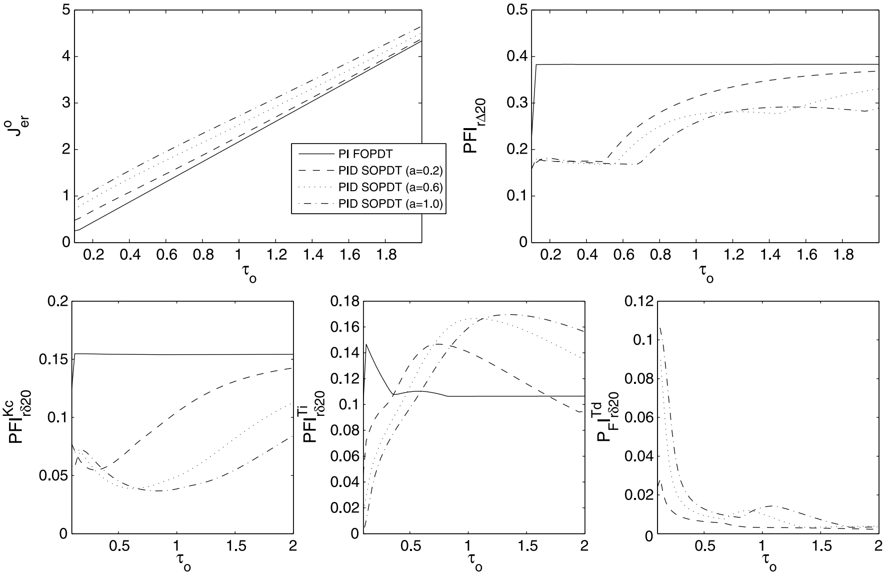


Fig. 12.10 *SIMC* servo-control nominal performance and fragility

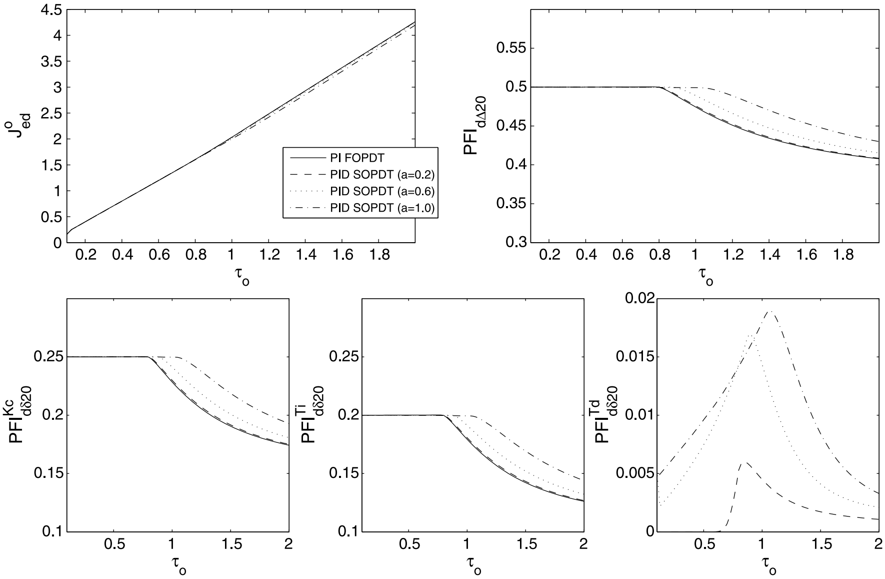


Fig. 12.11 *SIMC* regulatory control nominal performance and fragility

formance fragility is independent of the model normalized dead-time and of the robustness design criteria.

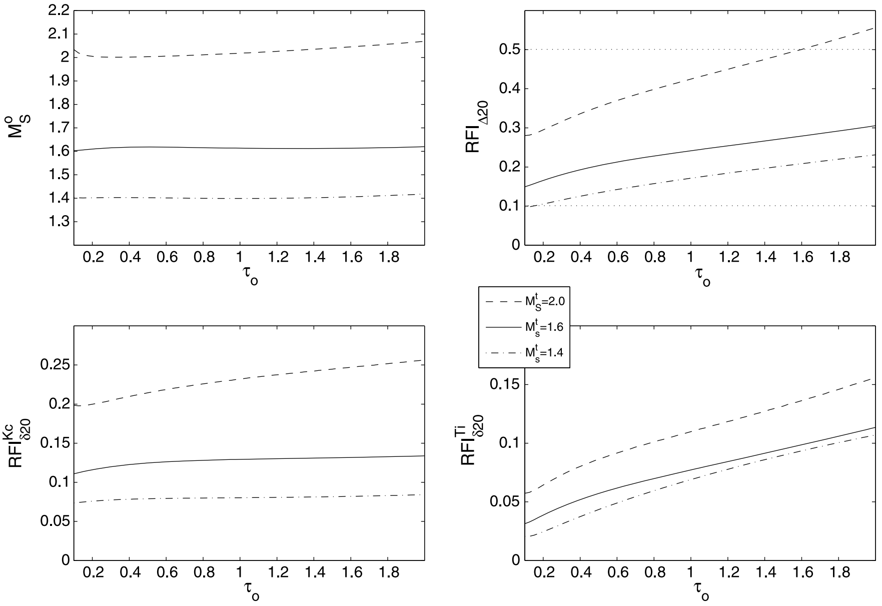


Fig. 12.12 PI$_{2Ms}$ nominal robustness and fragility

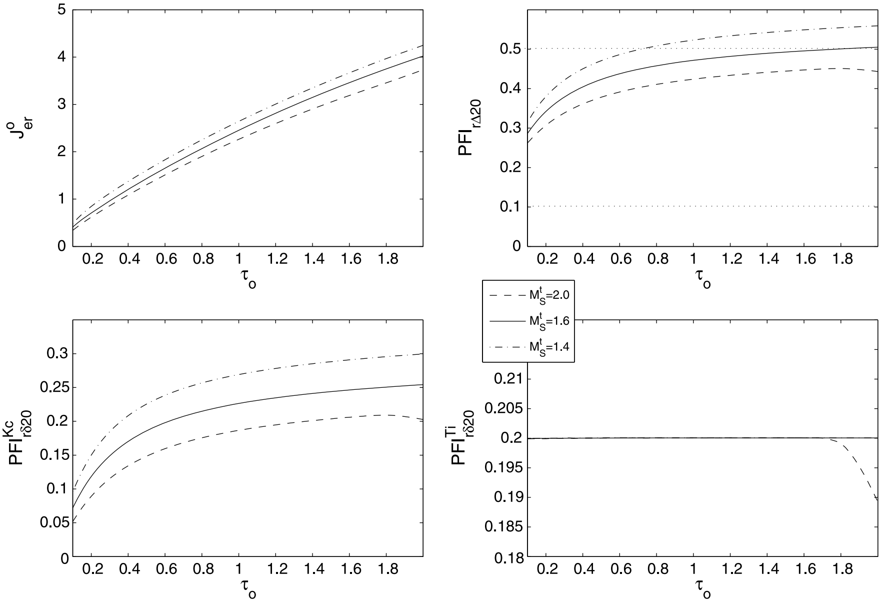


Fig. 12.13 PI$_{2Ms}$ servo-control nominal performance and fragility

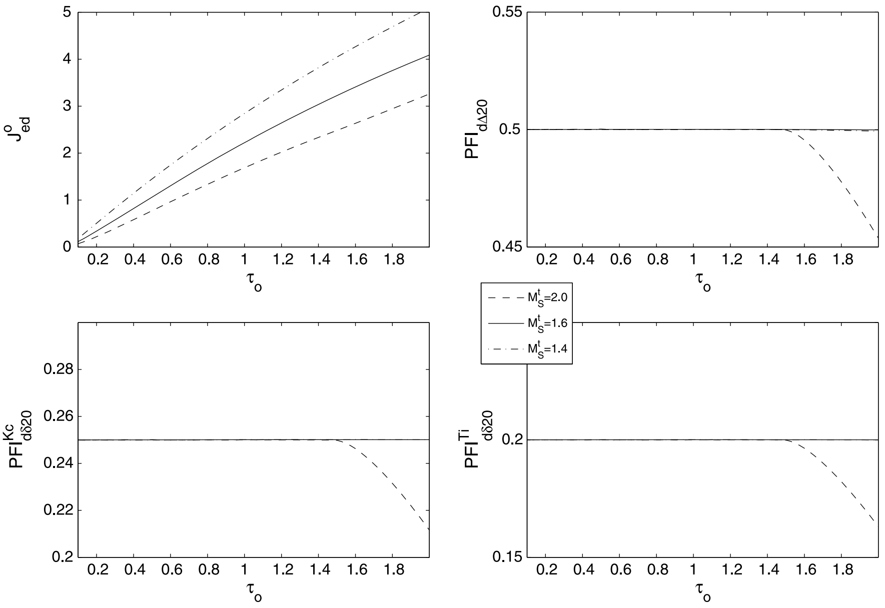


Fig. 12.14 PI$_{2Ms}$ regulatory control nominal performance and fragility

The regulatory control design criterion of the PI$_{2Ms}$ tuning is to obtain a non-oscillatory response to a load-disturbance step change with a specific robustness level. If the regulatory control step response is non-oscillatory, then its integrated absolute error, IAE, is equal to its integrated error, IE, and given by the following [12]:

$$J_{ed} = \int_0^\infty \left|e(t)\right| \mathrm{d}t = \int_0^\infty e(t)\,\mathrm{d}t = \frac{T_i}{K_p}, \tag{12.24}$$

which depends on the controller gain, K_p, and integral time constant, T_i, but not on its derivative time constant, T_d.

Using (12.24) with (12.18) and (12.19), and considering a decrement in the controller gain ($\delta\varepsilon_{K_p^-}$) and an increment in the controller integral time constant ($\delta\varepsilon_{T_i^+}$), the parametric delta-epsilon-performance-fragility indices for the regulatory control are as follows:

$$\mathrm{PFI}_{d\delta\varepsilon}^{K_p} = \frac{\delta\varepsilon_{K_p^-}}{1-\delta\varepsilon_{K_p^-}}, \qquad \mathrm{PFI}_{d\delta\varepsilon}^{T_i} = \delta\varepsilon_{T_i^+}, \qquad \mathrm{PFI}_{d\delta\varepsilon}^{T_d} = 0, \tag{12.25}$$

and the delta-epsilon-performance-fragility index is the following:

$$\mathrm{PFI}_{d\Delta\varepsilon} = \frac{\delta\varepsilon_{K_p^-} + \delta\varepsilon_{T_i^+}}{1-\delta\varepsilon_{K_p^-}} \tag{12.26}$$

Then, considering a 20% reduction in K_p and a 20% increment in T_i, from (12.25) and (12.26), $\text{PFI}^{K_p}_{d\delta 20} = 0.25$, $\text{PFI}^{T_i}_{d\delta 20} = 0.20$, $\text{PFI}^{T_d}_{d\delta 20} = 0$, and $\text{PFI}_{d\Delta 20} = 0.50$.

From this result it may concluded that the above regulatory performance-fragility figures reflect the non-oscillatory behavior of the PI_{2M_S} load-disturbance step response.

These characteristics were also noted with other tuning methods: PI/PID R_{IAE} and R_{ITAE}; PI *T&A* ($\tau_o \leq 1.6$); PID *KT* $M_S = 2.0$ ($\tau_o \geq 1.5$) and PI/PID *KT* $M_S = 1.4$ ($\tau_o \geq 1$); PI *AMIGO* ($\tau_o \geq 0.4$); PI/PID *SIMC* ($\tau_o \leq 0.8$); PI MOO ($\tau_o \geq 0.5$); PI/PID POS_S and POS_T ($\tau_o \leq 0.7$).

12.6 PID Controller Implementation Fragility

Although the PID controller fragility definitions and indices stated in Sect. 12.4 are independent of its implementation, in Sect. 12.5, the evaluation of the tuning rules fragility was made using a *standard* "non-interacting" PID controller (12.4) with $\gamma = 0$, which reduces to the following:

$$u(s) = K_p\left\{\beta r(s) - y(s) + \frac{1}{T_i s}\big[r(s) - y(s)\big] - \frac{T_d s}{\alpha T_d s + 1} y(s)\right\}. \tag{12.27}$$

However, this is not the only available implementation of the PID control algorithm [39, 42, 68]. In commercial industrial controllers, the *series* "interacting" PID is frequently presented by the following:

$$u(s) = K'_p\left\{(\beta - 1)r(s) + \left(\frac{T'_i s + 1}{T'_i s}\right)\left[r(s) - \left(\frac{T'_d s + 1}{\alpha_s T'_d s + 1}\right)y(s)\right]\right\}, \tag{12.28}$$

and in academic and research papers, the *parallel* or "three-gain" PID is often used:

$$u(s) = K_p\big[\beta r(s) - y(s)\big] + \frac{K_i}{s}\big[r(s) - y(s)\big] - \frac{K_d s}{\alpha_p K_d s + 1} y(s). \tag{12.29}$$

Applying parameter conversion factors [24], it is always possible to obtain a standard or parallel PID controller equivalent to a series controller, but only if $T_i \geq 4T_d$, then a series controller equivalent to a standard PID may be found. For PI controllers ($T_d = 0$), the standard and series implementations are equivalent.

The aim here is to evaluate how the PID controller robustness- and performance-fragility is affected by its implementation.

Since most of the tuning rules selected in Sect. 12.5.2 are for PI controllers and the selected rules for PID that achieve nearly constant robustness levels are based on zero/pole cancellation design techniques, the Brosilow and Joseph [B&J] [20] tuning was selected to evaluate the controller implementation influence over its fragility. This is an IMC-based PID tuning for FOPDT-controlled process models; the equations of which were found by expanding the IMC controller equation in a Maclaurin series and taking its first three terms. As IMC-based tuning, the B&J

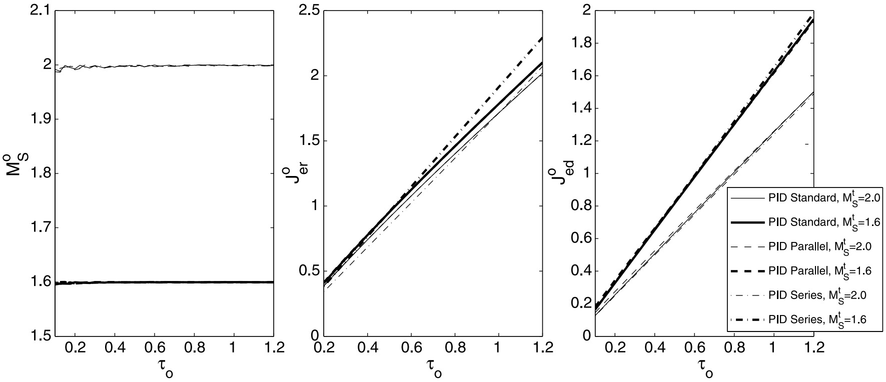


Fig. 12.15 B&J PID nominal robustness and performance

equations include a *tuning parameter*, the control system closed-loop time constant, which addresses the control system performance/robustness trade-off.

For the evaluation and changing of the design parameter, the parameters of a standard PID controller (12.27) were found for two target robustness levels, $M_S^t \in \{1.6, 2.0\}$. The model normalized dead-time was restricted to $0.1 \le \tau_o \le 1.2$ to obtain positive controller parameters. The obtained mean and [minimum, maximum] nominal robustness, M_S^o, values were 2.00 [1.99, 2.00] and 1.60 [1.60, 1.60], respectively.

Using conversion factors, the parameters of series (12.28) and parallel (12.29) PID controllers were found. The robustness obtained with these equivalent controllers was for the series PID: 1.97 [1.95, 1.98] and 1.61 [1.60, 1.62], which is very near to the target robustness, but the robustness was for the parallel PID: 2.09 [2.00, 2.51] and 1.61 [1.60, 1.72]. These differences between the target and the obtained nominal robustness with the equivalent series and parallel controllers were due to the influence of the controller derivative filter; in the robustness evaluation, the same derivative filter constants, $\alpha = \alpha_s = \alpha_p = 0.1$, were used. The control system robustness will be drastically reduced with the equivalent parallel PID controller, specially for lower τ_o.

To remove this unwanted side effect of the equivalent controllers, new sets of controller parameters were found for the series and parallel PIDs, using the B&J tuning joined with the conversion factors and controllers (12.28) and (12.29) to achieve the target robustness levels.

The nominal robustness and the nominal servo- and regulatory control performance, obtained with these new tunings, are shown in Fig. 12.15. It is important to remember that these are three different implementations of the PID control algorithm, standard, series and parallel, all tuned with the same rule but with three different sets of parameters that, as shown in this figure, all resulted in the same robustness and nearly the same performance for each target robustness level.

The robustness and performance fragility of the standard PID controller tuned with the B&J tuning method are shown in Fig. 12.16, corresponding to the series

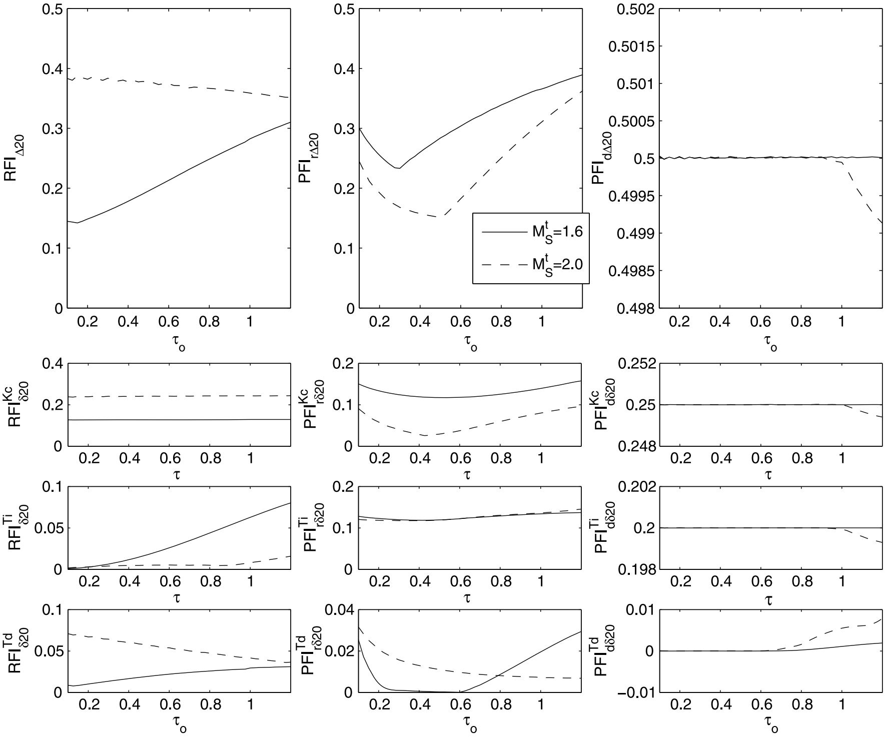


Fig. 12.16 B&J standard PID controller—robustness and performance fragility

PID in Fig. 12.17 and fragility indices for the parallel PID in Fig. 12.18. These figures will allow not only evaluating the tuning rule fragility but also the controller fragility due to its implementation.

Although the three controller implementations provide control systems with the same nominal robustness, and they all have in common that the low robustness controllers are more robustness-fragile than the high robustness ones, their robustness-fragility level and behavior are affected in different ways by the perturbation of the controller parameters and by the controlled process model normalized dead-time change.

It can be seen that the parallel implementation is less robustness-fragile, while the series is more robustness-fragile. It is also noted that the robustness-fragility of the parallel controller is more or less constant.

The controller gain parametric robustness-fragility, although it is the highest between the parametric-fragility indices, is nearly constant for all the controllers implementations, whereas the integral time parametric robustness-fragility increases with the model normalized dead-time, especially for the high robustness controllers.

With respect to the servo-control performance-fragility, it is the highest, but it is nearly constant in the series implementation.

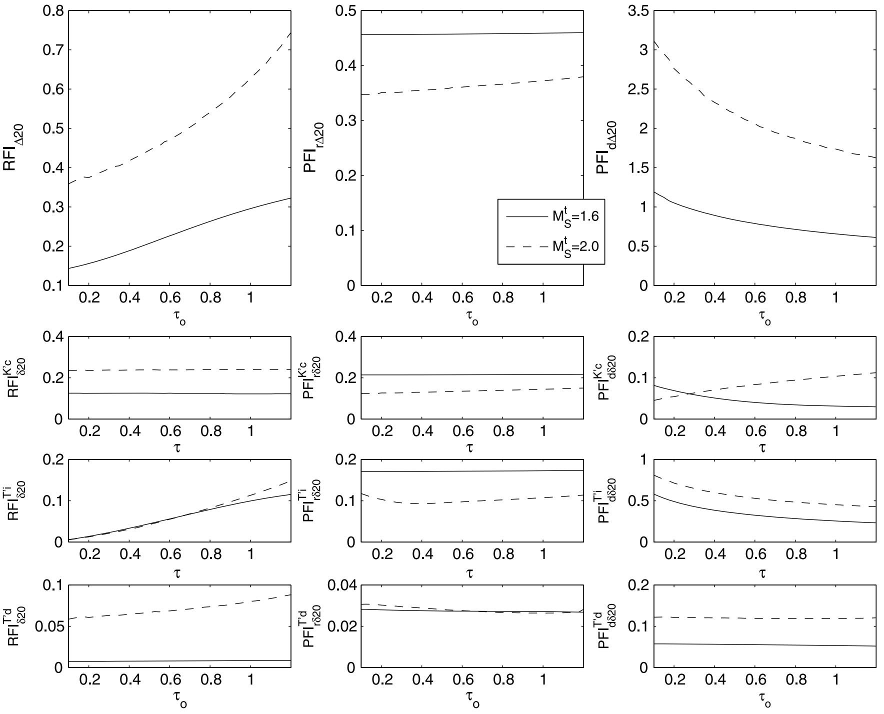


Fig. 12.17 B&J series PID controller—robustness and performance fragility

The regulatory control performance fragility of the standard implementation is completely flat, as are its gain and integral time constant parametric regulatory-control-fragility indices. Its regulatory performance-fragility is not affected by the derivative time constant.

The regulatory control performance of the series implementation of the PID control algorithm is extremely high and is very sensitive to the changes in the controller integral time constant.

In particular for the B&J tuning rule, if a predictable and flat behavior of the robustness- and performance-fragility indices is of interest, the parallel implementation of the PID control algorithm will be safer for performing the controller final fine-tuning.

12.7 Conclusions

Modern tuning methods for proportional integral derivative (PID) controllers must take into account the existing trade-offs between several conflicting design considerations: the *performance* with step changes in the set-point and load-disturbances, the *control effort* requirements, the control system *robustness* with the changing char-

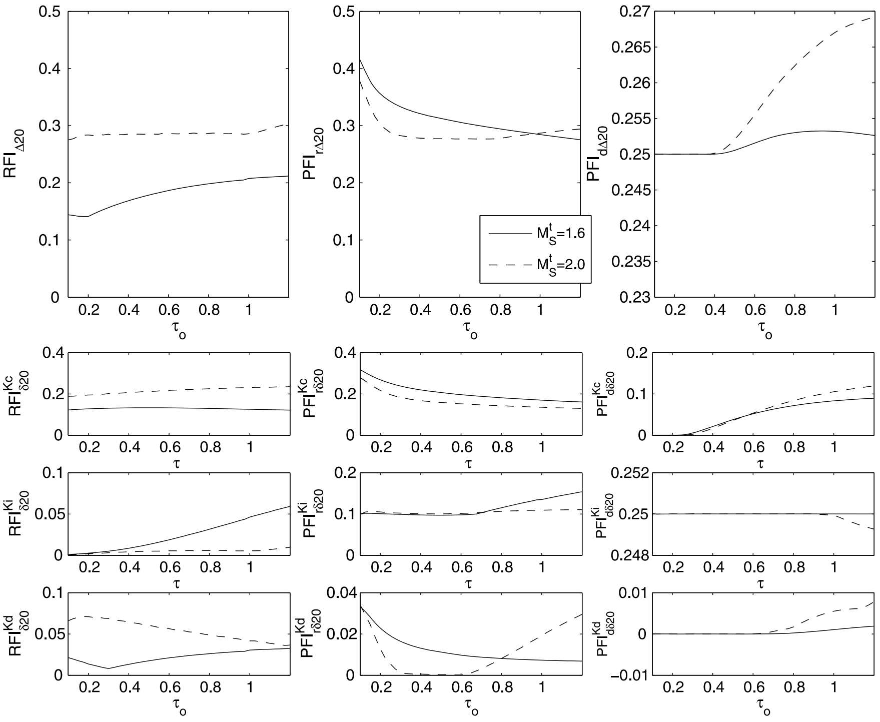


Fig. 12.18 B&J parallel PID controller—robustness and performance fragility

acteristics of the controlled process, and the controller robustness and performance *fragility* with the perturbation of its own parameters.

The fragility of a controller will depend not only on the tuning rule design considerations but also on the controller implementation, i.e., the PID control algorithm used.

The evaluation of the tuning rules presented in this chapter is a *worst-case analysis* for robustness and performance when all the controller parameters are perturbed by the same amount, a 20% change for our definition of the fragility indices. It may be expected that if there is a set of controller parameters that adversely affects the control system by reducing its robustness and/or performance, there should be another set of the controller parameters that may improve these characteristics of the closed-loop.

If the tuning rule can guarantee the control system design target robustness level, at least with the controlled process model used for tuning the controller, then it will be important to the control system designer that it be a robustness-non-fragile tuning rule. Preferably, it would be robustness-resilient, but not necessarily or even desirably that it also be servo- or regulatory performance-non-fragile, considering that, normally, the controller final fine-tuning has, as its main purpose, to modify in some way the transient response, and then the performance, of the control system.

A detailed analysis may be conducted in the controller parametric space to evaluate not only the control system reduction of the robustness and performance when they are perturbed but also the increment of these indicators. Using the perturbed controller extreme (maximum and minimum) robustness and performance measurements, the controller fragility may be redefined in such a way that the positive fragility indices will denote a loss of robustness or performance, and the negative fragility indices will denote an improvement in robustness or performance.

References

1. Afolabi, D.: Optimal controllers are fragile. In: American Control Conference, Denver, Colorado, USA, 4–6 June 2003
2. Alfaro, V.M.: PID controllers' fragility. ISA Trans. **46**, 555–559 (2007)
3. Alfaro, V.M., Vilanova, R., Arrieta, O.: Analytical robust tuning of PI controllers for first-order-plus-dead-time processes. In: 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA2008), Hamburg, Germany, 15–18 September 2008
4. Alfaro, V.M., Vilanova, R., Arrieta, O.: Fragility analysis of PID controllers. In: 18th IEEE Conference on Control Applications, Saint Petersburg, Russia, 8–10 July 2009
5. Alfaro, V.M., Vilanova, R., Arrieta, O.: NORT: a non-oscillatory robust tuning approach for 2-DoF PI controllers. In: 18th International Conference on Control Applications part of the 2009 IEEE Multi-Conference on Systems and Control (CCA2009), Saint Petersburg, Russia, 8–9 July 2009
6. Alfaro, V.M., Vilanova, R., Arrieta, O.: A single-parameter robust tuning approach for two-degree-of-freedom PID controllers. In: European Control Conference (ECC2009) Budapest, Hungary, 23–26 August 2009
7. Alfaro, V.M., Vilanova, R., Arrieta, O.: Maximum sensitivity based robust tuning for two-degree-of-freedom proportional-integral controllers. Ind. Eng. Chem. Res. **49**, 5415–5423 (2010)
8. Ali, A., Majhi, S.: PI/PID controller design based on IMC and percentage overshoot specification to controller setpoint change. ISA Trans. **48**, 10–15 (2009)
9. Araki, M.: On two-degree-of-freedom PID control system. Tech. rep., SICE Research Committee on Modeling and Control Design of Real Systems (1984)
10. Araki, M.: Two-degree-of-freedom control system—I. Syst. Control **29**, 649–656 (1985)
11. Åström, K.J., Hägglund, T.: Automatic tuning of simple regulators with specification on phase and amplitude margins. Automatica **20**(5), 645–651 (1984)
12. Åström, K.J., Hägglund, T.: PID Controllers: Theory, Design and Tuning. Instrument Society of America, Research Triangle Park (1995)
13. Åström, K.J., Hägglund, T.: Revisiting the Ziegler–Nichols step response method for PID control. J. Process Control **14**, 635–650 (2004)
14. Åström, K.J., Hägglund, T.: Advanced PID Control. ISA, Research Triangle Park (2006)
15. Åström, K.J., Murray, R.M.: Feedback Systems: An Introduction for Scientists and Engineers. Princeton University Press, Princeton (2008)
16. Åström, K.J., Hang, C.C., Persson, P., Ho, W.K.: Towards intelligent PID control. Automatica **28**(1), 1–9 (1992)
17. Åström, K.J., Panagopoulos, H., Hägglund, T.: Design of PI controllers based on non-convex optimization. Automatica **34**, 585–601 (1998)
18. Babb, M.: Pneumatic instruments gave birth to automatic control. Control Eng. **37**(12), 20–22 (1990)
19. Bennett, S.: The past of PID controllers. In: IFAC Digital Control: Past, Present and Future of PID Control, Terrassa, Spain (2000)

20. Brosilow, C., Joseph, B.: Techniques of Model-Based Control. Prentice Hall PTR, Upper Saddle River (2002)
21. Chien, I.-L., Fruehauf, P.S.: Consider IMC tuning to improve controller performance. Chem. Eng. Prog. **86**(10), 33–41 (1990)
22. Chien, I.L., Hrones, J.A., Reswick, J.B.: On the Automatic Control of generalized passive systems. Trans. ASME **74**, 175–185 (1952)
23. Cohen, G.H., Coon, G.A.: Theoretical considerations of retarded control. ASME Trans. **75**, 827–834 (1953)
24. Corripio, A.B.: Tuning of Industrial Control Systems, 2nd edn. ISA, Research Triangle Park (2001)
25. Datta, A., Ho, M.-T., Bhattacharyya, S.P.: Structure and Synthesis of PID Controllers. Springer, London (2000)
26. Dorato, P.: Non-fragile controller design: An overview. In: American Control Conference, Philadelphia, Pennsylvania, USA, June 1998
27. Fung, H.W., Wang, Q.G., Lee, T.H.: PI tuning in terms of gain and phase margin. Automatica **34**(9), 1145–1149 (1998)
28. Gerry, J.P., Hansen, P.D.: Choosing the right controller. Chem. Eng. **25**, 65–68 (1987)
29. Goodwin, G.C., Graebe, S.F., Salgado, M.E.: Control System Design. Prentice Hall, Upper Saddle River (2001)
30. Gorez, R.: New design relations for 2-DOF PID-like control systems. Automatica **39**, 901–908 (2003)
31. Hägglund, T., Åström, K.J.: Revisiting the Ziegler–Nichols tuning rules for PI control. Asian J. Control **4**, 354–380 (2002)
32. Hang, C.C., Cao, L.S.: Improvement of transient response by means of variable set point weighting. IEEE Trans. Ind. Electron. **4**, 477–484 (1996)
33. Herreros, A., Baeyens, E., Perán, J.R.: Design of PID-type controllers using multiobjective genetic algorithms. ISA Trans. **41**, 457–472 (2002)
34. Ho, M.T.: Non-fragile PID controller design. In: 39th IEEE Conference on Decision and Control Sydney, Australia, December 2000
35. Ho, W.K., Hang, C.C., Cao, L.S.: Tuning of PID controllers based on gain and phase margin specifications. Automatica **31**(3), 497–502 (1995)
36. Ho, W.K., Lim, K.W., Xu, W.: Optimal gain and phase margin tuning for PID controllers. Automatica **31**(8), 10091014 (1998)
37. Ho, M.T., Datta, A., Bhattacharyya, S.P.: Robust and non-fragile PID controller design. Int. J. Robust Nonlinear Control **11**, 681–708 (2001)
38. Huang, H.P., Jeng, J.C.: Monitoring and assessment of control performance for single loop systems. Ind. Eng. Chem. Res. **41**, 1297–1309 (2002)
39. Isaksson, A.J., Graebe, S.F.: Derivative filter is an integral part of PID design. IEE Proc., Control Theory Appl. **149**(1), 41–45 (2002)
40. Istepanian, R.S.H., Whidborne, J.F.: Finite-precision computing for digital control systems: current status and future paradigms. In: Istepanian, R.S.H., Whidborne, J.F. (eds.) Digital Controller Implementation and Fragility. A Modern Perspective, pp. 1–10. Springer, London (2001)
41. Istepanian, R.S.H., Wu, J., Chu, J., Whidborne, J.F.: Maximizing lower bond stability measure of finite precision PID controller realizations by nonlinear programming. In: American Control Conference Philadelphia, Pennsylvania, USA, June 1998
42. Johnson, M.A.: PID control technology. In: Johnson, M.A., Moradi, M.H. (eds.) PID Control—New Identification and Design Methods, pp. 1–46. Springer, London (2005)
43. Kaesbauer, D., Ackermann, J.: How to escape from the fragility trap. In: American Control Conference (1998)
44. Kaya, I.: Tuning PI controllers for stable process with specifications on gain and phase margins. ISA Trans. **43**, 297–304 (2004)
45. Keel, L.H., Battacharyya, S.P.: Robust, fragile or optimal? IEEE Trans. Autom. Control **42**, 1098–1105 (1997)

46. Kristiansson, B.: PID controllers, design and evaluation. Ph.D. thesis, Control and Automation Laboratory, Department of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden (2003)
47. Lee, C.H.: A survey of PID controller design based on gain and phase margins. Int. J. Comput. Cogn. **2**(3), 63–100 (2004)
48. López, A.M., Miller, J.A., Smith, C.L., Murrill, P.W.: Tuning controllers with error-integral criteria. Instrum. Technol. **14**, 57–62 (1967)
49. Mäkilä, P.M.: Comments on "Robust fragile, or optimal". IEEE Trans. Autom. Control **43**, 1265–1267 (1998)
50. Martin, J., Corripio, A.B., Smith, C.L.: Controller tuning from simple process models. Instrum. Technol. **22**(12), 39–44 (1975)
51. Mian, A.A., Wang, D., Ahmand, M.I.: Optimal non-fragile controller realization algorithm and application to control problems. In: Chinese Control and Decision Conference (CCDC2008), Yantai, China, 2–4 July, pp. 5331–5336 (2008)
52. Mian, A.A., Mian, I.A., Wang, D.: Controller fragility minimization algorithm and comparison with different controller realizations. In: Smart Manufacturing Application, 9–11 April, pp. 157–162. Kintex, Gyeonggi-do (2008)
53. Morari, M., Zafiriou, E.: Robust Process Control. Prentice-Hall, Englewood Cliffs (1989)
54. O'Dwyer, A.: Handbook of PI and PID Controller Tuning Rules, 2nd edn. Imperial College Press, London (2006)
55. Paattilammi, J., Mäkilä, P.M.: Fragility and robustness: A case study on paper machine headbox control. IEEE Control Syst. Mag. **20**(1), 13–22 (2000)
56. Panagopoulos, H.: PID control design, extension, application. Ph.D. thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden (2000)
57. Rivera, D.E., Morari, M., Skogestad, S.: Internal model control. 4. PID controller design. Ind. Eng. Chem. Process Des. Dev. **25**, 252–265 (1986)
58. Rovira, A., Murrill, P.W., Smith, C.L.: Tuning controllers for setpoint changes. Instrum. Control Syst. **42**, 67–69 (1969)
59. Shinskey, F.G.: Process control: As taught vs as practiced. Ind. Eng. Chem. Res. **41**, 3745–3750 (2002)
60. Silva, G.J., Datta, A., Bhattacharyya, S.P.: PID tuning revisited: Guaranteed stability and non-fragility. In: American Control Conference Anchorage, Arkansas, USA, 8–10 May (2002)
61. Silva, G.J., Datta, A., Bhattacharyya, S.P.: On the stability and controller robustness of some popular PID tuning rules. IEEE Trans. Autom. Control **48**(9), 1638–1641 (2003)
62. Silva, G.J., Datta, A., Bhattacharyya, S.P.: PID Controllers for Time-Delay Systems. Birkhäuser, Boston (2005)
63. Skogestad, S.: Simple analytic rules for model reduction and PID controller tuning. J. Process Control **13**, 291–309 (2003)
64. Taguchi, H., Araki, M.: Two-degree-of-freedom PID controllers—their functions and optimal tuning. In: IFAC Digital Control: Past, Present and Future of PID Control, Terrassa, Spain (2000)
65. Tavakoli, S., Tavakoli, M.: Optimal tuning of PID controllers for first order plus time delay models using dimensional analysis. In: The Fourth International Conference on Control and Automation (ICCA'03), Montreal, Canada, 10–13 June 2003
66. Tavakoli, S., Griffin, I., Fleming, P.J.: Robust PI controller for load disturbance rejection and setpoint regulation. In: IEEE Conference on Control Applications, Toronto, Canada, 28–31 August 2005
67. Tavakoli, S., Griffin, I., Fleming, P.J.: Multi-objective optimization approach to the PI tuning problem. In: IEEE Congress on Evolutionary Computing (CEC2007), pp. 3165–3171 (2007)
68. Visioli, A.: Practical PID Control. Springer, London (2006)
69. Whidborne, J.F.: Controller fragility. In: 25th Nathiagali Summer School on Physics and Contemporary Needs, Nathiagali, Pakistan, July 2000
70. Whidborne, J.F., Istepanian, R.S.H., Wu, J.: Reduction of controller fragility by pole sensitivity minimization. IEEE Trans. Autom. Control **46**(2), 320–325 (2001)

71. Xu, H.: Synthesis and design of PID controllers. Ph.D. thesis, Texas A&M University (2004)
72. Yaniv, O., Nagurka, M.: Design of PID controllers satisfying gain margin and sensitivity constrains on a set of plants. Automatica **40**, 111–116 (2004)
73. Ziegler, J.G., Nichols, N.B.: Optimum settings for automatic controllers. Trans. Am. Soc. Mech. Eng. **64**, 759–768 (1942)

Chapter 13
Benchmarking and Tuning PID Controllers

Andrew W. Ordys and Michael J. Grimble

13.1 Introduction—Setting the Scene

While considering applications of closed-loop control systems, two main tasks can be distinguished, namely regulatory control and tracking control. Regulatory control is the task of maintaining the value of the output of systems at a constant level, equal to the set-point value of the process. The tracking control refers to the situation where the desired value of the output is changing, following a changing reference signal.

Obviously, regulatory control can be considered a special case of tracking control, when speed of changes of the reference signal equals zero. Therefore the distinction may seem superficial. Nevertheless, different behaviour, performance measures and tuning methods are used for the two situations.

In regulatory control, the main task is to maintain the value of the output at a constant level. This is achieved by counteracting the effect of disturbances. Therefore, the main measure of the controller performance will be the variance of the output signal, which can be found by averaging over time (with some assumptions and approximations). Sometimes the variance of the input signal is also measured, as it signifies the amount of energy used to control the system. The tuning of the controller can then be related to the nature of the disturbance, for instance its frequency spectrum.

In tracking control, the main task is to follow the trajectory specified by the reference signal. This is achieved by a good sensitivity function response and fast

A.W. Ordys (✉)
School of Mechanical and Automotive Engineering, Faculty of Science, Engineering and Computing, Kingston University London, Friars Avenue, Roehampton Vale, London SW15 3DW, UK

M.J. Grimble
Industrial Control Centre, Department of Electronic and Electrical Engineering, University of Strathclyde, 204 George Street, Glasgow G1 1XW, UK

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_13, © Springer-Verlag London Limited 2012

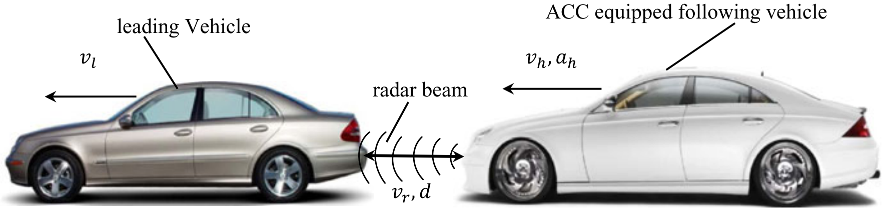


Fig. 13.1 Adaptive cruise control system for maintaining a safe breaking distance between vehicles

reaction of the controller. The main measure of the controller performance is the difference between the output and the reference signal (usually squared) over a finite time horizon. If the changes in the reference signal, compared to the disturbances, are large, the stochastic nature of these signals may be considered less important than the evaluation of their deterministic parts. The tuning of the controller can be related to the nature of the reference signal and the speed of its changes.

To illustrate the two groups of tasks, let us consider a control system: adaptive cruise control, which is becoming very popular in the automotive industry (Fig. 13.1).

The system has two modes of operation [33, 34]:

Cruise control mode In this case, a PI(D) controller maintains the speed of the vehicle constant equal to a "desired speed" (or set point), determined by the driver. It operates in this mode when the road in front of the ACC equipped vehicle is clear, i.e. there is no vehicle within clearance distance.

Constant distance mode When the controlled vehicle reaches a certain distance to the vehicle in front (so-called safe breaking distance), the controller will switch from constant speed mode to constant distance mode and will maintain the distance to the vehicle in front constant.

The Adaptive Cruise Controller consist of two loops. The inner loop (CC) is the speed control loop (Fig. 13.2), whose task is to maintaining the speed of the vehicle equal to the desired speed. The outer loop (ACC) decides on the mode of operation (either constant speed or constant distance) and hence determines the reference speed submitted to the inner loop. This reference speed will be either a constant "cruise control" speed or will be changing as required by the "constant distance" condition.

The distance control loop (ACC) is shown in Fig. 13.3.

Figure 13.4 [33] shows a possible scenario of changes in the distance to the vehicle in front as a function of time. Initially, the distance is large, and therefore the task of the controller is to reach the "safe breaking distance" as quickly as possible, with a constraint on the maximum vehicle velocity. This corresponds to the tracking task (disturbance rejection mode) of the controller. Once the "safe breaking distance" has been reached, at approximately 18 seconds, the task of the controller is to maintain this distance in spite of disturbances (e.g. variations of the speed of the vehicle in

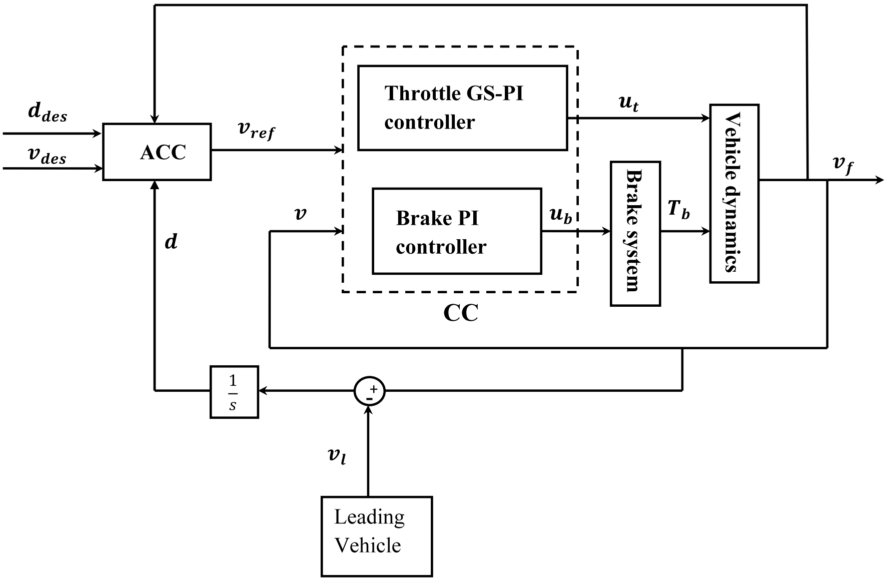


Fig. 13.2 Speed control for adaptive cruise control system

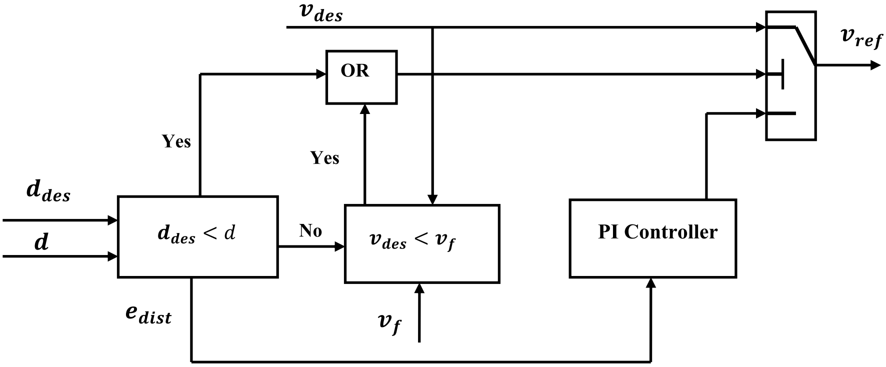


Fig. 13.3 PI controller for distance control function

front will cause changes in the distance). This corresponds to the regulatory task of the controller.

At 60 seconds, the vehicle in front changes its speed, which causes a change to the “safe braking distance”. The control system has to respond to the change in its set-point. This is also a tracking task. Finally, at 80 seconds, the vehicle in front accelerates to a speed higher that the cruise speed of the controlled vehicle, hence the distance controller is switched off.

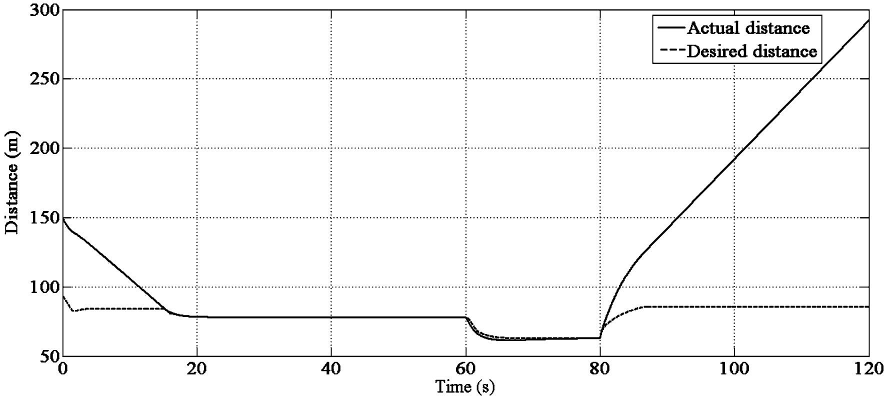


Fig. 13.4 Distance between vehicles as a function of time

13.2 Performance Assessment for Regulatory Tasks

The assessment of controller performance for regulatory tasks has long been recognised as an important issue. The use of benchmark criteria for control loop performance assessment was investigated by Desborough and Harris [8–10]. The control loop performance benchmarking techniques have built on ideas used successfully in business process benchmarking [1, 3, 7, 30]. The control loop tuning methods have mainly been developed with the process industries in mind, but benchmarking methods have been applied in other industrial sectors [4, 5].

The aim of a benchmarking technique is to diagnose control loop performance [1, 3] by providing tools to determine:

(i) The cost index and associated cost terms that may be used as a benchmark taking into considerations of restrictions on the controller structure.
(ii) The best achievable performance in terms of the benchmark and restrictions on controller structure.
(iii) The shortfall in performance that the current or nominal control loop provides.
(iv) The ways in which the loop tuning may be improved.

There has been considerable activity in the development of performance benchmarking tools based upon minimum variance criteria, recorded in the monograph by Huang and Shah [21]. Early work by Harris [19] showed how time series analysis techniques can be used to find an expression for the minimum possible cost from plant operating data.

The main difficulties with this approach are that the Minimum-Variance (MV) control law has a high gain, wide bandwidth, unrealistically large control signal variations and it normally requires a high-order controller (a controller order equal to that of the system). However, it has a value as a benchmark cost, since economic performance can often be related to the variance of the error. Unfortunately many real systems could not use a controller obtained from such a technique, because

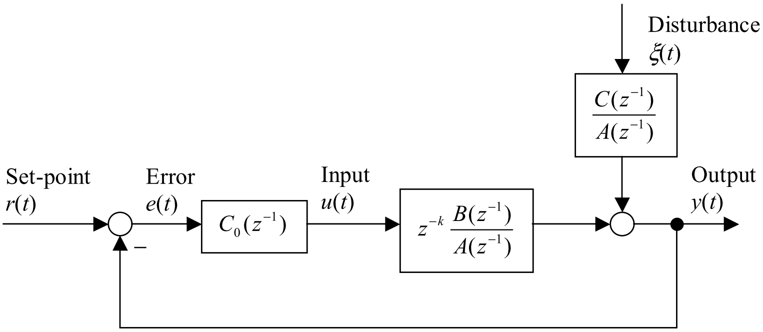


Fig. 13.5 Generic stochastic feedback control system structure

of the violence of the control signal changes and because a high-order controller cannot be used in practice.

A so-called *Generalised Minimum Variance* (*GMV*) controller has many features in common with the minimum-variance criterion but has a control signal costing facility [6]. This means that the control activity can be limited and a more practical criterion is obtained as discussed in [11]. Under certain assumptions on dynamic cost-function weighting choice, the control law has guaranteed stability, and the benchmarking results are almost as simple as those for the minimum-variance control case.

In the following sections, the *MV* and *GMV* benchmarking will be introduced briefly, before the attention moves to restricted structure (*PID*) benchmarking.

13.2.1 Minimum-Variance and Generalised Minimum-Variance Control

Consider the generic scalar control system shown in Fig. 13.5.

From Fig. 13.5, the relationship between the output and the inputs to the plant is

$$A(z^{-1})y(t) = z^{-k}B(z^{-1})u(t) + C(z^{-1})\xi(t) \tag{13.1}$$

where $y(t)$ represents variation of the output signal around a given steady-state operating point, $u(t)$ is the control signal, $\xi(t)$ denotes a disturbance which is assumed to be zero mean Gaussian white noise of variance σ^2, $A(z^{-1})$, $B(z^{-1})$, $C(z^{-1})$ are polynomials in the backward shift operator z^{-1}, and the roots of $C(z^{-1})$ and $B(z^{-1})$ are assumed to lie within the unit circle, i.e.:

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-1} + \cdots + a_n z^{-n_A}, \tag{13.2}$$

$$B(z^{-1}) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n_B} \quad b_0 \neq 0, \tag{13.3}$$

$$C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-1} + \cdots + c_n z^{-n_C}, \tag{13.4}$$

$$z^{-1} \cdot x(t) = x(t-1). \tag{13.5}$$

Thus z^{-k} in (13.23) represents a k-step delay in the control signal. This means that the control signal starts to act on the system after k time increments. The constant set-point $r(t)$ is for simplicity assumed to be zero. The plant is controlled by a linear feedback controller $C_0(z^{-1})$:

$$u(t) = -C_0\bigl(z^{-1}\bigr)y(t). \tag{13.6}$$

Let k equal the plant time delay of length k.

In *Minimum-Variance Control*, the cost index to be minimised is the variance of the output at time $t+k$ given all the information up to time t:

$$J(t) = E\bigl\{y(t+k)^2 \mid t\bigr\} \tag{13.7}$$

where $E\{.\}$ denotes the expectation operator.

The following polynomial identity can be defined to effectively split the disturbance term involving the polynomial C into two time frame components:

$$C = AF + z^{-k}G \tag{13.8}$$

where F and G are polynomials defined as

$$F\bigl(z^{-1}\bigr) = 1 + \sum_{i=1}^{k-1} f_i z^{-1}, \qquad g\bigl(z^{-1}\bigr) = g_0 + \sum_{i=1}^{n_G} g_i z^{-i} \tag{13.9}$$

with $n_G = \max(n_C - k, n_A - 1)$.

It may be assumed that the control signal must be a function of information available at time instant t, i.e. all past control signals, all past outputs and the present output:

$$Y(t) = \bigl[u(t-k-1) \quad u(t-k-2) \quad \dots \quad y(t) \quad y(t-1) \quad \dots\bigr]. \tag{13.10}$$

The variance of the output may be obtained as

$$\sigma_y^2 = E\bigl\{y^2(t+k)\bigr\} = E\left\{\left(\frac{BF}{C}u(t) + \frac{G}{C}y(t)\right)^2\right] + E\bigl[\bigl(F\xi(t+k)\bigr)^2\bigr\}. \tag{13.11}$$

Thence let

$$J_{\min} E\bigl\{\bigl(F\xi(t+k)\bigr)^2\bigr\} \quad \text{and} \quad J_0 = E\left\{\left(\frac{BF}{C}u(t) + \frac{G}{C}y(t)\right)^2\right\}, \tag{13.12}$$

$$J = \sigma_y^2 = J_{\min} + J_0, \tag{13.13}$$

where the variance $J_{\min}$ is independent of the control action, and the variance J_0 is dependent on the process control law. To achieve the minimum variance of the output variable, from (13.12) set:

$$\frac{BF}{C}u(t) + \frac{G}{C}y(t) = 0, \tag{13.14}$$

$$u(t) = -\frac{G}{BF}y(t). \tag{13.15}$$

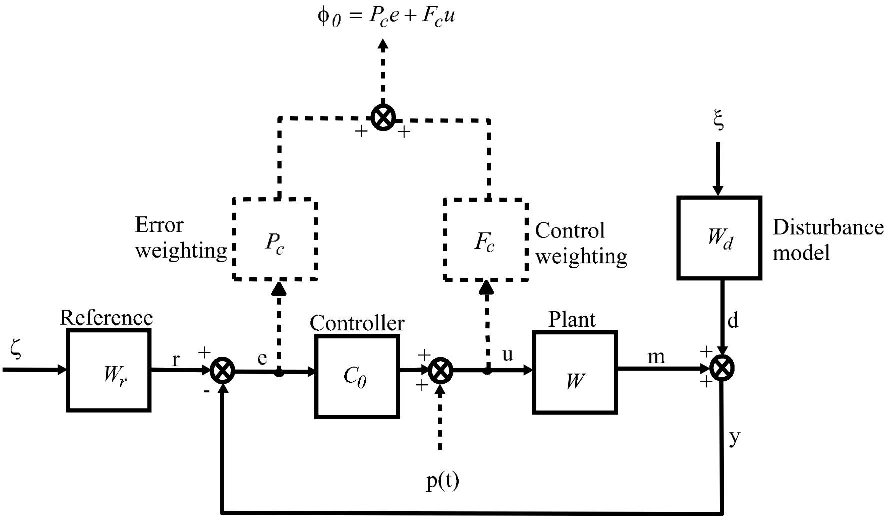


Fig. 13.6 Generalised Minimum Variance—generalised output signal

The minimum-variance control may be viewed as the stochastic equivalent of dead-beat control. It usually generates large input signal variations and is sensitive to process-model mismatch. It clearly cannot be used in this form for non-minimum phase processes.

The *Generalised Minimum-Variance* (*GMV*) control algorithm minimises the variance of a 'generalised' output, which is defined as a weighted combination of the process variable (or control error) and the control signal (see Fig. 13.6). By analogy with the cost-function employed in the original *GMV* control problem [6], let the inferred output signal to be minimised have the weighted error and control signal form:

$$\varphi_0(t) = P_c e(t) + F_c u(t). \tag{13.16}$$

The following weighted error and control signal variance is to be minimised:

$$J = E\{(P_c e(t) + F_c u(t))^2\} \tag{13.17}$$

where the dynamic cost-function weighting terms are defined in the polynomial form:

$$\begin{aligned} P_c &= P_{cd}^{-1} P_{cn}, \quad P_{cd}(0) = 1 \text{ and } P_{cn}(0) \neq 0, \\ F_c &= F_{cd}^{-1} F_{cn}, \quad F_{cd}(0) = 1 \text{ and } F_{cn} = F_{ck} z^{-k} \end{aligned} \tag{13.18}$$

where $F_{ck} = F_{n0} + F_{n1} z^{-1} + \cdots + F_{nn} z^{-n}$. The full-order GMV optimal control law, when there is no limitation on the controller structure and order, may be computed as

$$C_0 = (H_0 P_{cd})^{-1} G_0 F_{cd} \tag{13.19}$$

where (H_0, G_0) is the unique smallest-degree solution, with respect to the polynomial: $F_0 = f_0 + f_1 z^{-1} + \cdots + f_{nf} z^{-n_f}$ (where $n_f \leq k-1$) of the diophantine equations

$$P_{cd} A F_0 + z^{-k} G_0 = P_{cn} C, \tag{13.20}$$

$$F_{cd} B F_0 - z^{-k} H_0 = F_{cn} C. \tag{13.21}$$

13.2.2 MV and GMV Benchmarking

For practical systems, the controller performance does not match that of a minimum-variance controller [13]. Therefore,

$$\frac{BF}{C} u(t) + \frac{G}{C} y(t) \neq 0. \tag{13.22}$$

Hence, in the expression defining the future output $y(t+k)$, there is an extra term whose variance is nonzero:

$$y(t+k) = F\xi(t+k) + \hat{y}(t). \tag{13.23}$$

Note that there is no correlation between $\hat{y}(t)$ and $F\xi(t+k)$, and the variance of the output under non-minimum variance control can be computed as

$$\sigma_y^2 = E\{(y(t+k))^2\} = E\{(F\xi(t+k))^2\} + E\{(\hat{y}(t))^2\}, \tag{13.24}$$

$$\sigma_y^2 = \sigma_{\text{soc}}^2 + \sigma_{\text{mvc}}^2 \tag{13.25}$$

where the subscript soc is the short term for suboptimal control and mvc for minimum variance control.

13.2.2.1 MV and GMV Controller Performance Index

The *controller performance index* (*CPI*) is defined to be the ratio of the minimum variance of the signal to the actual variance. Define the controller performance index (*CPI*) as

$$\eta(k) = \frac{\sigma_{\text{mvc}}^2}{\sigma_{\text{soc}}^2 + \sigma_{\text{mvc}}^2} \in [0, 1]. \tag{13.26}$$

$\eta(k) = 1$ indicates the ideal case of minimum variance control, and $\eta(k) \to 0$ the case of the very poor control, when the variance of the output tends to infinity.

The estimation of the *GMV* performance index is performed in much the same way as in the minimum variance case. The minimum cost is equivalent to that for the minimum variance control, but now it refers to $\varphi_0(t)$ rather than to $y(t)$.

The value of the Controller Performance Index, $\eta(k)$, defined above can be estimated from routine closed-loop process data using a linear regression method.

The process output under feedback control is given by

$$y(t) = F\xi(t) + z^{-k}\left(\frac{G - BFC_0}{C}\right)y(t), \tag{13.27}$$

$$y(t) = F\xi(t) + \sum_{i=1}^{m} \alpha_i y(t - k - i + 1). \tag{13.28}$$

Running t over a range of values and stacking up similar terms yields

$$y = X\alpha + F\xi \tag{13.29}$$

with

$$\begin{aligned} y &= \begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_{k+m} \end{bmatrix}, \qquad X = \begin{bmatrix} y_n & y_{n-k-1} & \cdots & y_{n-k-m+1} \\ y_{n-k-1} & y_{n-k-2} & \cdots & y_{n-k-m} \\ \vdots & \vdots & \ddots & \vdots \\ y_m & y_{m-1} & \cdots & y_1 \end{bmatrix}, \\ \alpha &= \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix}. \end{aligned} \tag{13.30}$$

Linear regression is used to estimate autoregressive the parameters $\{\alpha_i\}$:

$$\hat{\alpha} = \left(X^{\mathrm{T}}X\right)^{-1}X^{\mathrm{T}}y. \tag{13.31}$$

The estimate of the minimum variance can be calculated as the residual variance:

$$\frac{1}{n - k - 2m + 1}(y - X\alpha)^{\mathrm{T}}(y - X\alpha) = \sigma^2_{\mathrm{mvc}} = \sum_{i=0}^{k-1} f_i^2\sigma^2 \tag{13.32}$$

whilst the actual variance follows as

$$\frac{1}{n - k - m + 1}y^{\mathrm{T}}y = \sigma_y^2 = \sigma^2_{\mathrm{soc}} + \sigma^2_{\mathrm{mvc}}. \tag{13.33}$$

Hence, the least squares estimate for the controller performance index $\eta(k)$ can be written as

$$\hat{\eta}(k) = \frac{(n - k - m + 1)}{(n - k - 2m + 1)} \cdot \frac{(y - X\alpha)^{\mathrm{T}}(y - X\alpha)}{y^{\mathrm{T}}y + (n - k - m + 1)\bar{y}^2} \tag{13.34}$$

where the mean square error was used rather than the variance, hence penalising nonzero steady-state errors. A more detailed derivation of *MV* and *GMV* benchmarks can be found in [27].

The value of *MV* or *GMV* performance index can also be calculated using spectral factorisation of signals. Following [11], the minimum value for the variance of $\varphi_0(t)$ and the cost associated with the individual weighted and unweighted terms in the cost-function are given as:

$$J_{\min} E\{\varphi_{0\min}(t)^2\} = \frac{1}{2\pi j}\oint_{|z|=1}\{F_0^* F_0\}\frac{dz}{z}, \tag{13.35}$$

$$J_{e_{\min}} E\{e_{\min}(t)^2\} = \frac{1}{2\pi j}\oint_{|z|=1}\left\{\left(\frac{P_{cd}H_0}{D_c U_0}\right)^*\left(\frac{P_{cd}H_0}{D_c U_0}\right)\right\}\frac{dz}{z}, \tag{13.36}$$

$$J_{u_{\min}} = E\{u_{\min}(t)^2\} = \frac{1}{2\pi j}\oint_{|z|=1}\left\{\left(\frac{F_{cd}G_0}{D_c U_0}\right)^*\left(\frac{F_{cd}G_0}{D_c U_0}\right)\right\}\frac{dz}{z}, \tag{13.37}$$

$$J_{p_{\min}} = E\{(P_c e_{\min}(t))^2\} = \frac{1}{2\pi j}\oint_{|z|=1}\left\{\left(\frac{P_{cn}H_0}{D_c U_0}\right)^*\left(\frac{P_{cn}H_0}{D_c U_0}\right)\right\}\frac{dz}{z}, \tag{13.38}$$

$$J_{f_{\min}} = E\{(F_c u_{\min}(t))^2\} = \frac{1}{2\pi j}\oint_{|z|=1}\left\{\left(\frac{F_{cn}G_0}{D_c U_0}\right)^*\left(\frac{F_{cn}G_0}{D_c U_0}\right)\right\}\frac{dz}{z}. \tag{13.39}$$

If the controller $C_0 = C_{0n}C_{0d}^{-1}$ is non-optimal (reduced order, restricted structure or possibly a classical design), the minimum cost (13.35) increases and becomes

$$J = E\{\varphi_0(t)^2\} = J_0 + J_{\min} \tag{13.40}$$

where

$$J_0 = \frac{1}{2\pi j}\oint_{|z|=1}\{T_0^* T_0\}\frac{dz}{z} \quad \text{and} \quad T_0 = \frac{C_{0d}(G_0 F_{cd}) - C_{0n}(H_0 P_{cd})}{(AC_{0d} + BC_{0n})(P_{cd}F_{cd})}.$$

13.2.3 Restricted Structure (PID) Controller Benchmarking

If the controller structure is restricted, or if the controller order is limited, additional complication is introduced into the computational procedure, but the results are easier to implement. Such a controller provides a benchmark cost figure for a system with controller of pre-specified structure. This enables the best performance, in terms of the criterion, to be determined and the optimum controller coefficients for the given controller structure to be determined. The restricted structure *GMV* controller which is proposed in the following is simpler than the alternative restricted structure *LQG* designs [17].

13.2.3.1 Benchmark Cost-Function Weighting Definitions

From the assumed choice of weightings the polynomial D_c must be strictly-Schur, where

$$D_c = P_{cn}F_{cd}B_{0k} - F_{ck}P_{cd}A_0.$$

To consider the physical significance of D_c, consider the term

$$(P_{cd}F_{cd})^{-1}D_cA_0^{-1} = \frac{P_{cn}}{P_{cd}}\frac{B_{0k}}{A_0} - \frac{F_{ck}}{F_{cd}} = z^k(P_cW - F_c)$$

where the weightings can be written as $P_c = z^{-k}P_{ck}$ and $F_c = z^{-k}F_{ck}$ so that

$$(P_{cd}F_{cd})^{-1}D_cA_0^{-1} = P_{ck}W - F_{ck}.$$

Thus the term D_c is the numerator in the operator $(P_cW - F_c)$. This is simply the transfer between the control signal and the combined signal φ_0 in Fig. 13.6. The above strictly Schur assumption therefore requires the weightings P_c and F_c to be chosen so that $P_cW - F_c$ is minimum phase. This is related, but not the same, as the simplification which arises in *minimum-variance* controllers when the plant is assumed minimum-phase.

The weightings P_c and F_c are usually also chosen to be an integrator and a high-pass term, respectively. If the plant W is minimum-phase, it will be easy to satisfy the minimum phase requirement if $|F_c|$ is small enough. If the plant is open-loop stable it will be simple to satisfy the minimum-phase requirements if $|F_c|$ is large enough. However, for say energy benchmarking purposes, the weightings must also be chosen to ensure performance requirements are met.

13.2.3.2 Weighting Choice for Process Control Applications

Many process control problems involve a plant model, which may be approximated by a transport-delay and open-loop stable, minimum-phase plant. In this case it is easy to find weightings P_c and F_c which ensure that D_c is minimum phase. Clearly this is guaranteed when either P_c is large and F_c small or vice versa. For energy benchmarking applications, the focus may therefore be on getting appropriate weightings to satisfy the energy balance requirements, consistent with the need to satisfy the stability requirement. This suggests that there is sometimes a need to compromise if this approach is to be used.

13.2.3.3 GMV Controller Performance Index

The *controller performance index* (*CPI*) in this GMV problem is defined as the ratio of the minimum variance of the signal $\{\varphi_0(t)\}$ to the actual variance. The CPI can be obtained from (13.26) as

$$\kappa = J_{\min}/(J_{\min} + J_0) = 1 - J_0/(J_{\min} + J_0).$$

Clearly $\kappa \leq 1$, and $\kappa = J_{\min}/J$ is the ratio of the best possible performance to the actual performance. If κ is close to unity, the system provides little opportunity for improvement. If the CPI indicator κ is close to zero, the system will have poor regulatory action, and retuning is recommended.

The above scalar is the same idea as that introduced by Desborough and Harris [8]. The definition is slightly different, since for good control, $\kappa \to 1$. For poor control, $\kappa \to 0$, and this is the reciprocal to the index they introduced. This is thought to be more logical for some application areas. However, recorded data can still be used (as in their method) to estimate κ from the ratio of the variance of the residuals to the mean square variance of the output. This is discussed in more detail below.

The minimum variance of the signal $\varphi_0 = P_c e + F_c u$ follows from (13.12) as:

$$J_{\min} = E\left\{\varphi_{0\min}^2(t)\right\} = f_0^2 + f_1^2 + \cdots + f_{k-1}^2.$$

It follows that the theoretically achievable lower bound on the variance of the weighted output can be estimated from a representative set of either open- or closed-loop operating data [23]. A time-series model may be fitted to the data and the variance evaluated from the k leading terms. The approach by Desborough and Harris [10] is to fit an approximate model to the observed data:

$$\varphi_0(t) - \mu_\varphi = F_0\varepsilon(t) + \sum_{i=1}^{m} \alpha_i \left(\varphi(t-k-i+1) - \mu_\varphi\right) \tag{13.41}$$

where μ_φ represents the mean of the signal $\varphi_0(t)$. The infinite series is approximated to m terms, and the autoregressive parameters $\{\alpha_i\}$ are estimated by least-squares. The data may be written in vector matrix form as

$$\underline{\tilde{\varphi}} = \underline{X\alpha} + \underline{\varepsilon} \tag{13.42}$$

and the vector of autoregressive parameters is then found as

$$\underline{\alpha} = \left(\underline{X}^{\mathrm{T}}\underline{X}\right)^{-1}\underline{X}^{\mathrm{T}}\underline{\tilde{\varphi}}.$$

The *controller performance index*, defined above for a large data set, can then be approximated, assuming the mean is zero, as

$$\kappa = \left(\underline{\tilde{\varphi}} - \underline{X\alpha}\right)^{\mathrm{T}}\left(\underline{\tilde{\varphi}} - \underline{X\alpha}\right)/\left(\underline{\tilde{\varphi}}^{\mathrm{T}}\underline{\tilde{\varphi}}\right). \tag{13.43}$$

This approach has been very successful based on the minimum-variance criterion, and, for example, the application of control loop performance assessment methods to a Refinery was considered by Thornhill et al. [36]. The CPI is of course relating the sub-optimal or classical control to the *best* full-order GMV control solution. It is equally valuable to compare the performance of the best restricted structure controller (defined in the next section), to the full-order optimal and sub-optimal solutions.

13.2.3.4 Restricted Structure Optimal Control and Benchmarks

There now follows a derivation for the restricted structure optimal control solution [17] and for the desired expressions for the minimum cost. The so-called implied diophantine equation is first required, obtained from (13.20) and (13.21) as

$$(P_{cn}D_f)F_{cd}B_0 - (F_{cn}D_f)P_{cd}A_0 = z^{-k}(G_0F_{cd}B_0 + H_0P_{cd}A_0).$$

The *implied equation* now follows using (13.18):

$$G_0F_{cd}B_0 + H_0P_{cd}A_0 = (P_{cn}F_{cd}B_{0k} - F_{ck}P_{cd}A_0)D_f = D_cD_f. \tag{13.44}$$

There now follows an expression for the signal $\{\varphi_0(t)\}$, and it is important to note that this is valid for any choice of controller $C_0 = C_{0n}C_{0d}^{-1}$. If the combined reference and disturbance signals are represented by the spectral factor with input a white noise innovations signal, then introducing the sensitivity and control sensitivity, we obtain:

$$\varphi_0 = P_ce + F_cu = (P_c + F_cC_0)SY_f\varepsilon = \frac{(F_{cd}P_{cn}C_{0d} + P_{cd}F_{cn}C_{0n})}{(AC_{0d} + BC_{0n})P_{cd}F_{cd}}D_f\varepsilon.$$

This equation can be expanded to obtain a term that is independent of the choice of control action:

$$\begin{aligned}\varphi_0 &= \frac{[(AC_{0d} + BC_{0n})F_{cd}P_{cn}/A + P_{cd}F_{cn}C_{0n} - BC_{0n}F_{cd}P_{cn}/A]D_f\varepsilon}{(AC_{0d} + BC_{0n})P_{cd}F_{cd}}\\ &= \frac{P_{cn}D_f}{P_{cd}A}\varepsilon + \frac{(P_{cd}F_{ck}A - BF_{cd}P_{cn})}{(AC_{0d} + BC_{0n})P_{cd}F_{cd}}C_{0n}\frac{D_f}{A}\varepsilon.\end{aligned} \tag{13.45}$$

To obtain terms which are statistically independent, substitute from the diophantine equation (13.20):

$$\begin{aligned}\varphi_0 &= F_0\varepsilon + \frac{z^{-k}G_0}{P_{cd}A}\varepsilon + z^{-k}\frac{(P_{cd}F_{ck}A - B_kF_{cd}P_{cn})}{(AC_{0d} + BC_{0n})(P_{cd}F_{cd})}C_{0n}\frac{D_f}{A}\varepsilon\\ &= F_0\varepsilon + z^{-k}\frac{(G_0F_{cd}(AC_{0d} + BC_{0n})(P_{cd}F_{ck}A - B_kF_{cd}P_{cn})C_{0n}D_f)}{(AC_{0d} + BC_{0n})(P_{cd}F_{cd}A)}\varepsilon.\end{aligned} \tag{13.46}$$

Note that from (13.44) the numerator term may be simplified as

$$\begin{aligned}&G_0F_{cd}B + P_{cd}F_{ck}AD_f - B_kF_{cd}P_{cn}D_f\\ &= \big(G_0z^{-k} - P_{cn}D_f\big)F_{cd}B_k + P_{cd} + P_{cd}F_{ck}AD_f\\ &= (-P_{cd}AF_0)F_{cd}B_k + P_{cd}F_{ck}AD_f = -AP_{cd}H_0.\end{aligned}$$

The desired expression for the inferred output signal $\{\varphi_0(t)\}$ now follows from this result by substituting into (13.26):

$$\varphi_0 = F_0\varepsilon(t) + z^{-k}\frac{(C_{0d}G_0F_{cd} - C_{0n}H_0P_{cd})}{(AC_{0d} + BC_{0n})P_{cd}F_{cd}}\varepsilon(t). \tag{13.47}$$

This expression can be used to determine the optimal controller and the performance benchmark [20, 23]. For the benchmarking result, note the expression involves statistically independent terms and this is valid whether the controller $C_0 = C_{0d}^{-1} C_{on}$ is of full or reduced order.

Because of the definition of the white noise signal $\{\varepsilon(t)\}$ and the fact that $\deg(F_0) < k$, the first term is statistically independent of the remaining terms in (13.47). It follows that the first term on the right of (13.47) is independent of the choice of controller. That is, the first k terms of the closed-loop response between $\varepsilon(t)$ and $\varphi_0(t)$, represented by the polynomial F_0, are *feedback invariant.*

It also follows from this result that the full-order (unrestricted structure) optimal controller sets the final terms in (13.47) to zero. The function $\varphi_0(t)$, at the optimum, then follows as

$$\varphi_{0\min}(t) = F_0 \varepsilon(t). \tag{13.48}$$

Substitute for the optimal control (13.19), the full-order optimal *sensitivity-function*:

$$S = A_0 H_0 P_{cd} / (D_c D_f). \tag{13.49}$$

Using this expression for the optimal sensitivity function, the values of the full-order optimal error and control signals follow as

$$e_{\min}(t) \frac{P_{cd} H_0}{D_c U_0} \varepsilon(t) \quad \text{and} \quad u_{\min}(t) \frac{F_{cd} G_0}{D_c U_0} \varepsilon(t). \tag{13.50}$$

13.2.3.5 Restricted Structure Cost Expressions

As noted earlier, expression (13.40) is valid for any controller, and the two sets of terms are statistically independent. It follows that the variance of the signal $\{\varphi_0(t)\}$ can be computed for optimal or non-optimal controllers as $J = J_{\min} + J_0$ where

$$J_{\min} = \frac{1}{2\pi j} \oint_{|z|=1} \{F_0^* F_0\} \frac{dz}{z} \quad \text{and} \quad J_0 = \frac{1}{2\pi j} \oint_{|z|=1} \{T_0^* T_0\} \frac{dz}{z} \tag{13.51}$$

where T_0 is defined in terms of the nominal controller $C_0 = C_{0d}^{-1} C_{0n}$ as

$$T_0 = (C_{0d} G F_{cd} - C_{0n} H P_{cd}) / \big(P_{cd} F_{cd} (A C_{0d} + B C_{0n})\big). \tag{13.52}$$

Thus, a loop which contains a restricted structure (suboptimal) controller will have a variance of magnitude $J > J_{\min}$, and the best that can be achieved is $J_0 = 0$, and in this case, $J = J_{\min}$.

13.2.3.6 GMV Restricted Structure Optimal Control Benchmarking

It is more likely that an existing process plant will have a classically designed controller, of conventional structure (like a PID controller), than it is to use a full-order

GMV solution. The value of the cost in this case can be computed from (13.40). Although it is of some value to compare this cost J with the best that can be achieved, $J_{\min}$ (see (13.35)), this is often an unfair benchmark comparison. The reason is of course that the restriction on the controller structure and the order may make the absolute minimum $J_{\min}$ way below what is achievable, using even the best tuned classical controller.

A more appropriate benchmark figure is therefore to compare the actual cost with that of the optimal cost assuming that the controller structure is specified a priori or fixed. That is, if a loop must use, say an existing PID controller, then the benchmark cost figure required is in terms of the best choice of PID gains. Many authors have considered this type of *constrained structure* optimal control problem [2, 22, 28, 29, 31, 32] but the theorem introduced below (see [14, 17]) enables the optimisation problem to be expressed in a very simple form.

The performance assessment problem will be to minimise the criterion (13.40) with the controller chosen to have a pre-specified structure, such as:

Reduced order:

$$C_0(z^{-1}) = \frac{c_{n0} + c_{n1}z^{-1} + \cdots + c_{np}z^{-p}}{c_{d0}c_{d1}z^{-1} + \cdots + c_{dv}z^{-v}}$$

where $v \geq p$ is less than the order of the system (plus weightings).

Lead lag:

$$C_0(z^{-1}) = \frac{(c_{n0} + c_{n1}z^{-1})(c_{n2} + c_{n3}z^{-1})}{(c_{d0} + c_{d1}z^{-1})(c_{d2} + c_{d3}z^{-1})}.$$

PID:

$$C_0(z^{-1}) = k_0 + k_1/(1 - z^{-1}) + k_2(1 - z^{-1})/(1 - \alpha z^{-1}). \tag{13.53}$$

An assumption in the restricted structure benchmarking problem that must be made is that a stabilising control law exists for the assumed controller structure. Also note that the controller structure should be consistent with the choice of error weighting, in the sense that, if $P_{cd}(z^{-1})$ includes a near unit-circle zero, then the RS controller denominator $C_{0d}(z^{-1})$ should also include such a zero. Thus, for a PID controller, the weighting $P_{cd}^{-1}(z^{-1})$ should represent an integrator.

Theorem (GMV Restricted Structure Optimal Control Benchmark) *Given the solution of the diophantine equations* (13.20), (13.21), *the optimal controller of restricted structure*, *to minimise the cost function* (13.17), *can be found by minimising the simplified criterion*:

$$J_0 = \frac{1}{2\pi}\int_0^{2\pi} \{T_0^+(e^{j\theta})T_0^+(e^{-j\theta})\}\, d\theta \tag{13.54}$$

where $z = e^{j\theta}$, $A_w = P_{cd}F_{cd}$ *and*

$$T_0^+ = (C_{0d}(G_0F_{cd}) - C_{0n}(H_0P_{cd}))/(A_w(AC_{0d} + BC_{0n})). \tag{13.55}$$

If the controller has a specified limited structure, the minimum of the cost term J_0 (*denoted* $J_{0\,\min}$) *will be non-zero. The minimum value of the criterion* (13.35), *for the restricted structure optimal solution, will be obtained as* $J_{r\,\min} = J_{0\,\min} + J_{\min}$. *For an unconstrained solution, the minimum is achieved when* $T_0^+ = 0$, *and the minimum of the cost term* J_0 (*denoted* $J_{0\,\min}$) *is zero.*

Proof The proof follows directly from the above results.

Remark

(i) Equation (13.40) reveals the increase in cost which occurs by using a sub-optimal controller or by restricting the controller structure, namely.

$$\Delta J_{\min} = \frac{1}{2\pi j} \oint_{|z|=1} \{T_0^+ T_0^{+*}\} \frac{dz}{z}.$$

(ii) The *Controller Performance Index* will be defined as the ratio of the minimum possible variance of the cost function (13.35) to the actual variance:

$$\kappa = J_{\min}/(J_{\min} + J_0) = 1 - J_0/(J_{\min} + J_0)$$

and the CPI lies between $0 \le \kappa \le 1$. If κ is close to unity, the system provides little opportunity for improvement. If the CPI is close to zero, retuning is recommended. As noted previously, this scalar is similar but not the same, as the assessment measure introduced by Desborough and Harris [8].

The optimal restricted structure controller, for a given benchmark cost index, can easily be found using the results of [14]. The optimal value of the restricted structure cost term $J_{0\,\min}$ then follows from (13.40).

13.2.3.7 Example of Restricted Structure GMV Benchmarking

The following example illustrates the use of GMV performance benchmarking to assess the operation of a closed-loop control system. The aim is to compare the performance of three designs including:

(i) Optimal (full-order) control solution
(ii) Optimal tuned low-order (restricted structure) control solution.
(iii) Classically designed second-order controller

for the reference model $W_r = 1/(1+3s)$. The sampled system becomes:

System Description (Sample time $T_s = 1$ second)

Plant:

$$W = \frac{(z-0.04)(z-0.9)}{z^2(z-0.6)(z-0.49)(z-0.698)}$$

Disturbance model:

$$W_d = \frac{z}{z - 0.98}$$

Reference model:

$$W_r = \frac{z}{z - 0.98}$$

Error weighting:

$$P_c = \frac{1.05z - 0.95}{z - 1}$$

Control weighting:

$$F_c = \frac{-1(21z - 19)}{z^3(z + 1)}$$

Reference filter:

$$W_{\text{ref}} = \frac{0.2834687z}{z - 0.7165313}$$

Polynomial System Description

$$\begin{aligned}
A(z^{-1}) &= (z - 0.6)(z - 0.49)(z - 0.698)(z - 0.98)z^{-4}, \\
B(z^{-1}) &= (z - 0.4)(z - 0.9)(z - 0.98)z^{-6}, \\
U_0(z^{-1}) &= (z - 0.98)z^{-1}, \\
B_0(z^{-1}) &= (z - 0.04)(z - 0.9)z^{-5}, \\
A_0(z^{-1}) &= (z - 0.6)(z - 0.49)(z - 0.698)z^{-3}.
\end{aligned}$$

Cost-Function Weighting Choice

Error weighting:

$$P_c(z^{-1}) = \frac{1.05z - 0.95}{z - 1}$$

Control weighting:

$$F_c(z^{-1}) = \frac{-1(21z - 19)}{z^3(z + 1)}$$

The frequency responses of the error P_c and control F_c weighting terms are shown in Fig. 13.7. The equivalent of LQG error $Q_c = P_c^* P_c$ and control $R_c = F_c^* F_c$ weightings are also shown in this figure. Notice that the characteristics intersect at a frequency of 0.1 radian/second. A rule of thumb, which can be verified later, is that the unity-gain crossover frequency for the optimal open-loop transfer function will be close to this frequency.

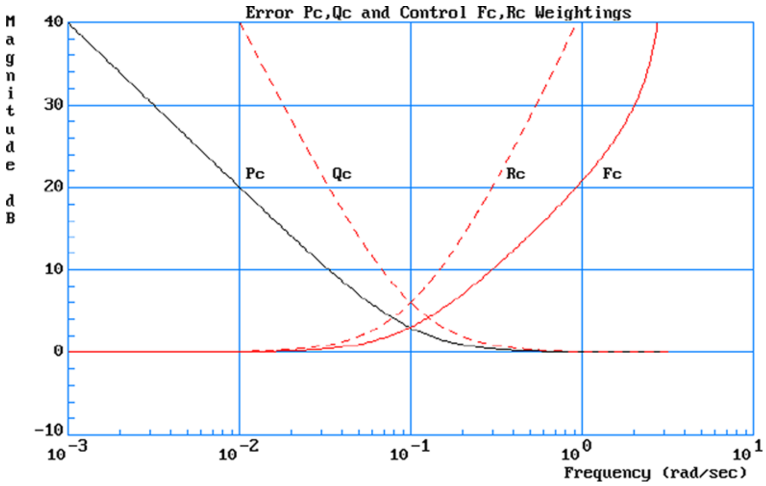


Fig. 13.7 Bode frequency responses of error and control signal weightings

Computed Spectral Term Recall that the choice of weightings should be such that the spectral term D_c is strictly Schur. From $D_c = P_{cn}F_{cd}B_{0k} - F_{ck}P_{cd}A_0$ in the theorem we obtain: $D_c(z^{-1}) = 22.05(z - 0.3812627)[(z - 0.6707553)^2 + 0.3186159^2](z - 0.9047619)(z - 0.9296076)z^{-5}$ which is strictly Schur by inspection.

Iterative Restricted Structure Gain Calculation Approximating the frequency interval by 25 points (see [15, 16]) and using 7 iterations to compute, we obtain the optimal restricted structure controller gains:

Step	Gain k_1	Gain k_2	Gain k_3
1	1	1	1
2	2.506077×10^{-3}	0.0715014	−0.0193219
3	0.0126633	0.0683640	−0.0236629
4	0.0120467	0.0687090	−0.0234386
5	0.0121582	0.0686745	−0.0234862
6	0.0121516	0.0686782	−0.0234838

Computed Controller Expressions

Full-order optimal controller:

$$C_0(z^{-1}) = \frac{C_{0n}}{C_{0d}}$$

where

$$C_{0n} = 5.815382 \times 10^{-2}(z-0.49)(z-0.6)(z-0.698)(z-0.9220146)(z+1)z^2,$$

$$C_{0d} = (z-0.36169)(z+0.1501461)\big[(z-0.4357288)^2+0.4992383^2\big]$$
$$\times (z-0.8851172)(z-0.9145451)(z-1)$$

Restricted structure optimal controller:

$$C_0(z^{-1}) = \frac{5.734642 \times 10^{-2}(z-0.4072771)(z+1.000294)}{(z-0.01)(z-1)}$$

Classically designed controller:

$$C_0(z^{-1}) = \frac{0.0173z(z-0.7)(z+1)}{(z+0.15)(z-0.9145)(z-1)}$$

Computed Benchmark Cost Values

Absolute minimum cost:	$J_{\min} = 7.66518$

Error and control signal costs

Absolute minimum error cost term:	$J_{e\min} = 9.444592$
Absolute minimum control cost:	$J_{u\min} = 17.35457$
Absolute minimum weighted error cost term:	$J_{p\min} = 31.62661$
Absolute minimum weighed control cost term:	$J_{f\min} = 23.96143$

Restricted structure controller cost terms

Restrained structure increase in cost:	$J_0 = 2.643136$
Restricted total cost:	$J = J_0 + J_{\min}$ and $J = 10.30832$
Controller performance index:	$\kappa = J_{\min}/(J_{\min} + J_0) = 0.7435918$
Restricted error cost:	$J_e = 9.91775$
Restricted control cost:	$J_u = 10.51039$

Classical controller cost terms

Classical controller increase in cost:	$J_0 = 9.629648$
Classical total cost:	$J = J_0 + J_{\min}$ and $J = 17.29483$
Controller performance index:	$\kappa = J_{\min}/(J_{\min} + J_0) = 0.4432065$
Classical error cost:	$J_e = 14.77414$
Classical control cost:	$J_u = 21.86504$

Frequency and Time-Domain Response Results The frequency responses of the three controllers are shown in Fig. 13.8. The restricted-structure controller response is clearly close to that of the optimal full-order solution, even though the controller is of much lower order. The classical solution is rather different in the mid-frequency range.

The corresponding open-loop transfer-function frequency responses are shown in Fig. 13.9, and as expected from the weighting choice, the unity-gain crossover

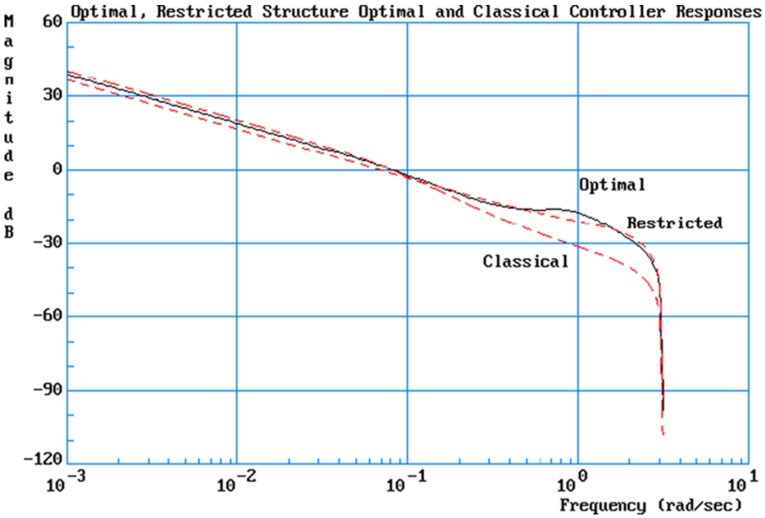


Fig. 13.8 Bode frequency responses of optimal, restricted structure and classically designed controllers

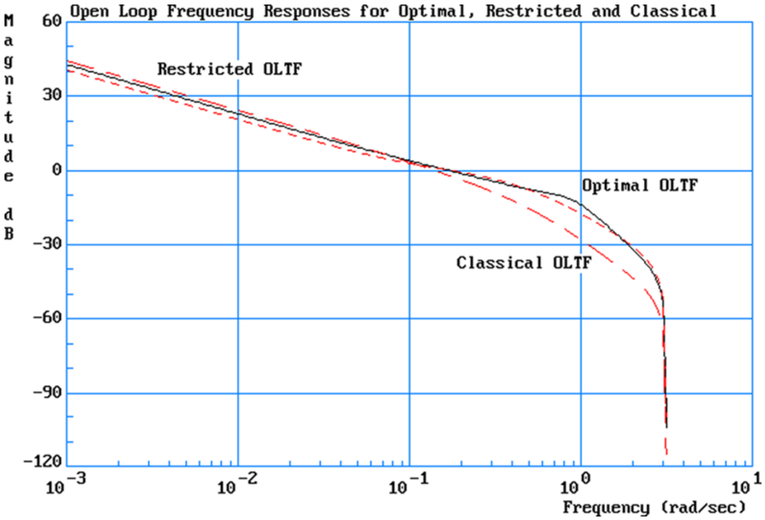


Fig. 13.9 Bode open-loop frequency responses of optimal, restricted structure and classical designs

frequency is close to 0.1 radians/second (actually 0.2 radian/second). The integral action at low frequencies is apparent in all the responses and the roll-off at high frequencies. The benchmarking results discussed below reveal a larger difference on variances than these frequency response results might suggest.

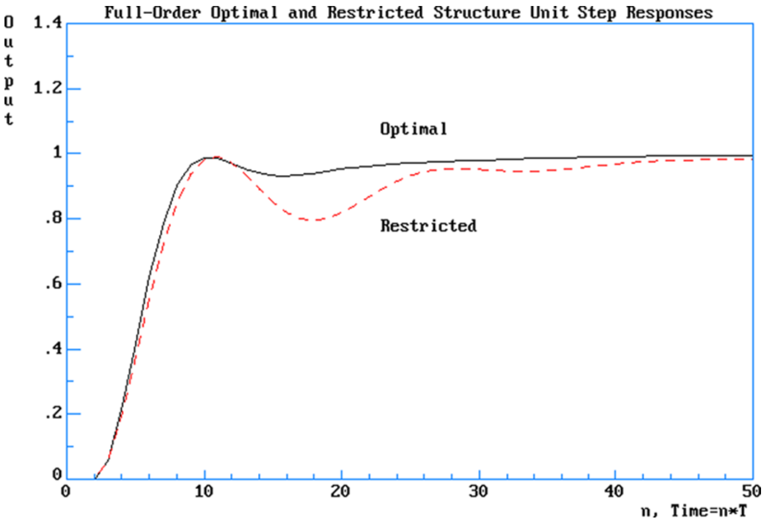


Fig. 13.10 Unit step responses of optimal and restricted structure control designs

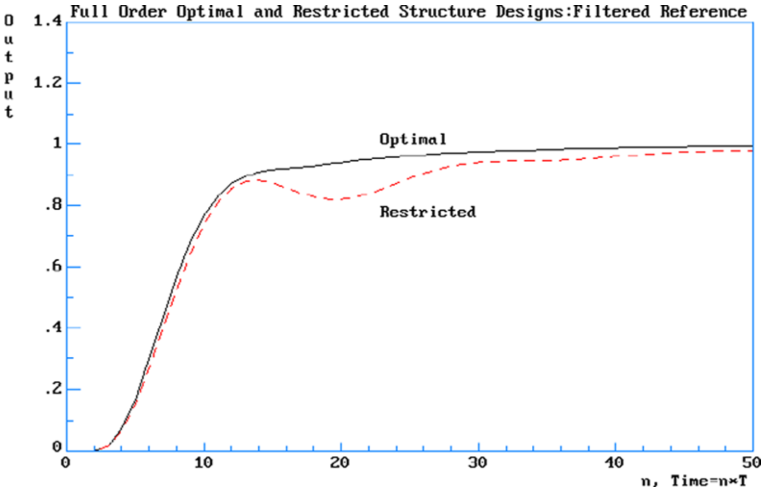


Fig. 13.11 Unit step responses of optimal and restricted structure control designs with a filtered reference

The unit-step responses of the full-order optimal and restricted structure control designs are shown in Fig. 13.10. The dominant time-constant is not too far from the $1/0.1 = 10$ seconds, predicted from the weighting choice. The restricted structure

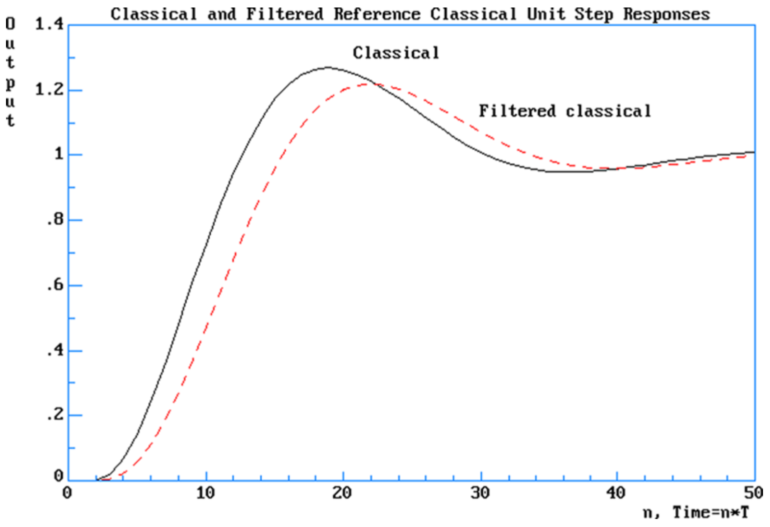


Fig. 13.12 Unit step responses of classical design with a filtered reference

solution is acceptable, particularly if the reference changes are not so severe as a step-function. The results in Fig. 13.11 are for the case where the step reference is filtered using W_{ref}.

These results may be compared with the unit step responses for the classically designed system (with and without the reference filter), shown in Fig. 13.12.

13.2.3.8 Benchmarking Results Discussion

The simplest indicators of performance are the computed values for the *controller performance index* (CPI) for the restricted structure and classical designs. As expected, the restricted structure design $\kappa \backsimeq 0.74$ is much better than the classical design $\kappa \backsimeq 0.44$. The optimal error $J_{e\,\mathrm{min}} \backsimeq 9.44$ and control $J_{u\,\mathrm{min}} \backsimeq 9.44$ may be compared with the restricted structure values $J_e \backsimeq 9.9$ and $J_u \backsimeq 10.5$ and with the classical design variance terms $J_e \backsimeq 14.77$ and $J_u \backsimeq 21.86$.

The classical error and control signal variances are all higher than the other designs, but the restricted structure solution has variances which are similar to those of the full-order optimal solution. In fact the control signal variance is actually lower than the full-order design, and this partly explains why the step responses are a little worse. The optimality is of course with respect to the GMV criterion J, and individual error and control variances can be higher or lower than the combined GMV cost term.

13.3 Performance Assessment for Tracking Tasks

The assessment of control performance for tracking tasks is based on comparison with performance obtained when using predictive controllers.

13.3.1 Equations of Predictive Controller

The derivations presented below are based on state-space models [25, 27]. State space models are now a common framework for predictive control; moreover, they are more convenient for representation of multivariable systems. Consider the system model in the state space form:

$$\begin{aligned} x_{t+1} &= A \cdot x_t + B \cdot u_t + G \cdot v_t, \\ y_t &= D \cdot x_t + w_t, \end{aligned} \tag{13.56}$$

where x_t is a vector of system states of size n_x, u_t is a vector of control signals of size n_u, y_t is a vector of output signals of size n_y, and A, B, G, D are constant matrices. v_t and w_t are vectors of disturbances, assumed to be Gaussian white noises with zero mean value and covariance matrix

$$E\left\{\begin{bmatrix} v_t \\ w_t \end{bmatrix}\right\}\begin{bmatrix} v_t^{\mathrm{T}} & w_t^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix}. \tag{13.57}$$

Remark The predictive control algorithms usually incorporate an integral action. It is possible to introduce the integral action in the controller design by changing the model so that it will have as the control input the increment in the control action Δu_t rather than the actual control. This can be achieved by the following substitution [18]:

$$\chi_{t+1} = \begin{bmatrix} x_{t+1} \\ u_t \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}\begin{bmatrix} x_t \\ u_{t-1} \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix}\Delta u_t + \begin{bmatrix} G \\ 0 \end{bmatrix} v_t, \tag{13.58}$$

$$y_t = [D \quad 0]\chi_t + w_t \tag{13.59}$$

where $\Delta u_t = u_t - u_{t-1}$ is a new control signal.

From (13.56) and (13.28), the k-step prediction of the output signal may be calculated from the relationship [26]

$$\hat{y}_{(t+k)/t} = E\{y_{t+k} \mid t\} = DA^k \hat{x}_{t/t} + \sum_{j=1}^{k} DA^{k-j} B u_{t+j-1}. \tag{13.60}$$

Collecting together the formulae for $\hat{y}_{(t+k)/t}$ when k changes from 1 to $N+1$, we obtain in a block matrix form:

$$\begin{aligned}
\hat{Y}_{t,N} &= \begin{bmatrix} \hat{y}_{(t+1)/t} \\ \hat{y}_{(t+2)/t} \\ \vdots \\ \hat{y}_{(t+N+1)/t} \end{bmatrix} \\
&= \begin{bmatrix} D \\ DA \\ \vdots \\ DA^N \end{bmatrix} A\hat{x}_{t/t} + \begin{bmatrix} DB & O & \dots & O \\ DAB & DB & & \vdots \\ \vdots & \vdots & \ddots & O \\ DA^N B & DA^{N-1}B & \dots & DB \end{bmatrix} \begin{bmatrix} u_t \\ u_{t+1} \\ \vdots \\ u_{t+N} \end{bmatrix}, \\
\hat{Y}_{t,N} &= \Phi_N A\hat{x}_{t/t} + S_N U_{t,N}
\end{aligned} \tag{13.61}$$

where $U_{t,N}$ is a block vector of $N+1$ future control signals.

The performance index to be minimised is defined as follows:

$$J_t = E\left\{\sum_{j=0}^{N}\left[(y_{t+j+1} - r_{t+j+1})^{\mathrm{T}} \breve{Q}_e (y_{t+j+1} - r_{t+j+1}) + u_{t+j}^{\mathrm{T}} \breve{Q}_u u_{t+j}\right]\right\} \tag{13.62}$$

where r_{t+j+1} represents a vector of reference (set point) signals, and $\breve{Q}_e > 0$ and $\breve{Q}_u \geq 0$ are weighting matrices. Denoting by $R_{t,N}$ a block vector of $N+1$ future reference signals, the performance index can be expressed in a static form:

$$J_t = (\hat{Y}_{t,N} - R_{t,N})^{\mathrm{T}} Q_e (\hat{Y}_{t,N} - R_{t,N}) + U_{t,N}^{\mathrm{T}} Q_u U_{t,N}. \tag{13.63}$$

Substituting (13.61) and finding the stationary point, the vector of optimal control signals can be calculated as

$$U_{t,N}\left(S_N^{\mathrm{T}} Q_e S_N + Q_u\right)^{-1} S_N^{\mathrm{T}} Q_e (R_{t,N} - \Phi_N A\hat{x}_{t/t}). \tag{13.64}$$

13.3.2 Calculation of the Performance in Closed-Loop System

Assume that the controller applied in the system is a multivariable feedback controller, defined as

$$U(z^{-1}) = \begin{bmatrix} c_o^{11}(z^{-1}) & \dots & c_o^{1m}(z^{-1}) \\ \vdots & \ddots & \vdots \\ c_o^{r1}(z^{-1}) & \dots & c_o^{rm}(z^{-1}) \end{bmatrix} \left(R(z^{-1}) - Y(z^{-1})\right) \tag{13.65}$$

where

$$U(z^{-1}) = \begin{bmatrix} u^1(z^{-1}) \\ \vdots \\ u^l(z^{-1}) \end{bmatrix}, \qquad R(z^{-1}) = \begin{bmatrix} r^1(z^{-1}) \\ \vdots \\ r^m(z^{-1}) \end{bmatrix} \quad \text{and}$$

$$Y(z^{-1}) = \begin{bmatrix} y^1(z^{-1}) \\ \vdots \\ y^m(z^{-1}) \end{bmatrix}.$$

The polynomial transfer functions $c_o^{ij}(z^{-1})$ are elements of the multivariable controller, for instance:

Reduced order:

$$C_0(z^{-1}) = \frac{c_{n0} + c_{n1}z^{-1} + \cdots + c_{np}z^{-p}}{c_{d0} + c_{d1}z^{-1} + \cdots + c_{dv}z^{-v}}$$

Lead lag:

$$C_0(z^{-1}) = \frac{(c_{n0} + c_{n1}z^{-1})(c_{n2} + c_{n3}z^{-1})}{(c_{d0} + c_{d1}z^{-1})(c_{d2} + c_{d3}z^{-1})}$$

PID:

$$C_0(z^{-1}) = k_0 + k_1\frac{1}{1-z^{-1}} + k_2\frac{1-z^{-1}}{1-\alpha z^{-1}}. \tag{13.66}$$

It is well known that a minimal order discrete-time state space equation can be obtained for each rational function describing the multivariable controller defined above:

$$\begin{aligned} x_{t+1}^{co} &= A_{co}x_t^{co} + B_{co}y_t + g_{co}\tilde{r}_t, \\ u_t &= C_{co}x_t^{co} + D_{co}y_t + f_{co}\tilde{r}_t. \end{aligned} \tag{13.67}$$

For consistency with the predictive control framework, the vector of $N+1$ future reference signals r_t will be considered as an input to the state space equations. The controller can then be rewritten in the form

$$\begin{aligned} x_{t+1}^{co} &= A_{co}x_t^{co} + B_{co}y_t + G_{co}\tilde{R}_{t,N}, \\ u_t &= C_{co}x_t^{co} + D_{co}y_t + F_{co}\tilde{R}_{t,N}. \end{aligned} \tag{13.68}$$

The structure of the closed-loop system is given in Fig. 13.13. The controller is described by the equations above, and the system by state-space equations (13.56).

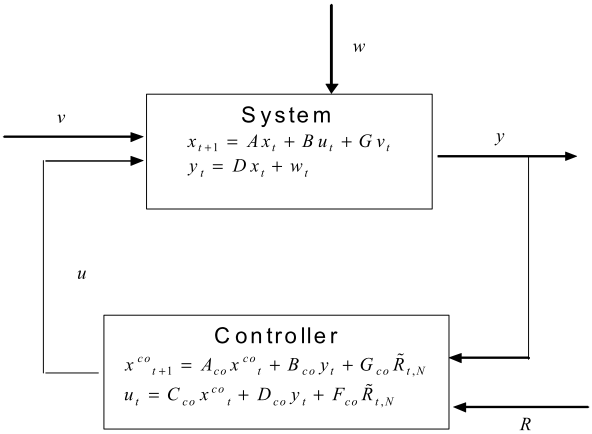

w
v
System
$x_{t+1} = Ax_t + Bu_t + Gv_t$
$y_t = Dx_t + w_t$
y
u
Controller
$x^{co}{}_{t+1} = A_{co}x^{co}{}_t + B_{co}y_t + G_{co}\tilde{R}_{t,N}$
$u_t = C_{co}x^{co}{}_t + D_{co}y_t + F_{co}\tilde{R}_{t,N}$
R

Fig. 13.13 Closed-loop system state-space block diagram

Combining together the equations for the system and for the controller, we obtain:

$$\begin{aligned}\chi_{t+1} = \begin{bmatrix} x_{t+1} \\ x^{co}_{t+1} \end{bmatrix} &= \begin{bmatrix} A + BD_{co}D & BC_{co} \\ B_{co}D & A_{co} \end{bmatrix} \begin{bmatrix} x_t \\ x^{co}_t \end{bmatrix} \\ &+ \begin{bmatrix} BD_{co} & BG \\ B_{co} & O \end{bmatrix} \begin{bmatrix} w_t \\ v_t \end{bmatrix} + \begin{bmatrix} BF_{co} \\ G_{co} \end{bmatrix} \tilde{R}_{t,N}, \qquad (13.69) \\ \zeta_t = \begin{bmatrix} u_t \\ y_t \end{bmatrix} &= \begin{bmatrix} D_{co}D & C_{co} \\ D & O \end{bmatrix} \begin{bmatrix} x_t \\ x^{co}_t \end{bmatrix} + \begin{bmatrix} D_{co} \\ I \end{bmatrix} w_t + \begin{bmatrix} F_{co} \\ O \end{bmatrix} \tilde{R}_{t,N} .\end{aligned}$$

Note that, for predictive controllers,

$$\tilde{R}_{t,N} = R_{t,N} = \begin{bmatrix} r_{t+1} \\ r_{t+3} \\ \vdots \\ r_{t+N+1} \end{bmatrix}$$

and, for non-predictive controllers,

$$\tilde{R}_{t,N} = \begin{bmatrix} r_t \\ r_{t+3} \\ \vdots \\ r_{t+N} \end{bmatrix} .$$

Example 13.1 Consider state-space equations of the Generalised Predictive Controller. From the perspective of the state-space equations, the "output" here is the control signal, the "inputs" are: the plant output and the reference signal, and the state can be defined as the state of the Kalman filter. Starting with the "output" equation, the control signal at time instant t is the first (vector) component (n_u elements) of the vector $U_{t,N}$:

$$\begin{aligned} u_t &= [I \quad O \quad \ldots \quad O]\big(S_N^{\mathrm{T}} Q_e S_N + Q_u\big)^{-1} S_N^{\mathrm{T}} Q_e \big(R_{t,N} - \Phi_N A \hat{x}_{t/t}\big) \\ &= k_{\mathrm{GPC}}^R R_{t,N} + k_{\mathrm{GPC}}^x \hat{x}_{t/t}. \end{aligned} \tag{13.70}$$

Replacing the state estimate with the state prediction obtained from a stationary Kalman filter

$$\begin{aligned} &\hat{x}_{t/t}(I - KD)\hat{x}_{t/t-1} + K y_t, \quad \text{one obtains as the “output equation”}, \\ &u_t = k_{\mathrm{GPC}}^R R_{t,N} + k_{\mathrm{GPC}}^x (I - KD)\hat{x}_{t/t-1} + k_{\mathrm{GPC}}^x K y_t. \end{aligned} \tag{13.71}$$

The state prediction can be calculated from the *Kalman filter* predictor equation:

$$\begin{aligned} \hat{x}_{t+1/t} &= (A - AKD)\hat{x}_{t/t-1} + AK y_t + B u_t \\ &= \big((A - AKD) + B k_{\mathrm{GPC}}^x (I - KD)\big)\hat{x}_{t/t-1} \\ &\quad + \big(AK + B k_{\mathrm{GPC}}^x K\big) y_t + B k_{\mathrm{GPC}}^R R_{t,N}. \end{aligned} \tag{13.72}$$

From the above two equations, the parameters of the state-space model of GPC are:

$$\begin{aligned} A_{co}^{\mathrm{GPC}} &= A - AKD + B k_{\mathrm{GPC}}^x (I - KD), \\ B_{co}^{\mathrm{GPC}} &= AK + B k_{\mathrm{GPC}}^x K, \\ G_{co}^{\mathrm{GPC}} &= B k_{\mathrm{GPC}}^R, \\ C_{co}^{\mathrm{GPC}} &= k_{\mathrm{GPC}}^x (I - KD), \\ D_{co}^{\mathrm{GPC}} &= k_{\mathrm{GPC}}^x K, \\ F_{co}^{\mathrm{GPC}} &= k_{\mathrm{GPC}}^R. \end{aligned} \tag{13.73}$$

Example 13.2 PID controller. Consider the single-input, single-output discrete-time PID given by (13.66). The state-space equations for this PID controller can be developed in the following form:

$$\begin{aligned} x_{t+1} &= \begin{bmatrix} 1 & 0 \\ 0 & \alpha \end{bmatrix} x_t + \begin{bmatrix} k_t \\ -k_2(1-\alpha) \end{bmatrix} (r_t - y_t), \\ u_t &= [1 \quad 1] x_t + (k_0 + k_1 + k_2)(r_t - y_t). \end{aligned} \tag{13.74}$$

Hence:

$$\begin{aligned} A_{co}^{\mathrm{PID}} &= \begin{bmatrix} 1 & 0 \\ 0 & \alpha \end{bmatrix}, \\ B_{co}^{\mathrm{PID}} &= \begin{bmatrix} -k_1 \\ k_2(1-\alpha) \end{bmatrix}, \end{aligned}$$

$$\begin{aligned}
G_{co}^{\mathrm{PID}} &= \begin{bmatrix} -k_1 \\ -k_2(1-\alpha) \end{bmatrix} [1 \quad 0 \quad \ldots \quad 0], \\
C_{co}^{\mathrm{PID}} &= [1 \quad 1], \\
D_{co}^{\mathrm{PID}} &= -(k_0 + k_1 + k_2), \\
F_{co}^{\mathrm{PID}} &= (k_0 + k_1 + k_2)[1 \quad 0 \quad \ldots \quad 0].
\end{aligned}$$

13.3.3 Time Evaluation of the Mean Value and Variance

Given a system as described by (13.56) and a multivariable controller as defined in (13.68), combined system equations can be derived, to evaluate the stochastic characteristics of the closed-loop system. From (4.2.76) the system parameters in the extended state equation can be defined as follows:

$$\begin{aligned}
&A_{\mathrm{sys}} = \begin{bmatrix} A + BD_{co}D & BC_{co} \\ B_{co}D & A_{co} \end{bmatrix}, \qquad B_{\mathrm{sys}} \begin{bmatrix} BF_{co} \\ G_{co} \end{bmatrix}, \\
&G_{\mathrm{sys}} = \begin{bmatrix} BD_{co} & BG \\ B_{co} & O \end{bmatrix}, \\
&\chi_{t+1} = \begin{bmatrix} x_{t+1} \\ x_{t+1}^{co} \end{bmatrix}, \qquad y_{t+1} = \begin{bmatrix} w_t \\ v_t \end{bmatrix}, \qquad P_{\mathrm{sys}} = \begin{bmatrix} D_{co}D & C_{co} \end{bmatrix}, \\
&D_{\mathrm{sys}} = \begin{bmatrix} D & O \end{bmatrix}.
\end{aligned} \tag{13.75}$$

Then by taking the expectation of (4.276) the mean of system states can be expressed as

$$E\{\chi_{t+1}\} = \bar{\chi}_{t+1} = \begin{bmatrix} \bar{x}_{t+1} \\ \bar{x}_{t+1}^{co} \end{bmatrix} = A_{\mathrm{sys}}\bar{\chi}_t + B_{\mathrm{sys}}\tilde{R}_{t,N}. \tag{13.76}$$

Define the mean square of the system states as $\underline{\chi}_{t+1} = \mathrm{trace}\{E\langle \chi_{t+1}\chi_{t+1}^{\mathrm{T}}\rangle\}$. Then following the analyses of the stochastic properties of the *GPC/LQGPC* algorithm as considered by Ordys [25], by using (13.75), we deduce:

$$\underline{\chi}_{t+1} = \mathrm{trace}\left\{ \begin{array}{l} A_{\mathrm{sys}}E\langle \chi_t\chi_t^{\mathrm{T}}\rangle A_{\mathrm{sys}}^{\mathrm{T}} + B_{\mathrm{sys}}\tilde{R}_{t,N}\bar{\chi}_t^{\mathrm{T}}A_{\mathrm{sys}}^{\mathrm{T}} \\ \quad + A_{\mathrm{sys}}\bar{\chi}_t\tilde{R}_{t,N}^{\mathrm{T}}B_{\mathrm{sys}}^{\mathrm{T}} + B_{\mathrm{sys}}\tilde{R}_{t,N}\tilde{R}_{t,N}^{\mathrm{T}}B_{\mathrm{sys}}^{\mathrm{T}} + G_{\mathrm{sys}}E\langle \gamma_t\gamma_t^{\mathrm{T}}\rangle G_{\mathrm{sys}}^{\mathrm{T}} \end{array} \right\}, \tag{13.77}$$

$$\begin{aligned}
\underline{\chi}_{t+1} = \mathrm{trace}\big\{ & A_{\mathrm{sys}}\underline{\chi}_t A_{\mathrm{sys}}^{\mathrm{T}} + B_{\mathrm{sys}}\tilde{R}_{t,N}\bar{\chi}_t A_{\mathrm{sys}}^{\mathrm{T}} + A_{\mathrm{sys}}\bar{\chi}_t\tilde{R}_{t,N}^{\mathrm{T}}B_{\mathrm{sys}}^{\mathrm{T}} \\
& + B_{\mathrm{sys}}\tilde{R}_{t,N}\tilde{R}_{t,N}^{\mathrm{T}}B_{\mathrm{sys}}^{\mathrm{T}} + G_{\mathrm{sys}}\underline{\gamma}_t G_{\mathrm{sys}}^{\mathrm{T}}\big\}.
\end{aligned} \tag{13.78}$$

Let the state co-variance be defined as

$$\begin{aligned}\underline{\tilde{\chi}}_{t+1} &= E\left\{(\chi_{t+1} - \bar{\chi}_{t+1})(\chi_{t+1} - \bar{\chi}_{t+1})^{\mathrm{T}}\right\},\\ \underline{\tilde{\chi}}_{t+1} &= A_{\mathrm{sys}} E\left\{(\chi_t - \bar{\chi}_t)(\chi_t - \bar{\chi}_t)^{\mathrm{T}}\right\} A_{\mathrm{sys}}^{\mathrm{T}} + G_{\mathrm{sys}} E\left\{\gamma_t \gamma_t^{\mathrm{T}}\right\} G_{\mathrm{sys}}^{\mathrm{T}},\end{aligned} \tag{13.79}$$

$$\underline{\tilde{\chi}}_{t+1} = A_{\mathrm{sys}} \underline{\tilde{\chi}}_t A_{\mathrm{sys}}^{\mathrm{T}} + G_{\mathrm{sys}} \underline{\gamma}_t G_{\mathrm{sys}}^{\mathrm{T}}. \tag{13.80}$$

The mean and variances of the output (y_t) and input (u_t) signals can be obtained from (13.69) as:

$$E\{y_{(t)}\} = \bar{y}_{(t)} = D_{\mathrm{sys}} \bar{\chi}_t, \tag{13.81}$$

$$E\{u_{(t)}\} = \bar{u}_{(t)} = P_{\mathrm{sys}} \bar{\chi}_t + f_{co} r_t \tag{13.82}$$

and

$$\underline{\tilde{y}}_{(t)} = E\left\{(y_t - \bar{y}_t)(y_t - \bar{y}_t)^{\mathrm{T}}\right\} = D_{\mathrm{sys}} E\left\{(\chi_t - \bar{\chi}_t)(\chi_t - \bar{\chi}_t)^{\mathrm{T}}\right\} D_{\mathrm{sys}}^{\mathrm{T}} + E\left\{w_t w_t^{\mathrm{T}}\right\}, \tag{13.83}$$

$$\underline{\tilde{y}}_{(t)} = D_{\mathrm{sys}} \underline{\chi}_t D_{\mathrm{sys}}^{\mathrm{T}} + \underline{w}_t, \tag{13.84}$$

$$\begin{aligned}\underline{\tilde{u}}_{(t)} &= E\left\{(u_t - \bar{u}_t)(u_t - \bar{u}_t)^{\mathrm{T}}\right\}\\ &= P_{\mathrm{sys}} E\left\{(\chi_t - \bar{\chi}_t)(\chi_t - \bar{\chi}_t)^{\mathrm{T}}\right\} P_{\mathrm{sys}}^{\mathrm{T}} + D_{co} E\left\{w_t w_t^{\mathrm{T}}\right\} D_{co}^{\mathrm{T}},\end{aligned} \tag{13.85}$$

$$\underline{\tilde{u}}_{(t)} = P_{\mathrm{sys}} \underline{\chi}_t P_{\mathrm{sys}}^{\mathrm{T}} + D_{co} \underline{w}_t D_{co}^{\mathrm{T}}. \tag{13.86}$$

13.3.4 Control Benchmarking

Benchmarking for tracking tasks requires the definition of the performance index taking into account a time horizon sufficiently long for the tracking to be completed, i.e. the output has aligned with the reference (set-point). Practically, the horizon should be several times longer than the dominant time constant of the closed-loop system. Theoretically, required horizon would be infinite. Hence, the performance measure, useful for benchmarking purposes, could be proposed as

$$J = E\left\{\lim_{\mathrm{T}\to\infty} \frac{1}{T+1} \sum_{t=0}^{\mathrm{T}} \left[(Y_{t,N} - R_{t,N})^{\mathrm{T}} Q_e (Y_{t,N} - R_{t,N}) + Y_{t,N}^{\mathrm{T}} Q_u U_{t,N}\right]\right\}, \tag{13.87}$$

$$J = \lim_{\mathrm{T}\to\infty} \frac{1}{T+1} \sum_{t=0}^{\mathrm{T}} E_{\mathrm{wms}}(t) + U_{\mathrm{wms}}(t)$$

where $E_{\mathrm{wms}}(t)$ and $U_{\mathrm{wms}}(t)$ are the weighted mean squares of the process output error and of the process input, respectively.

The approach to controller benchmarking, using the performance index defined above, is described in [27].

However, there are ambiguities associated with this process. The tracking performance is related to a changing reference signal. Future values of the reference signal may be known within a certain (finite) horizon but, in most cases, would be unknown beyond such finite horizon. Hence:

1. the benchmark index should be updated step by step
   - to allow for incorporation of future references when they become available,
   - to take into account the effect of disturbances which may drive the state of the system away from the assumed trajectory.
2. In the benchmark index calculated at a given time the values of future outputs and future controls will need to be the "predicted values" rather than the "actual values", because the actual values are not available at the time of calculation.
3. For "reactive controllers", such as *PID*, the predicted values can be calculated in several ways, depending on the assumptions used.
4. The optimal *PID* controller for the tracking performance, with the performance index defined above, can be calculated. However, as it follows from the arguments above, the parameters of the controller will change step by step (see also [37, 38]).

The definition and calculation of the benchmarks for PID controllers for tracking tasks remains an open problem. One possible simple approach is to use the equations from the previous section and examine how the variances of tracking error and of input signal change during transient periods.

Example Consider the adaptive cruise control problem, described at the beginning of this chapter. Assume the following scenario: Initially the vehicle drives at a constant speed of 20 m/s. At time 30 s, a "vehicle in front" is detected, and, consequently, the velocity is reduced to match that of the "vehicle in front". At time 40 s, the two vehicles have synchronised velocities, and now they travel in "constant distance" mode. At time 60 s, a higher speed limit on the road is encountered. Consequently, the "vehicle in front" accelerates, and the controlled vehicle follows up to a new cruise control speed of 30 m/s and then continues at that speed.

Figure 13.14 shows the velocity of the controlled vehicle as a function of time. There are three cases where the velocity is controlled by PI, LQ or predictive algorithm. In all three cases tracking is performed well, with LQ controller reacting only a little slower than the other two. Figures 13.15, 13.16, and 13.17 show:

- The variance of the input signal (acceleration or braking).
- The variance of the tracking error.
- The weighted sum of the two (where the weight is the same as used in both the LQ and predictive performance indices).

Notice that, for this particular case, LQ controller shows the lowest variance of the input signal during the periods of the speed increase but the highest during the

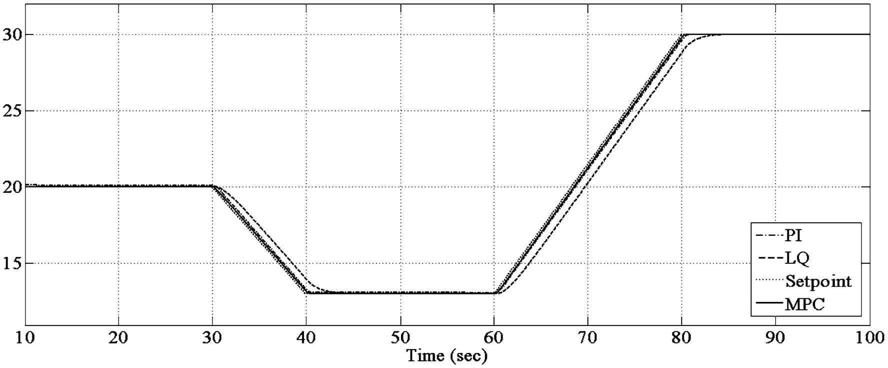


Fig. 13.14 Velocity as a function of time

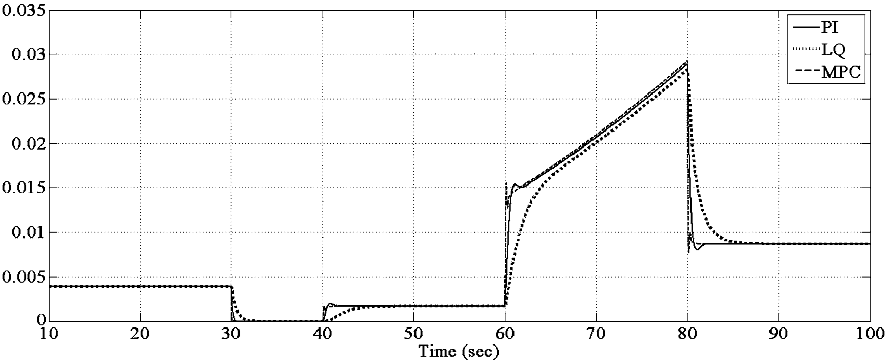


Fig. 13.15 Variance of the controlled input (u^2) as a function of time

periods of speed decrease. In terms of variance of the tracking error and the weighted sum, the predictive controller is outperforming the other two.

13.4 Concluding Remarks

The aim was to provide benchmarking criteria by which the performance of restricted structure controllers, and particularly PID designs, could be assessed and tuned. The *minimum-variance* benchmark criterion has proved successful in commercial industrial control systems, since the results are easy to use for controller performance assessment. However, the control signal variations are normally unrealistic, and such a controller cannot be used in practice. Such a criterion is not therefore so valuable for assessing real systems [24]. The GMV control law overcomes the problem of large control signal variations and therefore provides a more meaningful measure for comparison with classical designs.

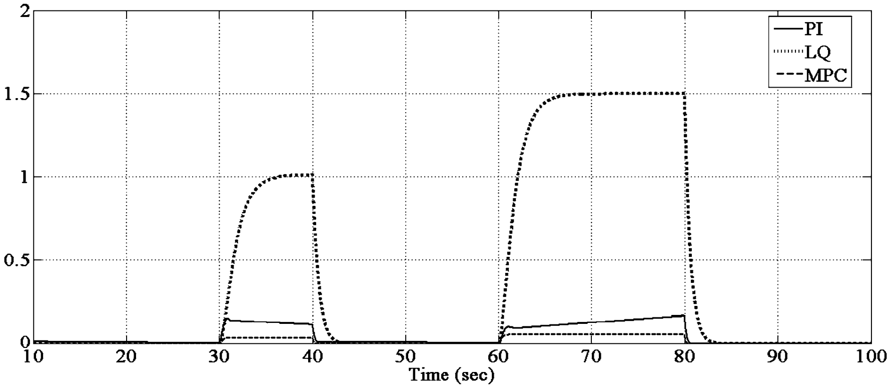


Fig. 13.16 Variance of the tracking error $(r - y)^2$ as a function of time

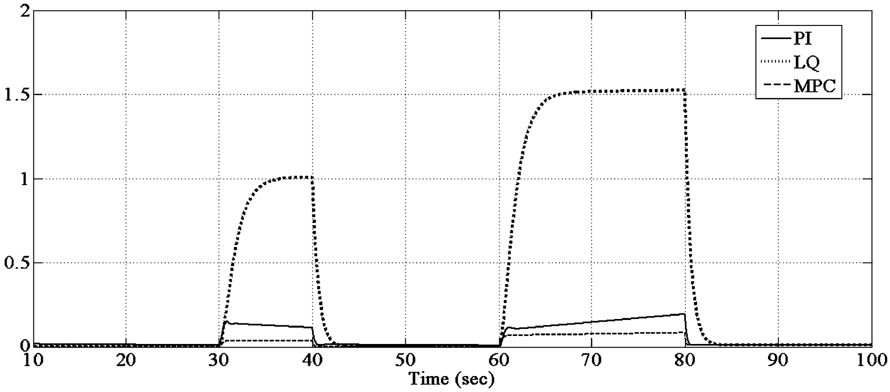


Fig. 13.17 Weighted sum of input and error variances as a function of time

A second contribution was the development of GMV benchmark cost expressions of controllers of restricted (simple) structure. A benchmark is not so valuable if it implies the use of a high-order controller, which is unlikely ever to be used. The restricted structure GMV optimal benchmark cost is more useful, since it provides a direct measure of how poorly tuned an existing PID (or classically tuned controller) might be. The advantage over comparable H_2 benchmarking methods [12, 35] lies in its simplicity.

Finally, the performance assessment for tracking tasks was introduced. In this case, the predictive controller was used as a base for comparison with other algorithms (including PID). The equations derived in this section enable calculation of variances of input and output signals for transient periods of operation of the system.

Performance assessment of controllers has already proven to be a valuable tool in the portfolio of plant operators. It is being routinely used in process industry, especially petrochemical plants. In this type of applications, there is a prevailing dominance of PID control on regulatory level and of Predictive Control on supervisory level (tracking tasks). Hence, the methods proposed in this chapter will have immediate relevance. Moreover, other industries can benefit for those approaches, as illustrated by an example from automotive sector.

The challenge of current research in this field is to make the performance assessment methods more compatible with real-life situations. Steps in this direction are the restricted structure benchmarking and benchmarking for transient processes. Other research challenges include:

- Performance assessment method(s) which would cope with regulatory tasks and tracking tasks at the same time.
- Performance assessment for gain scheduled controllers.
- Realistic measures for performance assessment in strongly nonlinear systems.

Acknowledgements The authors would like to thank Payman Shakouri for a help in numerical examples related to the Adaptive Cruise Control system.

References

1. Ahmad, M., Benson, R.: Benchmarking in the process industries. In: IChemE. The Cromwell Press, Trowbridge (1999)
2. Anderson, B.D.O., Lin, Y.: Controller reduction concepts and approaches. IEEE Trans. Autom. Control **34**, 802–812 (1989)
3. Andersen, B., Pettersen, P.G.: The Benchmarking Handbook. Chapman and Hall, New York (1996)
4. Calligaris, M., Johnson, M.A.: Power system studies and new benchmarking concepts. In: Proc. IASTED Int. Conf. on Control and Applications, Banff, Canada, pp. 211–216 (1999)
5. Calligaris, M., Johnson, M.A.: Benchmarking for hierarchical voltage control. In: IFAC Symp. on Power Plants and Power Systems Control 2000, Brussels, Belgium (2000)
6. Clarke, D.W., Hastings-James, R.: Design of digital controllers for randomly disturbed systems. Proc. IEEE **118**(10), 1503–1506 (1971)
7. Codling, S.: Best Practice Benchmarking. Uniskill Ltd. (1992)
8. Desborough, L.D., Harris, T.J.: Performance assessment measures for univariate feedback control. Can. J. Chem. Eng. **70**, 1186–1197 (1992)
9. Desborough, L.D., Harris, T.J.: Performance assessment measures for univariate feedback/feedforward control. Can. J. Chem. Eng. **71**, 605–615 (1993)
10. Desborough, L.D., Harris, T.J.: Control performance assessment. Pulp Pap. Can. **95**(11), 441–443 (1994)
11. Grimble, M.J.: Generalised minimum variance control law revisited. Optim. Control Appl. Methods **9**, 63–77 (1988)
12. Grimble, M.J.: Restricted structure controller tuning and performance assessment. IEE Proc. Part D (1992)
13. Grimble, M.J.: Robust Industrial Control: Optimal Design Approach for Polynomial Systems. Prentice Hall, New York (1994)
14. Grimble, M.J.: Restricted structure feedforward stochastic optimal control. In: CDC 99, Phoenix, Arizona (1999)
15. Grimble, M.J.: Restricted structure LQG optimal control for continuous-time systems. IEE Proc. Control Theory Appl. **147**(2), 185–195 (2000)
16. Grimble, M.J.: Industrial Control Systems Design. Wiley, Chichester (2000)
17. Grimble, M.J.: Restricted structure optimal linear pseudo-state filtering for discrete-time systems. In: American Control Conference, Chicago (2000)

18. Hangstrup, M., Ordys, A.W., Grimble, M.J.: Dynamic algorithm of LQGPC predictive control. In: 4th International Symposium on Methods and Models in Automation and Robotics, Miedzyzdroje, August 1997
19. Harris, T.: Assessment of closed loop performance. Can. J. Chem. Eng. **67**, 856–861 (1989)
20. Huang, B., Shah, S.L., Kwok, E.K.: Good, bad or optimal, Performance assessment of multivariable processes. Automatica **33**(6), 1175–1183 (1997)
21. Huang, B., Shah, S.L.: Performance Assessment of Control Loops: Theory and Applications. Springer, London (1999)
22. Hyland, D.C., Bernstein, D.A.: The optimal projection equations for model reduction and the relationships among the methods of Wilson, Skelton and Moore. IEEE Trans. Autom. Control **30**, 1201–1211 (1985)
23. Levine, W.S. (ed.): The Control Handbook. CRC Press, Boca Raton (1996), Chap. 62
24. Moden, P.E., Soderstrom, T.: On the achievable accuracy in stochastic control. In: 17th IEEE Conf on Decision and Control, pp. 490–495 (1978)
25. Ordys, A.W.: Model-system parameters mismatch in GPC control. Int. J. Adapt. Control Signal Process. **7**, 239–253 (1993)
26. Ordys, A.W., Clarke, D.W.: A state-space description for GPC controllers. Int. J. Syst. Sci. **24**(9) (1993)
27. Ordys, A.W., Uduehi, D., Johnson, M.: Process Control Performance Assessment. From Theory to Implementation. Monograph Series: Advances in Industrial Control. Springer, London (2007). ISBN 978-1-84628-623-0
28. Rivera, D.E., Morari, M.: Control relevant model reduction problems for SISO H_2, H_∞ and μ controller synthesis. Int. J. Control **46**(2), 505–527 (1987)
29. Rivera, D.E., Morari, M.: Low order SISO controller tuning methods for the H_2, H_∞ and μ objective functions. Automatica **26**(2), 361–369 (1990)
30. Rolstadas, A.: Benchmarking—Theory and Practice. Chapman and Hall, London (1995)
31. Saeki, M., Kimura, J.: Design method of robust PID controller and CAD system. In: 11th IFAC Symposium on System Identification, vol. 3, pp. 1587–1593 (1997)
32. Saeki, M., Aimoto, K.: PID controller optimization for H_∞ control by linear programming. Int. J. Robust Nonlinear Control **10**, 83–99 (2000)
33. Shakouri, P., Ordys, A., Askari, M., Laila, D.S.: Longitudinal vehicle dynamics using Simulink/Matlab. In: UKACC International Conference CONTROL 2010, Coventry, 7–10 September 2010
34. Shakouri, P., Ordys, A., Laila, D.S., Askari, M.: Adaptive cruise control system: Comparing gain-scheduling PI and LQ controllers. In: IFAC Word Congress, Milano, 28 August–2 September 2011
35. Stanfelj, N., Marlin, T.E., MacGregor, J.F.: Monitoring and diagnosing process control performance: the single loop case. Ind. Eng. Chem. Res. **32**, 301–314 (1993)
36. Thornhill, N.F., Oettinger, M., Fedenczuk, P.: Refinery-wide control loop performance assessment. J. Process Control **9**, 109–124 (1999)
37. Uduehi, D., Ordys, A.: Multivariable PID controller design using online generalised predictive control optimisation. In: IEEE Conference on Control Applications & IEEE Conference on Computer Aided Control Systems Design, Glasgow, 17–20 September 2002
38. Uduehi, D., Ordys, A., Grimble, M.: A generalised predictive control benchmark index for SISO systems. In: IEEE Control and Decision Conference, Las Vegas, December 2002

Chapter 14
Industrial Applications of PID Control

Gregory K. McMillan

Nomenclature

a	PID process variable amplitude for relay method of auto tuning (%)
B_v	valve backlash (deadband) (%)
d	PID output step size for relay method of auto tuning (%)
n	PID process variable noise band for relay method of auto tuning (%)
A_o	amplitude of limit cycle (%)
CO_{t1}	controller output at time $t1$ before correction for load disturbance (%)
CO_{t2}	controller output at time $t2$ after correction for load disturbance (%)
E_i	integrated error (% s)
E_L	open loop error corrected for load disturbance time constant (%)
E_o	open loop error for unmeasured step disturbance (%)
E_x	peak error (%)
K_c	PID gain (dimensionless)
K_i	integrating process gain (% per second per %)
K_o	open loop gain for self-regulating processes (dimensionless)
K_u	ultimate gain (dimensionless)
K_v	valve gain (Flow e.u./CO %)
K_p	process gain (PV e.u./Flow e.u.)
K_m	measurement gain (PV %/PV e.u.)
N_m	measurement noise (%)
$\mathrm{PV}_{100\%}$	process variable at 100% of measurement scale (PV e.u.)
$\mathrm{PV}_{0\%}$	process variable at 0% of measurement scale (PV e.u.)
$\Delta\mathrm{CO}$	change in controller output (%)
$\Delta\mathrm{CO}_{\max}$	maximum available change in controller output to output limit (%)
$\Delta\mathrm{PV}$	change in controller process variable (%)
$\Delta\mathrm{SP}$	change in controller setpoint (%)

G.K. McMillan (✉)
CDI Process & Industrial, 11404 Dodge Cattle Dr., Austin, TX 78717, USA
e-mail: gkmcmi@austin.rr.com

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_14, © Springer-Verlag London Limited 2012

S_{1x}	1st split ranged span (PV e.u.)
S_{2x}	2nd split ranged span (PV e.u.)
S_v	valve stick-slip (resolution-threshold sensitivity) (%)
S_m	wireless trigger level (measurement threshold sensitivity) (%)
T_i	PID integral time (s/repeat)
T_o	period of limit cycle (s)
T_r	rise time of setpoint response (s)
T_u	ultimate period (s)
λ	closed loop time constant for setpoint change (s)
θ_i	implied total loop deadtime (s)
θ_o	original total loop deadtime (s)
θ_s	effective deadtime from threshold sensitivity setting (s)
$\theta_{\Delta T}$	effective deadtime from update time interval (s)
θ_v	effective deadtime from valve resolution-sensitivity (stick-slip) (s)
θ_w	effective deadtime from wireless settings (s)
τ_f	signal filter or volume attenuating time constant (s)
τ_L	load disturbance time constant (s)
τ_o	open loop time constant (s)
τ_p	process time constant (s)
ΔT_x	controller execution time (s)

14.1 Introduction

The PID controller is an essential part of the control loop in the process industry [1]. Studies have shown that the PID provides an optimal solution of the regulator problem (rejection of disturbances) and with simple enhancements, provides an optimum servo response (setpoint response) [3]. Tests show that the PID performs better than Model Predictive Control (MPC) for unmeasured disturbances in terms of peak error, integrated error, or robustness [8]. The PID controller in the modern Distributed Control System (DCS) has an extensive set of features. However, primarily due to the lack of understanding of the functionality and applicability of the PID, the full power of the PID is rarely utilized [22]. This section explores key PID features and provides examples of their importance for addressing challenging applications and control objectives for common unit operation applications in the process industry.

Industrial processes are characterized by unmeasured disturbances, nonlinear process dynamics, noise, measurement delays and lags, resolution and sensitivity limits, and valve nonlinearities and non-idealities. It will be shown that the total PID loop deadtime in industrial processes determines the ultimate limit to loop performance. The total loop deadtime has many sources most of which are variable. The process deadtimes and time constants are rarely constant. In a first order plus deadtime approximation, all of the time constants smaller than the largest open loop time constant (τ_o) become an equivalent deadtime (θ_o). The fraction of the small time constant converted to deadtime approaches 1 as the ratio of small to largest time constant approaches 0 [6]. Examples of small time constants are valve actuator lags, process heat transfer and mixing lags, thermowell and sensor lags, transmitter damping settings, and signal filters. The deadtime from these lags are summed with the pure delays from valve pre-stroke delay, valve backlash and stiction, process and sample transportation delays, analyzer and wireless measurement update times, and PID execution time [2, 4, 6, 22, 24]. Except for damping settings, signal filters, analyzer and wireless update times, and PID execution times, these lags and delays are generally unknown and variable. The key features in a PID offer the flexibility and capability to achieve the ultimate limit to loop performance despite the challenging characteristics of industrial processes [21, 23].

14.2 Challenges and Solutions

A myriad of options and techniques have been used to address industrial automation system limitations and process objectives. Before we look at specific solutions used in industry, we need to understand the practical and ultimate limits to PID performance for unmeasured load disturbances in industrial processes. The first subsection provides practical equations developed over the years to detail the important relationships between load performance and dynamics and tuning. This subsection also offers a new equation to show how an important metric for setpoint performance also depends upon dynamics and tuning. Since setpoint changes are exactly known unlike unmeasured disturbances many methods exist to circumvent the limitations imposed by tuning. Subsequent subsections discuss methods such as setpoint feedforward and smart bang-bang logic.

14.2.1 Practical and Ultimate Limits to PID Performance

Special algorithms can be designed to deal with measured load disturbances at the process input, setpoint changes, and disturbances at the process output (e.g., noise). Often neglected is the overriding requirement that controllers in industrial applications must be able to deal with unmeasured and unknown load disturbances at the process input. Fortunately, the PID controller excels at this load disturbance rejection. An estimate of the current and best possible load rejection as a function of the process and automation system dynamics and controller tuning provides the information on what can be done to improve plant design and tuning. A simple set of equations can be developed that estimates the integrated error and peak error for a step change in a load disturbance. The value is more in helping guide decisions on improvements rather than predicting actual errors because of the uncertainty of the size and speed of load disturbances and the nonlinear and non-stationary nature of industrial processes. The equations are simple enough to provide key insights as the relative effects of the controller gain and integral time and the first order plus deadtime approximation (FOPDT) of the process and automation system dynamics. In the FOPDT model, a fraction of each of the time constants smaller than the largest time constant is taken as equivalent deadtime and summed with the pure deadtimes to become the total loop deadtime (θ_o) termed a process deadtime (θ_p) in the literature. The fraction of the small time constants not taken as deadtime is summed with the largest time constant to become the open loop time constant (τ_o). While the equations for tuning and estimation of errors is based on the open loop time constant, we will assume the largest time constant is in the process so we have the more common term of process time constant (τ_p) seen in the literature. In reality fast loops, such as liquid flow and pressure, have a time constant in the FOPDT model much larger than the flow response deadtime due to a transmitter damping setting and signal filter time constant. Similarly, the equations seen in the literature use a process gain (K_p) rather than the open loop gain (K_o) that is the product of the final control element, process, and measurement gain. For improving dynamics, a distinction of the location of nonlinearities, deadtime, and the largest time constant are important. By avoiding the categorization of dynamics as being solely in the process, a better understanding of the effect of the final control element size, installed characteristic, stick-slip, and backlash, the effect of measurement noise, lag, delay, calibration span, and the effect of PID filter and execution time is possible. The nomenclature used in the quantification of these effects is defined at the beginning of the chapter.

Since a controller cannot compensate for an unmeasured load disturbance before the loop deadtime, the peak error (E_x) (maximum error for a disturbance) is the excursion of the first order response to the step disturbance (E_o) based on the open loop time constant for a time duration of the loop deadtime (14.1) [2]. The open loop error is the final error seen at the PID from an unmeasured load disturbance if the PID was in manual. The terms "open loop" and "closed loop" are used for a response without and with feedback correction, respectively.

$$E_x = \left[1 - e^{-\frac{\theta_o}{\tau_o}}\right] E_o. \tag{14.1}$$

If the total loop deadtime is much larger the open loop time constant, then the peak error is basically the open loop error. If the deadtime was less than the time constant, then (14.1) can be simplified to (14.2), eliminating the exponential term [2, 4]

$$E_x = \frac{\theta_o}{(\theta_o + \tau_o)} E_o. \tag{14.2}$$

The minimum integrated error (E_i) can be approximated as the area of two right triangles with the altitude equal to the peak error and the base equal to the deadtime. Taking the area of each triangle as $\frac{1}{2}$ the base multiplied by the altitude we obtain (14.3) where the integrated error is simply the peak error multiplied by the deadtime and consequently proportional to the deadtime squared [2, 4]

$$E_i = \frac{\theta_o^2}{(\theta_o + \tau_o)} E_o. \tag{14.3}$$

Equations (14.2) and (14.3) are for the minimum possible errors determined by the open loop process and system automation system dynamics. It is not possible to do better than what is permitted by the dynamics. Thus, these are the ultimate limits to loop performance for unmeasured load disturbances. What is achieved in feedback control depends upon the tuning. In practice, controllers are not tuned aggressively enough to achieve the ultimate limit because the response tends to be too oscillatory especially for large setpoint changes and the controller lacks robustness. A 25% increase in loop deadtime or open loop gain or 25% decrease in the open loop time constant can result in oscillations that do not sufficiently decay. We can develop the equations that set the practical limit in terms of controller tuning settings from the equations for the ultimate limit based on open loop dynamics. We will also see that we can independently arrive at the same equation for the integrated error from the response of the PI algorithm to a step disturbance.

If we divide through by the deadtime term in (14.2), we have (14.4) where the peak error depends upon the ratio of the open loop time constant to total loop deadtime

$$E_x = \frac{1}{(1 + \tau_o/\theta_o)} E_o. \tag{14.4}$$

Most tuning methods for maximum disturbance rejection use a controller gain (K_c) that is proportional to the ratio of the open loop time constant to total loop deadtime and inversely proportional to the open loop gain (14.5) [2, 4, 6]

$$K_c = \frac{\tau_o}{\theta_o K_o}. \tag{14.5}$$

If we solve for the open loop time constant to total deadtime ratio, we see that this ratio is simply the product of the controller gain and open loop gain ($K_c K_o$). If we substitute the product for the ratio in (14.4), we have (14.6), which is the practical limit to the peak error [2, 4]. Peter Harriott developed the same form of the equation but with a numerator of 1.5 for the peak error from a proportional only controller tuned for quarter amplitude decaying response [5]

$$E_x = \frac{1}{(1 + K_c K_o)} E_o. \tag{14.6}$$

For time constant-to-deadtime ratios that are much larger than one, which is the case for pressure and temperature control of vessels and columns, the product of the controller gain and open loop gain is much greater than one leading to the peak error being simply inversely proportional to the product. Since the controller gain used in practice is about half of the gain for maximum disturbance rejection, we end up with (14.7) for the peak error

$$E_x = \frac{2}{K_c K_o} E_o. \tag{14.7}$$

Equation (14.7) corresponds to a peak error reached in about two deadtimes. If we approximate the integrated error as the area of two right triangles each with a base equal to two deadtimes and consider the integral time (T_i) setting as being 4 deadtimes, we end up with (14.8) for the integrated error [2, 4, 6]

$$E_i = \frac{T_i}{K_c K_o} E_o. \tag{14.8}$$

We can derive (14.8) from the equation for a PI controller's response to an unmeasured load disturbance. The change in controller output from time $t1$ to time $t2$ is the sum of the contribution from the proportional mode and the integral mode (14.9a). The module execution time (Δt) is added to the reset or integral time (T_i) to show the effect of how the integral mode is implemented in digital controllers. An integral time of zero ends up as a minimum integral time equal to the execution time so there is not a zero in the denominator for the integral mode. For analog controllers, the execution time is effectively zero [4, 6]

$$\mathrm{CO}_{t2} - \mathrm{CO}_{t1} = K_c(E_{t2} - E_{t1}) + \left[\frac{K_c}{(T_i + \Delta t)}\right] \int_{t1}^{t2} E_t \Delta t. \tag{14.9a}$$

The errors before the disturbance (E_{t1}) and after the controller has completely compensated for the disturbance (E_{t2}) are zero ($E_{t1} = E_{t2} = 0$). Therefore, the long term effect of the proportional mode, which is first term in (14.9a), is zero. Equation (14.9a) reduces to (14.9b) [4, 6]

$$\Delta\mathrm{CO} = \left[\frac{K_c}{(T_i + \Delta t)}\right] \int_{t1}^{t2} E_t \Delta t. \tag{14.9b}$$

The integrated error is the integral term in (14.9b), giving (14.9c). For overdamped response the integrated error and the integrated absolute error (IAE) are identical

$$E_i = \int_{t1}^{t2} E_t \Delta t. \tag{14.9c}$$

If we substitute (14.9c) into (14.9b), we have (14.9d)

$$\Delta\mathrm{CO} = \left[\frac{K_c}{(T_i + \Delta t)}\right] E_i. \tag{14.9d}$$

The change in controller (ΔCO) multiplied by the open loop gain (K_o) must equal the open loop error (E_o) for the effect of the disturbance to be eliminated. We can express this requirement as the change in output being equal to the open loop error divided by the open loop gain (14.9e) [4, 6]

$$\Delta\mathrm{CO} = \frac{E_o}{K_o}. \tag{14.9e}$$

If we substitute (14.9e) into (14.9d) and solve for the integrated error, we end up with (14.9f), which is the same as (14.8) except for the addition of the execution time interval for the digital implementation of the PI algorithm [6]

$$E_i = \left[\frac{(T_i + \Delta t)}{K_o K_c}\right] E_o. \tag{14.9f}$$

Recently, Greg Shinskey added a term to the numerator to include the effect of a signal filter time constant on the integrated error (14.10) [24, 30–32]. In Shinskey's presentation of the equation, the change in controller output rather than the open loop error is used, which eliminates the open loop gain in the denominator. Equation (14.10) is applicable regardless of tuning settings. The additional equivalent deadtime from the filter time and execution time interval may necessitate a decrease in controller gain and increase in integral time further degrading performance [4, 24]

$$E_i = \left[\frac{(T_i + \Delta t + \tau_f)}{K_o K_c}\right] E_o. \tag{14.10}$$

To summarize, in the process industry, automation system and process dynamics, and in particular the loop deadtime, set the ultimate limit to loop performance but controller tuning sets the practical limit for unmeasured disturbances. For example, a loop with a small deadtime will perform as badly as a loop with a large deadtime if the PID has sluggish tuning. On the other hand, a PID with fast tuning may have an excessive oscillatory response for increases in the loop deadtime or process gain. Equation (14.6) shows the practical limit to the peak error (E_x) is inversely proportional to 1 plus the product of the PID gain (K_c) and the open loop gain (K_o) [2, 4, 6, 22, 24]. Equation (14.9f) indicates the integrated error (E_i) is proportional to the ratio of the PID integral time to gain (T_i/K_c) [2, 4, 6, 22, 24, 30–32, 34]. For small filters (τ_f) and PID execution time (ΔT_x), the controller gain is decreased and the integral time is increased based on the increase in loop deadtime. The filter and execution time can be added to the integral time for the integrated error to show the increase in the practical limit (14.10) [10]. For a deadtime much less than the open loop time constant, (14.2) reveals that the ultimate limit to the peak error depends upon the ratio of the total loop deadtime (θ_o) to the open loop time constant (τ_o) [2, 4, 6, 22, 24]. Equation (14.3) indicate that the integrated error depends upon the ratio of the loop deadtime squared to open loop time constant. A PID controller tuned for maximum disturbance rejection has a controller gain proportional to the ratio of the largest open loop time constant to loop deadtime (τ_o/θ_o), and an integral time proportional to the loop deadtime [2, 4, 6, 22, 24, 30–32, 34]. Note that the controller tuning depends upon the largest open loop time constant and not the process time constant. If the largest time constant is in the measurement path, the observed peak error in the measurement predicted by (14.2) will be smaller than the actual peak error in the process because of the signal filtering effect of the measurement time constant.

The peak error is important for preventing: shutdowns from reaching trip settings of safety instrumentation systems (SIS), environmental emissions and process losses from reaching the relief settings of rupture discs and relief valves, off-spec paper sheet and plastic web from exceeding permissible variation in thickness and clarity, compressor shutdowns from crossing surge curve, and recordable incidents by exceeding environmental limits [22].

The integrated error is a good indicator of the quantity of liquid product off-spec in equipment with back mixing. In these volumes, positive and negative fluctuations in concentration are averaged out unless irreversible reactions are occurring [22].

An important emerging consideration is the realization that initial open loop response in the FOPDT approximation of a self-regulating is a ramp seen in the response of an integrating process such as level and batch temperature [2, 4, 6, 9, 24]. The ramp is more persistent in a self-regulating process with a large open loop time constant. The process is termed "near integrating" or "pseudo integrating". An equivalent integrating process gain (K_i) can be approximated as the open loop gain divided by the open loop time constant (14.11). For processes where the open loop time constant is more than ten times larger than the deadtime, the identification of this near integrator gain in 3 deadtimes can reduce the time required for process identification by more than 90% compared to those techniques that go to the 98% response time. The process variable (PV) is passed through a deadtime block to create an old PV that is subtracted from the new PV to create a ΔPV and then an integrating gain by dividing by the deadtime and the change in controller output. The maximum of a continuous train of these "near integrating" process gains updated every execution of the PID module can be used for tuning controllers on all types of processes

$$K_i = \frac{K_o}{\tau_o}. \tag{14.11}$$

If we substitute the near integrating gain for the open loop gain to time constant ratio in (14.5), we have (14.12). Recently, this method was found to even work on processes where the deadtime was greater than the time constant. To provide a smoother response, less setpoint overshoot, and more robust settings, the controller gain in both (14.5) and (14.12) is cut in half

$$K_c = \frac{1}{\theta_o K_i}. \tag{14.12}$$

The optimum integral time depends upon the type of process. The integral time ranges from about 4 times the deadtime for integrating and "near integrating" processes to one half the deadtime for severely deadtime dominant process ($\theta_o \gg \tau_o$). Equation (14.13) provides a reasonable curve fit to the required relationship for self-regulating processes. For a deadtime less much less than the time constant ($\theta_o < 0.1\tau_o$), the ultimate period is about 4 times the deadtime and the denominator is about 1, giving an integral time that is about 4 times the deadtime. For a deadtime much greater than the time constant ($\theta_o > 10\tau_o$), the ultimate period is about 2 times the deadtime and the denominator is about 4, giving an integral time that is $\frac{1}{2}$ of the deadtime.

For self-regulating processes:

$$T_i = \frac{T_u}{\min(4, 3(\frac{4\theta_o}{T_u} - 1)^2 + 1)}. \tag{14.13}$$

For a deadtime dominant process, the combination of (14.13) for integral time and (14.5) for controller gain results in almost an integral-only controller. Since the controller gain is so low, this process is a candidate for setpoint feedforward to reduce the setpoint response rise time.

For an integrating process, the product of the controller gain and integral time must be greater than twice the inverse of the integrating process gain to prevent slowly decaying oscillations from the integral mode dominating the proportional mode [9, 24]. If the user is confident in the knowledge of the integrating process gain, this relationship can be used to find the integral time (14.14a). Since the maximum controller gain allowable on many level and batch temperature loops is greater than 100 and the actual controller gain used is often less than 10, the integral time must be increased to prevent the slow rolling oscillations. Consequently, while an integral time of 4 deadtimes is possible for an integrating process, in practice an integral time of 40 deadtimes is more appropriate because the maximum controller gain is beyond the user's comfort level (14.14b).

To prevent slowly decaying oscillations integrating processes from excessive integral action, the following is used:

$$T_i > \frac{2}{K_c K_i}. \tag{14.14a}$$

The positive feedback in the runaway processes necessitates an integral time ten times larger than the integral time for a "near integrating" self-regulating process. The integral time should be 40 deadtimes or larger for a runaway process (14.14b). Some highly exothermic polymerization batch reactors have gone to proportional plus derivative control to avoid the problem of a user setting too small of an integral time.

For integrating processes with controller gains less than 10 times the maximum permissible controller gain and for runaway processes, one takes:

$$T_i = 40\theta_o. \tag{14.14b}$$

Too small of a controller gain or too large of a controller gain can cause a runaway reaction. There is a window of allowable controller gains for positive feedback processes [6, 24]. Any changes in tuning settings particularly for runaway reactions must be closely monitored.

Common metrics for a setpoint response are rise time (time to reach setpoint), overshoot (maximum error after crossing setpoint), and settling time (time settle out within a specified band around the setpoint). The ultimate limit for rise time is proportional to the loop deadtime. The ultimate limit for overshoot and settling time is theoretically zero. The practical limit to rise time is similar to the practical limit for peak error for fast tuning settings but degrades to the relationship for the integrated error for sluggish tuning settings. Fortunately, there are many features that can be used to readily help achieve the ultimate limit to the rise time. The practical limits for overshoot and settling time depend upon a balance between the contributions from the integral and proportional modes. In general, the controller gain for maximum disturbance rejection can be used to minimize rise time and the integral time can be increased to minimize overshoot and settling time [22, 24].

The minimum rise time (T_r) can be approximated as the change in setpoint (ΔSP) divided by the maximum rate of change of the process variable. For an integrating or "near integrating" process, the maximum PV ramp rate is the integrating process (K_i) gain multiplied by the change in controller output as detailed in the denominator of (14.15a). If the step change in controller output from the proportional mode for a structure of proportional action on error is less than the maximum available output change (difference between current output and output limit), (14.15a) simplifies to (14.15b) for feedback control. The output change must be corrected for methods used to make the setpoint response faster. For setpoint feedforward, the step change in output is a combination of the feedforward and feedback action. For smart bang-bang logic, the step output change is the maximum available output change

$$T_r = \frac{\Delta \mathrm{SP}}{K_i \min(|\Delta \mathrm{CO}_{\max}|, (K_c + K_{ff})\Delta \mathrm{SP})} + \theta_o. \tag{14.15a}$$

For a maximum available output change larger than the step from the proportional mode ($|\Delta \mathrm{CO}_{\max}| > K_c \Delta \mathrm{SP}$), the change in setpoint in the numerator and denominator cancel out, yielding a simpler equation without feedforward:

$$T_r = \frac{1}{(K_i K_c)} + \theta_o. \tag{14.15b}$$

For the "near integrating" process response seen in vessel and column temperature loops where the process time constant is significantly larger than the total loop deadtime, the integrating gain is the open loop gain (K_o) divided by the open loop time constant (τ_o) and (14.15b) becomes (14.15c) [6, 11, 24]

$$T_r = \frac{\tau_o}{(K_o K_c)} + \theta_o. \tag{14.15c}$$

The practical and ultimate limit to loop performance can be reconciled by realizing that there is an implied deadtime (θ_i) from the tuning. Equation (14.16) shows the implied deadtime that can be approximated as the original deadtime (θ_o) plus Lambda (λ) multiplied by a factor that is 0.5 [15, 17, 21, 23]. Lambda is the closed loop time constant for a setpoint change. For a PID tuned for maximum disturbance rejection, Lambda is set equal to the original deadtime. The implied deadtime is then equal to the original deadtime [15, 17, 22, 24]

$$\theta_i = 0.5(\lambda + \theta_o). \tag{14.16}$$

The peak and integrated errors for unmeasured step disturbances represent the worst case. Step disturbances originate from manual actions, safety, switches, and sequential operations. If discrete actions (e.g., the opening and closing of on-off valves and the starting and stopping of pumps) are replaced by control loops with modulated final control elements (throttling valves and variable speed drives) or are attenuated by intervening volumes, the step disturbances are smoothed. The attenuated load disturbance has a time constant (τ_L) that is the residence time of the volume or closed loop time constant of the upstream control loop. To include the effect of a load time constant, the process excursion in the first deadtime, which is the key time for determining minimum peak error, can be computed by (14.17). The open loop error (E_o) in the equations for peak and integrated error can be replaced with the load disturbance (E_L), that is, the open loop error multiplied by the exponential response of the disturbance in one deadtime The effect is mitigated by a reset time that is slow relative to the disturbance time constant [22, 24]

$$E_L = \left(1 - e^{-\theta_o/\tau_L}\right) E_o. \tag{14.17}$$

PID controllers tuned too fast can introduce process variability from an oscillatory response, PID controllers tuned too slow can make a loop with good dynamics perform as badly as a loop with poor dynamics. In other words, money invested to reduce process deadtime or to get faster measurements and valves is wasted unless the PID controller tuning is commensurate with the speed of the process so that the practical limit approaches the ultimate limit to loop performance.

In some cases, slower tuning, longer wireless update times, and a PID enhanced for wireless will reduce the oscillations from feedforward timing errors and interaction between loops. Also, in cases of blend control, all of the flow loops may be forced through tuning to be as slow as the flow loop with the largest deadtime to provide a coordination of flows that leads to greater product consistency.

Since industrial processes have valve, process, and measurement dynamics that vary with time, operating point, and step size, it is important to have automated methods of tuning.

14.2.2 On-demand and Adaptive Tuning

On-Demand and Adaptive Tuning integrated into the PID function block in a DCS enables the use of PID tuning that achieves the ultimate performance limit. The relay method by Karl Åström provides a straightforward On-Demand Tuner [1, 2, 4, 6, 11]. A user-selected step change is injected into the PID output initially and any time the process variable reverses direction and crosses the setpoint and the corresponding noise band. The controller action is used to determine if the reversal in the process variable is in the correct direction to drive the process variable back to setpoint. The ultimate period (T_u) is the oscillation period. Equation (14.18) is used to compute the ultimate gain (K_u) from the PID output step size (d) and the process variable amplitude (a) corrected for the noise band (n). Figure 14.1 shows the relay oscillation method with a large change in the process variable (PV) for illustrative purposes. For processes with large time constants, the PV amplitude (a) is so small, the oscillation is barely perceptible and the oscillation period is about 4 deadtimes. Since the more important PID loops, such as temperature, have a large process time constant, the auto tuner provides a test that is less disruptive and faster

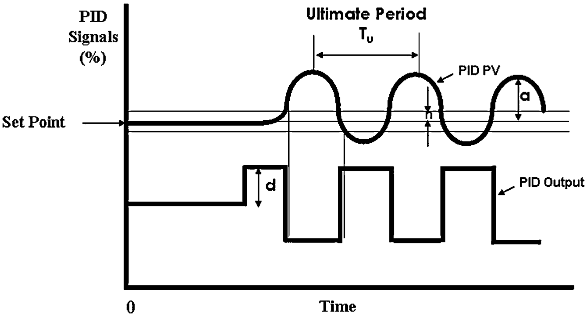


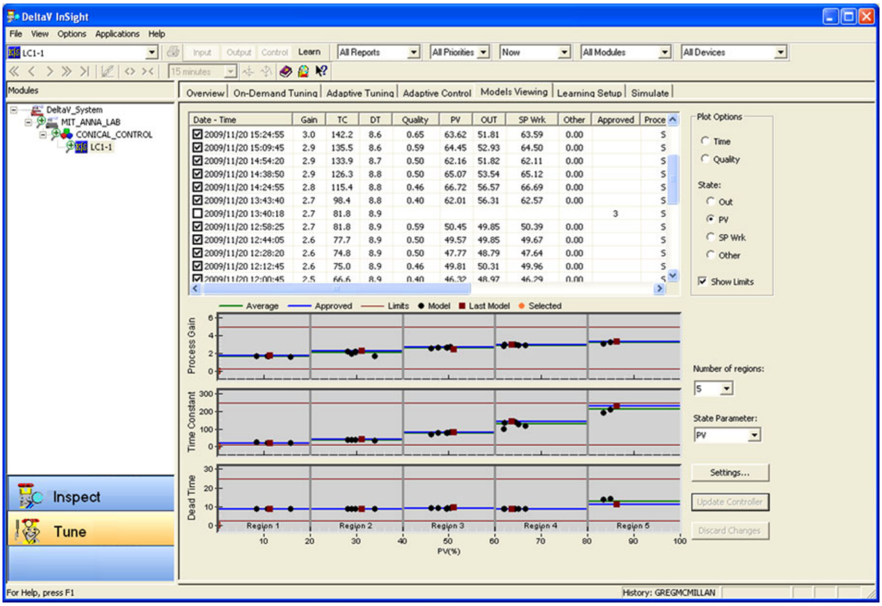
Fig. 14.1 Relay oscillation method offers fast tuning test [23]

than an open loop test that is waiting to reach a new steady state to identify the process time constant. The time constant identified in relay oscillation method is not very accurate. Thus, when the relay oscillation method, tuning settings based on the ultimate period and ultimate gain are more accurate than those that require knowledge of the process time constant

$$K_u = \frac{4d}{\pi + \sqrt{a^2 - n^2}}. \tag{14.18}$$

The PID gain is the ultimate gain multiplied by a 0.25 factor [22, 24]. The PID integral time is the ultimate period multiplied by 1.0 factor for self-regulating and 10.0 for non-self-regulating processes [11, 22, 24]. The PID rate time is the ultimate period multiplied by 0.1 when derivative action is beneficial [11, 22, 24]. If the ultimate period is less than 3 times the deadtime, the rate time should be 0 since the loop is deadtime dominant (deadtime is significantly greater than the largest time constant in the loop) [11]. If the ultimate period is greater than 4 times the deadtime, rate time should be used to prevent a runaway since the process may have positive feedback and an unstable open loop response. These factors are generally in the direction to provide a non oscillatory PID response that is more robust (more resistance to excessive oscillations from changes in process dynamics). The Ziegler–Nichols factors were designed to provide a quarter amplitude response (amplitude of each succeeding oscillation is $\frac{1}{4}$ the amplitude of the last oscillation). Most publications on tuning based on the ultimate period and ultimate gain use the Ziegler–Nichols factors leading to improper conclusions on smoothness and robustness of the tuning method [11].

Adaptive tuners use a more advanced method to identify process dynamics without relay oscillations. Significant manual and remote output changes and setpoint changes trigger the search for the dynamic parameters for a first order plus deadtime approximation (process gain, deadtime, and time constant) that provides a model's response that matches the process response. A particular adaptive tuner computes the integrated squared error (ISE) between the model and the process output for changes in each of three model parameters from the last best value. Exploring all combinations of three values (low, middle, and high) for three parameters, results in 27 models. The correction in each model parameter is interpolated by the application of weighting factors that are based on the ISE for each model normalized



Date - Time	Gain	TC	DT	Quality	PV	OUT	SP Wrk	Other	Approved	Proce
☑ 2009/11/20 15:24:55	3.0	142.2	8.6	0.65	63.62	51.81	63.59	0.00		S
☑ 2009/11/20 15:09:45	2.9	135.5	8.6	0.59	64.45	52.93	64.50	0.00		S
☑ 2009/11/20 14:54:20	2.9	133.9	8.7	0.50	62.16	51.82	62.11	0.00		S
☑ 2009/11/20 14:38:50	2.9	126.3	8.8	0.50	65.07	53.54	65.12	0.00		S
☑ 2009/11/20 14:24:55	2.8	115.4	8.8	0.46	66.72	56.57	66.69	0.00		S
☑ 2009/11/20 13:43:40	2.7	98.4	8.8	0.40	62.01	56.31	62.57	0.00		S
□ 2009/11/20 13:40:18	2.7	81.8	8.9						3	S
☑ 2009/11/20 12:58:25	2.7	81.8	8.9	0.59	50.45	49.85	50.39	0.00		S
☑ 2009/11/20 12:44:05	2.6	77.7	8.9	0.50	49.57	49.85	49.67	0.00		S
☑ 2009/11/20 12:28:20	2.6	74.8	8.9	0.50	47.77	48.79	47.64	0.00		S
☑ 2009/11/20 12:12:45	2.6	75.0	8.9	0.46	49.81	50.31	49.96	0.00		S
☑ 2009/11/20 12:00:45	2.5	66.6	8.9	0.40	46.32	48.97	46.29	0.00		S

Plot Options
○ Time
○ Quality

State:
○ Out
● PV
○ SP Wrk
○ Other

☑ Show Limits

Number of regions: 5

State Parameter: PV

Fig. 14.2 Models enable adaptive level control of conical tank [19]

to a total ISE for all the models over the period of interest. After the best values are computed for each parameter, they are assigned as the middle values for the next iteration. This model switching with interpolation and re-centering has been proven mathematically by the University of California, Santa Barbara to be equivalent to a least square identification that provides an optimum approach to the correct model [10, 36].

Adaptive tuners schedule tuning settings identified for regions defined by a user-selected variable. For valves with nonlinear characteristics such as equal percentage, the variable for scheduling is the PID's output. For nonlinear processes, such as pH, the variable for scheduling is the PID's process variable. The scheduling provides preemptive correction of the tuning settings eliminating the delay in performance associated with the re-identification of settings as the PID moves into another region [10, 24, 29, 36]. For a gravity discharge conical tank, adaptive tuning made the level setpoint response fast with a consistent settling time over the entire range of operation by increasing the process time constant as the cross-section area increased from bottom to top [29]. In this example, the gravity discharge flow makes the process self-regulating rather than integrating. Consequently, the nonlinearity of the change in cross-sectional area predominantly affects the process time constant rather than the process gain. Figure 14.2 shows the models automatically identified in five regions for scheduling tuning settings to account for the changes in cross-section with level.

Since an adaptive tuner uses current tuning settings to compute process dynamics as the starting point for its search, the number of tests required to get an adaptive

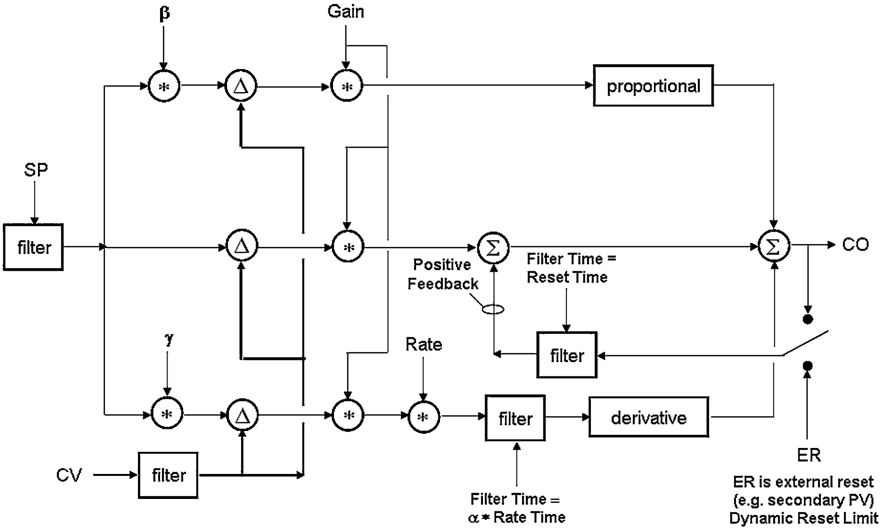


Fig. 14.3 Positive feedback integral mode enables key PID features (external reset, wireless enhancement, and deadtime compensation) [4, 23]

model with a high fidelity rating can be minimized by first running the On-Demand tuner with a requirement of just 2 or 3 cycles. Since the cycle period is on the average the ultimate period, the test is usually faster than an Adaptive Tuning test, especially for the overly conservative (sluggish) tuning commonly found in industrial PID controllers that have not been tuned by an automated method.

The step size in the output for On-Demand and Adaptive Tuning should be at least: 5 times the noise band, the trigger level of a wireless device, and the dead band and resolution-threshold sensitivity of the control valve [2]. Note that these step changes will not show the deadtime from wireless update times and valve backlash and stick-slip. For wireless devices, about half of the default update rate should be added to the identified deadtime [15, 22, 24, 30–32, 36].

14.2.3 Positive Feedback Implementation of Integral Mode

Instead of integrating the error, the feeding back of the controller output or external reset signal through a filter block and adding it to the contribution of the proportional and derivative modes creates an integral mode action where the filter time constant is the integral time setting [4, 24]. When the error is zero, the output of the filter block is simply the controller output or external reset signal and integral action stops. The positive feedback implementation illustrated in Fig. 14.3 enables several important PID options, such as dynamic reset limit, enhancement for wireless, and deadtime compensation. Figure 14.3 is for the ISA standard form for the PID controller. The eight structures commonly used in industrial processes are obtained by setting the setpoint weight factor β for the proportional and the setpoint weight factor γ for the integral mode in Fig. 14.3. If the factor is zero, a setpoint change does not affect the contribution to the output from respective mode (action is on PV only). If the factor is one, the full effect of a setpoint change is included (full action is on error). A factor between zero and one provides the ability to include but moderate the effect of a setpoint change (balanced action on setpoint change and PV change). In this figure, the multiplication symbol "*" in a circle is used to denote the multiplication by the β or γ weight factor.

The eight PID structures commonly used in industrial processes are:

1. PID action on error ($\beta = 1, \gamma = 1$)
2. PI action on error, D action on PV ($\beta = 1, \gamma = 0$)
3. I action on error, PD action on PV ($\beta = 0, \gamma = 0$)
4. PD action on error ($\beta = 1, \gamma = 1$)(no I action)
5. P action on error, D action on PV ($\beta = 1, \gamma = 0$)(no I action)
6. ID action on error ($\gamma = 1$) (no P action)
7. I action on error, D action on PV ($\gamma = 0$) (no P action)
8. Two degrees of freedom controller (β and γ adjustable 0 to 1)

β and γ are setpoint multiplication factors for the proportional and derivative modes, respectively, to determine how much proportional and derivative action occurs on setpoint changes. These factors do not affect the ability of the PID to reject disturbances. For the fastest possible setpoint response, structures 1 and 2 are used. If preventing overshoot is more important than minimizing rise time, structure 3 is used. If the ability to customize the balance between fast rise time and minimum overshoot for a setpoint response is needed, structure 8 is used. This structure also offers the ability to achieve both good load and setpoint responses.

14.2.4 Dynamic Reset Limit (External Reset)

When an external signal is used as the input to a "Filter" block in the positive feedback implementation of the integral mode, the integral action will not drive the controller output faster than the external reset signal is changing. This capability is particularly important for slow final control elements (large valves and variable frequency drives), cascade control, and override control.

If the external reset signal is the actual valve position or variable frequency drive (VFD) speed, the PID controller output will not ramp faster than the valve or VFD can respond [24]. Control valves and dampers have a slewing rate that increases with actuator size and stroke length. Damper slewing rate is particularly slow due to the need to prevent positive feedback from negative torque requirement. VFDs have velocity limiting of the command signal to prevent overloading the motor. If the external reset signal is the secondary loop process variable (PV) for cascade control, the primary PID cannot ramp the setpoint of the secondary PID faster than the secondary PID PV can respond. This capability is important for inherently preventing severe oscillations from breaking out for large setpoint changes or large disturbances [23, 33]. The use of the selected PID output as an external reset signal for override control also inherently prevents the unselected PID controllers from ramping off-scale. PID algorithms without the positive feedback implementation of integral action, add a “Filter” block to the external reset signal with a filter time equal to the PID reset time to prevent the ramping off-scale of the unselected PID output. The dynamic reset limit is a key feature that enables the development of an enhancement of the PID for wireless measurements that also has the ability to eliminate oscillations from threshold sensitivity and resolution limits and feedforward timing errors [20, 21, 24].

The dynamic reset limit can open opportunities important for sustainable manufacturing and in particular abnormal situation management and optimization. If a setpoint velocity limit is set in the analog output block, the dynamic reset limit prevents the PID from going faster than the velocity limit. The PID can achieve a slow approach to an optimum and a fast recovery upon encroachment of a constraint such as encountered in the prevention of compressor surge, exothermic reactor runaway, RCRA pH violation, and Bioreactor biomass starvation. Previously, an open loop back-up (kicker) has been used for these applications because the tuning of the controller for drastically different speeds of actuation is problematic. The dynamic reset limit option eliminates the need to tune the controller based on direction and the concern about the exact value of the velocity limit. The tuning is set for the fastest recovery. The velocity limit is adjusted for the slowest approach to the optimum.

There are many more examples where an intelligent adaptation of the speed of actuation of the final control element or secondary loop could be beneficial. In general, you want to approach optimums slowly to minimize disruption but as you operate close to the edge, you depend upon a fast recovery to prevent going over the edge. With compressor surge control the edge is literally a cliff. While other applications might not be as dramatic, the technique opens a wide spectrum of PID techniques for sustainable manufacturing, which in its broadest definition includes efficiency, flexibility, operability, maintainability, safety, and profitability [35].

14.2.5 Enhancements for Wireless

Wireless measurement devices have a “default update rate” (time interval for periodic reporting) and a “trigger level” (threshold sensitivity limit for exception reporting) set as large as possible to conserve battery life. The integral mode in the traditional PID will continue to ramp while the PID is waiting for an updated measurement from a wireless device. Also, when an update is received, the traditional PID considers the entire change to have occurred within the PID execution time interval (ΔT_x). If derivative mode is used, the rate of change of the measurement is the difference between the new and old measurement divided by the PID execution time interval. The result is a spike in the controller output.

The non-continuous update scenario occurs for many applications besides wireless devices. During the time when a measurement is not updated due to a failure,

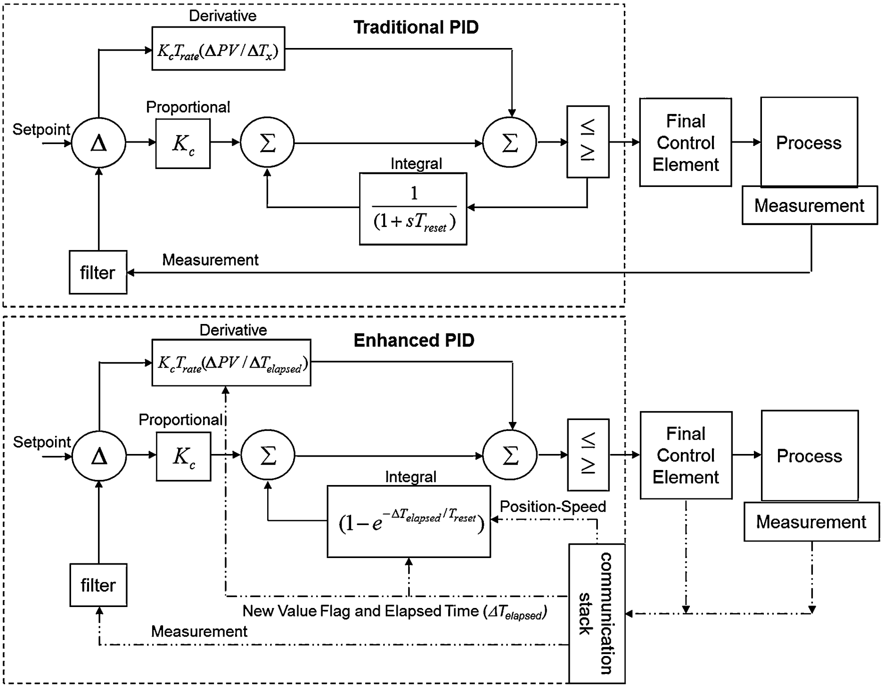


Fig. 14.4 Enhancements of PID for wireless prevent the ramping from integral action and the spikes from derivative action for discontinuous updates [21]

resolution limit, threshold sensitivity limit, or backlash, the PID output continues to ramp from the integral mode. Failures, resolution limits, and threshold sensitivity limits can originate in an analyzer, sensor, transmitter, communication system, or control valve. Analyzers also have a time interval between updates determined by the sample time and cycle time.

The enhanced PID for wireless executes the PID algorithm as fast as wired devices. A change in setpoint, feedforward signal, and remote output translates immediately (within PID execution time interval) to a change in PID output. However, integral action does not make a change in the output until there is an update. When an update occurs, the elapsed time between the updates is used in an exponential calculation that mimics the action of the filter block in the positive feedback implementation of integral action. If derivative action is used, the elapsed time rather than the PID execution time interval is used to calculate the rate of change of the process variable. The integral and derivative calculations are executed only once upon a change in setpoint or measurement [21, 23, 24]. A threshold sensitivity setting is used to prevent an update from noise. Figure 14.4 compares a simplified block diagram of the traditional PID to the enhanced PID.

A traditional PID will have to be detuned to prevent instability for a large increase in the time between updates. The enhanced PID will continue to be stable for even the longest update time interval. For a measurement update time interval larger than the process response time, the enhanced PID controller gain can be set equal to the inverse of the open loop gain (product of valve, process, and measurement gain) to provide a complete correction for setpoint change or update. Subsequent sections show the enhanced PID can suppress oscillations from a wide variety of sources. This reduction in variability results from the suspension of integral action and the wait in feedback correction till there is a more complete response [26]. To achieve these benefits, the user simply enables the enhanced PID option in the PID block, which automatically enables the dynamic reset limit option. No retuning is necessary to achieve a smooth response but if the update time is larger than the process response time the enhanced PID can be tuned with a much higher gain.

14.2.6 Deadtime Compensation

Adding a "Deadtime" block to the external reset of a positive feedback implementation of the integral mode can provide deadtime compensation equivalent to a Smith Predictor but with the advantage that the setup is simpler and the process variable is intact. In the positive feedback implementation of deadtime compensation, the user simply needs to set the deadtime parameter in the "Deadtime" block equal to the total loop deadtime. The dynamic reset limit option for the PID must be enabled so the external reset signal is used. The process gain and process time constant parameters used in a Smith Predictor are not necessary for this implementation of deadtime compensation. To get the benefit from the PID knowing the effect of deadtime, the integral time needs to be decreased and the controller gain needs to be increased [23]. Like the Smith Predictor, this deadtime compensator is more sensitive to an overestimate rather than an underestimate of the total loop deadtime. Normally, a PID will just become sluggish if an overestimate of the deadtime is used for the tuning settings. For PID controllers with deadtime compensation, high frequency oscillations will rapidly start for overestimates of the loop deadtime [11, 23, 24]. Thus, for robustness it is better to use a deadtime that is always less than the minimum loop deadtime often associated with high production rates.

In tests, the following myths about deadtime compensators were exposed [23]:

1. **Deadtime is eliminated from the loop.** The Smith predictor, which created a PV without deadtime, fools the controller into thinking there is no deadtime. However, for an unmeasured disturbance, the loop deadtime still causes a delay in terms of when the loop can see the disturbance and when the loop can enact a correction that arrives in the process at the same point as the disturbance. The ultimate limit to the peak error and integrated error for an unmeasured disturbance are still proportional to the deadtime, and deadtime squared, respectively.

2. **Control is faster for existing tuning settings.** The addition of deadtime compensation actually slows down the response for the existing tuning settings. Setpoint metrics, such as rise time, and load response metrics, such as peak error, will be adversely affected. Assuming the PID was tuned for a smooth stable response, the controller must be retuned for a faster response. For a PID already tuned for maximum disturbance rejection, the gain can be increased by 250%. For deadtime dominant systems where the total loop deadtime is much greater than the largest loop time constant (hopefully the process time constant), the reset time must also be decreased or there will be severe undershoot. If you decrease the reset time to its optimum, undershoot and overshoot are about equal. For a test case where the total loop deadtime to primary process time constant ratio was 10:1, the reset time could be decreased by a factor of 10. Further study is needed as to whether the ratio of the old to new reset time is comparable to the ratio of deadtime to time constant and whether half of the total loop deadtime or the module execution time (0.5 s) is the low limit to the reset time for an accurate deadtime estimate.
3. **Compensator works better for loops dominated by a large deadtime.** The reduction in rise time is greatest and the sensitivity to percent deadtime modeling error particularly for an overestimate of deadtime is least for the loop that was dominated by the process time constant. You could have a deadtime estimate that was 100% high before you would see a significant jagged response when the process time constant was much larger than the process deadtime. For a deadtime estimate that was 50% too low, some rounded oscillations developed for this loop. The loop simply degrades to the response that would occur from the high PID gain as the compensator deadtime is decreased to zero. While the magnitude of the error in deadtime seems small for the test case, you have to remember that for an industrial temperature control application, the loop deadtime and process time constant would be often at least 100 times larger. For a 400 second deadtime and 10,000 second process time constant, a compensator deadtime 200 seconds smaller or 400 seconds larger than actual would start to cause a problem. In contrast, the deadtime dominant loop developed a jagged response for a deadtime that was high or low by just 10%. This requirement is unreasonable in industrial processes. A small filter of 1 second on the input to the deadtime block can help.
4. **An underestimate of the deadtime leads to instability.** In tuning calculations for a conventional PID, a smaller than actual deadtime can cause an excessively oscillatory response. Contrary to the effect of deadtime on tuning calculations, a compensator deadtime smaller than actual deadtime will only cause instability if the controller is tuned aggressively after the deadtime compensator is added.
5. **An overestimate of the deadtime leads to sluggish response and greater stability.** In tuning calculations for a conventional PID, a larger than actual deadtime simply causes a slow smooth response. Contrary to the effect of deadtime on tuning calculations, a compensator deadtime greater than actual deadtime will cause jagged irregular oscillations.

14.2.7 Fast Setpoint Response

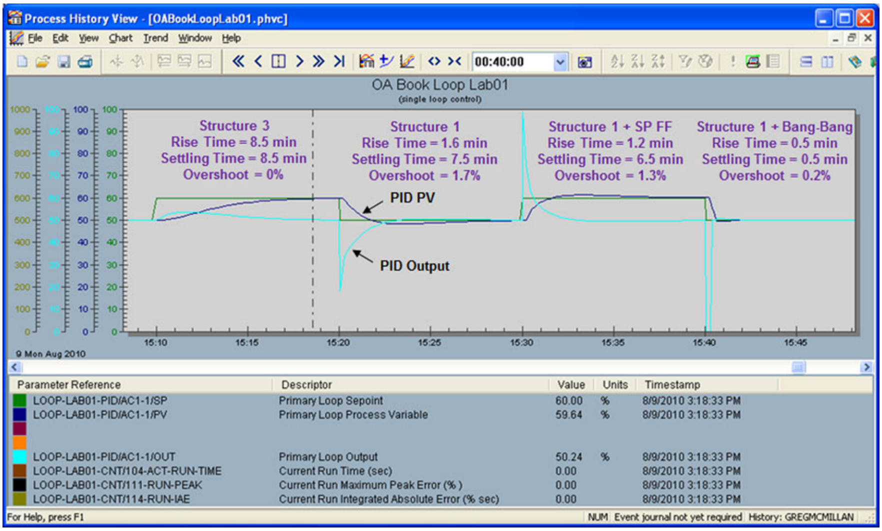
The rise time can be minimized by using the maximum possible controller gain and using a PID structure that has proportional and derivative action on setpoint changes. For slow loops, such as temperature and composition on vessels and columns, it is particularly important to make the approach to setpoint as fast as possible to minimize cycle time for batch operations and the startup and product grade transition time for continuous operations. Overdrive (driving the PID output beyond its final resting value) is essential for getting these slow loops to setpoint quickly. Fortunately, the deadtime to process time constant ratio for these loops is so small that large PID gains are permissible. Temperature loops often have significant secondary time constants from heat transfer surface and thermowell lags that would benefit from the use of rate action. A structure of PID action on error will provide a step in the PID output from the proportional mode and a bump in the PID output from the derivative mode for a step change in the setpoint. For this beneficial action to occur, the option SP track PV should be used and the controller must be in the auto mode when the setpoint change is made.

A setpoint feedforward signal added to the controller output can be useful if the change in PID output from the proportional mode is not sufficient. For small setpoint changes and for low controller gains, setpoint feedforward can reduce rise time. The feedforward action is the process action, which is the opposite of the control action, taking into account valve action. In other words, for a reverse control action, the feedforward action is direct, provided the valve action is inc-open or the analog output block, I/P, or positioner reverses the signal for an inc-close valve. For control loops where the loop deadtime is larger than the process time constant, the feedforward gain is approximately the inverse of the open loop gain minus the PID gain for enhanced PID structures with P action on error.

If the final resting value (FRV) is preloaded as an external reset signal during the rise time (dynamic reset limit option is enabled), the overshoot of the setpoint can be minimized [34]. If the FRV is not accurately known, the preload is prematurely disabled. Various versions of "batch controllers" since the 1960s have preloaded the integral mode.

The FRV can be captured from previous batches and is often the split range point. For changes in the setpoint of batch temperature and pH loops already in service that have an integrating response, the FRV is the PID output just before the setpoint change. For continuous process loops, the FRV is PID output just before the setpoint change plus the setpoint change multiplied by the inverse of the open loop gain for a self-regulating response [13, 24]. In all of these cases, the estimated FRV is based on an assumption that changes in process load and process disturbances are negligible during the setpoint change.

The fastest possible rise time with minimum overshoot and settling time is obtained by smart bang-bang logic that uses a simple prediction of when the process variable will reach setpoint. The PID output is stepped to its output limit to maximize the rate of approach to setpoint. When the projected PV equals the setpoint less a bias, the PID output is repositioned to the FRV. The PID output is held at the



	Parameter Reference	Descriptor	Value	Units	Timestamp
🟩	LOOP-LAB01-PID/AC1-1/SP	Primary Loop Sepoint	60.00	%	8/9/2010 3:18:33 PM
🟦	LOOP-LAB01-PID/AC1-1/PV	Primary Loop Process Variable	59.64	%	8/9/2010 3:18:33 PM
🟪					
🟧					
🟦	LOOP-LAB01-PID/AC1-1/OUT	Primary Loop Output	50.24	%	8/9/2010 3:18:33 PM
🟫	LOOP-LAB01-CNT/104-ACT-RUN-TIME	Current Run Time (sec)	0.00		8/9/2010 3:18:33 PM
⬛	LOOP-LAB01-CNT/111-RUN-PEAK	Current Run Maximum Peak Error (%)	0.00		8/9/2010 3:18:33 PM
🟨	LOOP-LAB01-CNT/114-RUN-IAE	Current Run Integrated Absolute Error (% sec)	0.00		8/9/2010 3:18:33 PM

For Help, press F1 | NUM | Event journal not yet required | History: GREGMCMILLAN

Fig. 14.5 Setpoint response shows rise time can be successively faster by a PID on error structure, setpoint feedforward, and a smart bang-bang logic [24]

FRV for one deadtime and then released for feedback control [13, 24]. A deadtime (DT) block must be used to compute the rate of change so that new values of the PV are seen immediately as a change in the rate of approach. If the total loop deadtime is used in the DT block, the projected PV is simply the current PV minus the output of the DT block (expected PV change over next deadtime) plus the current PV [24]. For self-regulating processes, such as flow with the loop deadtime approaching or greater than the largest process time constant, the logic is revised to step the PID output immediately to the FRV. Overdrive is not advisable in loops where the process time constant or the inverse of the integrating process gain is less than the deadtime because the potential for overshoot is too great for a slight underestimate of the deadtime. The total loop deadtime (without the effect of threshold sensitivity and resolution limits) can be identified at the start of the setpoint response.

Figure 14.5 shows that a PID structure of proportional and derivative action on error (structure 1) and setpoint feedforward can make the setpoint response faster, but the biggest improvement is achieved by smart bang-bang logic for an integrating process [24]. For self-regulating processes where the deadtime is larger than the process time constant, setpoint feedforward suffices.

14.2.8 Signal Linearization

Signal characterizers are used on the PID output to compensate for the nonlinear gain of control valves. The characterizer computes the percent stroke (X axis of the installed characteristic) from the percent flow (Y axis) based on the inherent trim characteristic (e.g., equal percentage), a given system resistance curve, and the net static head [6]. A pressure drop measurement across the valve eliminates the need to know the resistance curve and static head, but is rarely available. While characterizers are available in positioners, the location of the characterizer in the configuration provides better visibility and accessibility and offers the opportunity to display the valve signal to the operator after linearization as well as before linearization so no arises confusion in checking valve positions in the field.

Signal characterizers are used on the PID process variable (PV) to compensate for operating point nonlinearities of the process gain. The most common example is pH where the signal characterizer computes the percent reagent demand (X axis of the titration curve) from pH (Y axis) for a given composition of acids and bases in the feed. While the X-axis in actually the ratio of reagent flow to feed flow, for purposes of signal characterization the X axis is simply scaled 0 to 100% reagent demand [7, 24, 28]. Since the operator typically wants to enter an SP and see and trend the PV in pH units, both the SP and PV as pH and reagent demand are on the operator interface. A signal characterizer is used to convert the operator entered SP from pH to percent reagent demand. Since ensuring stability with faster tuning settings for higher process gains is the primary objective of the signal characterizer, the fidelity of the titration curve slope in the steepest regions is most important [7, 24].

Signal characterizers could also be used to compensate for the nonlinear process gain for temperature control in distillation columns. For example, signal characterizer based on a plot of tray or packing temperature versus reflux to feed ratio could be used to compute a linear reflux demand signal from temperature. The PID SP would be percent reflux demand from an operator-entered temperature setpoint. Pressure compensation could be applied to either the PID setpoint or input but not to both.

The use of signal characterization provides a finer resolution of process gain corrections than adaptive tuning but does not eliminate the need for adaptive tuning. The curves on which the signal characterization is based will change with operating conditions. The use of characterization and adaptive tuning is a synergistic relationship. If the signal characterization was perfect, the process gain would be one. The degree of difference between the identified process gain and a unity gain is a measure of the fidelity of the curve in various operating regions that could lead to a better curve.

14.2.9 Open Loop Backup (Kicker)

Some excursions are too fast and the consequences too severe or unstable to rely on feedback control alone. In these cases, an open loop backup is used to rapidly change the PID output to get the loop out of danger. The most common example is preventing compressor surge but other examples including preventing environmental violations and runaway reactions [24].

Upon detection of an excursion towards an unstable condition or environment violation, the open loop backup puts the PID in remote output (ROUT) and increments or decrements the PID output until the excursion is stopped. The open loop backup then waits for at least one deadtime before switching the PID back to its last mode bumplessly. The detections cited in the literature have been based on a PV trigger point, a PV rate of change, or a predicted PV similar to what is used by smart bang-bang logic for a fast setpoint response [24]. The PID execution must be less than 10% of the total deadtime. There may be some overuse of energy or reagent in the process to ensure personnel, property, and environmental protection for worst case conditions. In general, the avoidance of loss of damage and downtime more than pays for the short term costs.

The use of direction velocity limits in the analog output block in conjunction with the dynamic reset option in a PID with the positive feedback implementation of integral action can replace the need for some open loop backups. The velocity limits are set to provide a slow approach to the optimum and a fast getaway. For compressor surge control, the oscillations may be too fast and too severe for recovery by just PID feedback control.

14.2.10 Final Element Resolution, Threshold Sensitivity, and Backlash

Final control elements use the PID output to manipulate a flow, which is the predominant input to an industrial process. The most common final control element used in production units is a control valve. For utility systems, the final control element is often a damper due to large sizes and ducts. For agitators and pumps with large variable dynamic loads, a variable frequency drive is used to manipulate the speed as the final control element. For mixing, the agitating rate creates mixing flow rate that is called the agitator pumping rate.

All control valves have some degree of stick-slip and backlash. For sliding stem control valves with diaphragm actuators and digital positioners, the stick-slip and backlash can be less than 0.2%. For rotary valves originally designed for on-off tight shutoff rather than throttling service, the stick-slip and backlash can be as large as 10%. Furthermore, smart digital positioners are most likely measuring actuator shaft position in these cases which is not indicative of the actual position of the internal closure element (plug, disc, or ball) [9, 14, 18, 23–25]. The control valve with the least stick-slip and backlash is the sliding stem valve with low friction packing, diaphragm actuator, and a digital positioner. Figure 14.6 shows a control valve with a large backlash and some stick-slip near the closed position.

Stick-slip can be approximated as either a resolution limit or a threshold sensitivity limit [25]. For a resolution limit, the valve moves in steps which correspond to a slip equal to a stick. For a threshold sensitivity limit, slip is variable and results in the position of the internal closure element (plug, disc, or ball) momentarily catching up with the desired position set by the output of the PID. The source of stick-slip

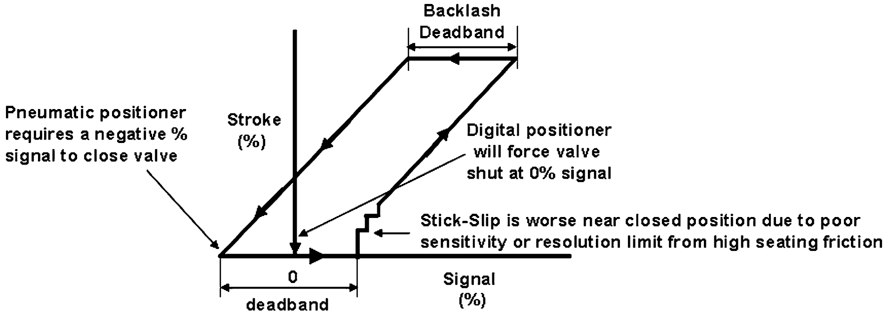


Fig. 14.6 Backlash causes a deadband on signal reversal and stick-slip causes a stair step response that is largest near the seat where the friction is greatest [24]

is friction of valve stem packing and friction of the seating and sealing of the internal closure element. The stick-slip is greatest near the closed position as the internal closure element approaches the closed position. Valves designed for tight shut-off tend to have the greatest stick-slip. For rotary valves, the sealing of the ball or disc can cause shaft windup where the shaft or stem twists but the ball or disc does not move. When the ball or disc breaks free, the valve jumps (slips) to a position that can exceed the desired position. This behavior can be approximated as a threshold sensitivity limit with overshoot [9, 14, 18].

Rack and pinion piston actuators and gear driven motor actuators have a resolution limit that corresponds to the teeth spacing. Double acting pistons have a threshold sensitivity limit that depends upon the pressure unbalance needed to overcome O-ring seal friction. Variable frequency drives (VFD) have a resolution limit set by the number of bits in the A/D input card for the command signal. Unfortunately, the standard VFD input card has only 8 bits which with 1 sign bit leaves only 7 bits for resolution of the signal (0.78% resolution) [9, 14, 18].

Backlash can be approximated by a deadband which is the amount of signal change required to reverse the direction of the valve stroke. Once the valve moves, the valve position catches up to the signal if there is no stick-slip. While technically valve deadband is defined for a full scale stroke, it can occur at any position for a signal reversal [9, 14, 18]. The primary sources of deadband are the translation of linear actuator to rotary motion and what might seem like insignificant gaps in linkages and looseness in actuator to valve stem connections [18]. Backlash is greatest for dampers and rotary valves with piston actuators originally designed for on-off action. Variable frequency drives introduce a backlash by a deadband setting used to prevent the drive from hunting or reacting to noise. Unfortunately, this deadband is often set too large because users are not aware of the detrimental impact of the backlash created.

Resolution and threshold sensitivity limits cause a limit cycle (constant amplitude sustained oscillation) in any process where the PID has integral action. Equation (14.19a) shows that the oscillation amplitude (A_o) is set by the open loop gain (K_o) and valve stick-slip (threshold sensitivity) (S_s) and is therefore independent of PID tuning. Equation (14.19b) shows the period depends upon the PID gain and integral time besides the open loop gain. The open loop gain is the product of the valve, process, and measurement gains as detailed by (14.19c). For high process gains, such as that encountered on the steep slopes of titration curves for pH control, stick-slip can cause unacceptable variability from the PID in automatic. The limit cycle oscillation period (T_o) will increase as the PID integral time is increased. For small degrees of stick-slip, noise, frequent disturbances, or historian data compression the limit cycle pattern may not be discernable [9, 18]

$$A_o = S_v K_o, \tag{14.19a}$$

$$T_o = 4T_i\big[1/(K_o K_c) - 1\big], \tag{14.19b}$$

$$K_o = K_v K_p K_m, \tag{14.19c}$$

$$K_m = \frac{100\%}{\mathrm{abs}(\mathrm{PV}_{100\%} - \mathrm{PV}_{0\%})}. \tag{14.19d}$$

If there are two or more integrators in the process and control system that affect the final element, deadband from backlash (B_v) will cause a limit cycle. The integrators can be the result of a cascade loop that has integral action in both the primary and secondary controllers or the result of an integrating process with a single controller with integral action. For an integrating process, such as level, the controller output is a sinusoidal oscillation and the process variable ramps with some rounding of the peaks (smoothed sawtooth) from backlash. The flow is a clipped oscillation. If there are no disturbances, the result is a limit cycle. Equation (14.20a) indicates the limit cycle amplitude is the deadband divided by the controller gain per. Equation (14.20b) shows the limit cycle period is proportional to integral time and is inversely related to controller gain. Detuning the controller (decreasing the controller gain) increases both limit cycle amplitude and period. Detuning also increase the deadtime for disturbances [9, 18]

$$A_o = B_v / K_c, \tag{14.20a}$$

$$T_o = 5T_i\big[1 - 2/\big(K_c^{0.5}\big)\big]. \tag{14.20b}$$

Deadband (backlash) limit cycles can be eliminated by suspending integral action in controllers so that the total number of integrators in the control loop including the process is one or less. Thus backlash limit cycles can be eliminated in processes with a single integrator, such as level, by turning off integral action in all PID controllers. For a self-regulating process, integral action would be permitted in one PID. Backlash limit cycles cannot be eliminated in a loop with two integrating processes. While processes with two integrators are rare, a possible example is a cascade loop where the primary loop is batch temperature and the secondary loop is vessel pressure. A runaway process, such as a highly exothermic reaction, counts as an integrator [24].

Threshold sensitivity or resolution (stick-slip) limit cycles can be eliminated by suspending integral action in controllers so that the total number of integrators in the control loop including the process is zero. Thus stick-slip limit cycles can be eliminated in self-regulating processes by turning off integral action in all PID controllers [20, 21, 24].

Many PID controllers have the option to turn off integral action when the process variable (PV) is within a band centered about the setpoint. The band is set equal to maximum amplitude of the limit cycle above and below setpoint. In some PID blocks, the parameter is called integral deadband or "IDEADBAND". Since the amount of backlash and stick-slip varies considerably with stroke and time due to changes in piston actuator O-rings, linkages, packing tightness, solids, pressure, and temperature, finding the right setting is problematic. Also, tuning algorithms generally do not include the effect of the integral deadband setting in tuning calculations. Finally, the response to load disturbances is delayed particularly for loops where the PID gain is low and hence the proportional action alone is insufficient.

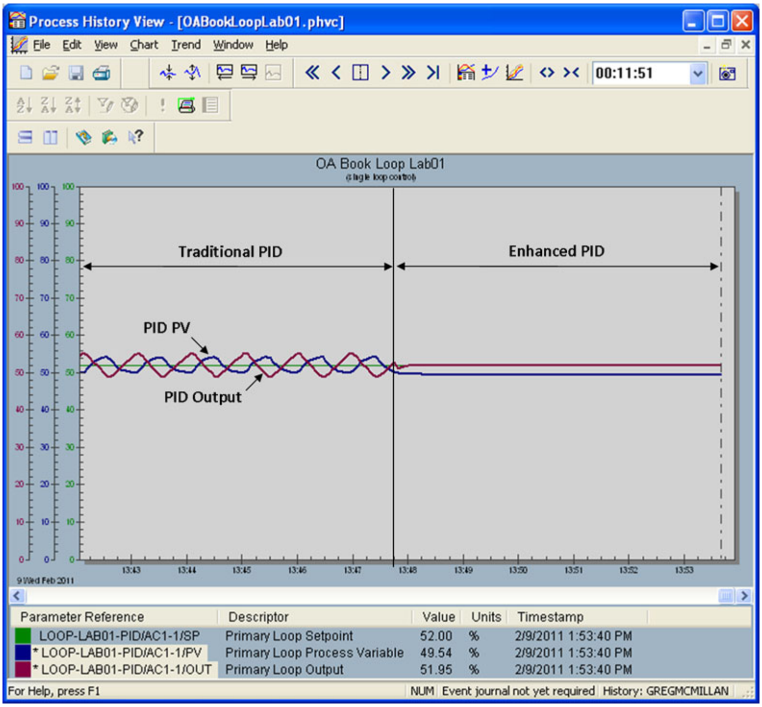
The enhanced PID inherently suspends integral action when there is no change in the PV. Hence, the wireless PID can automatically eliminate limit cycles from backlash in a single integrating process and limit cycles from stick-slip in a self-regulating process. No settings are required and normal tuning procedures can be used. Furthermore, the integral reaction to disturbances is not delayed. The use of wireless transmitters is not required. The user simply needs to enable the enhanced PID [20, 21, 24]. Figure 14.7 shows the enhanced PID inherently stops the limit cycle without the need of adjustments regardless of deadband or stick-slip size.

While the application described is for limit cycles originating in the valve or variable speed drive response, limit cycles can also occur due to threshold sensitivity or resolution limits in measurements. Pneumatic, mechanical, wireless, and analytical instruments have precision limits that can create appreciable limit cycles. Again, the enhanced PID inherently eliminates the limit cycles [25, 26].

14.2.11 Slow Final Control Element Response

Control valves have a velocity-limited exponential response. The velocity limit known as the slewing rate can be quite slow for large valves and dampers. Dampers are particularly slow with slewing rates of less than 1% per second because of large duct sizes and the need to prevent dynamic instability of the wafer or blade. Variable frequency drives also have a velocity limit or ramp rate to prevent motor overload. This ramp rate is often set too conservatively because of a lack of knowledge of the impact on process control.

If the PID controller output changes faster than the final control element can respond, the loop will break out into oscillations. The problem may only occur for large changes in setpoint or disturbances. The fact that the loop is fine for small changes and intermittently develops stability problems is confusing to operations. The solution is to use positive feedback implementation of integral action with the external reset signal for the dynamic reset limit being a readback of valve stroke or variable frequency drive speed [24, 33]. The use of auxiliary variables (second, third, or fourth process variable) in HART communication is generally not fast enough.



Parameter Reference	Descriptor	Value	Units	Timestamp
🟩 LOOP-LAB01-PID/AC1-1/SP	Primary Loop Setpoint	52.00	%	2/9/2011 1:53:40 PM
🟦 * LOOP-LAB01-PID/AC1-1/PV	Primary Loop Process Variable	49.54	%	2/9/2011 1:53:40 PM
🟥 * LOOP-LAB01-PID/AC1-1/OUT	Primary Loop Output	51.95	%	2/9/2011 1:53:40 PM

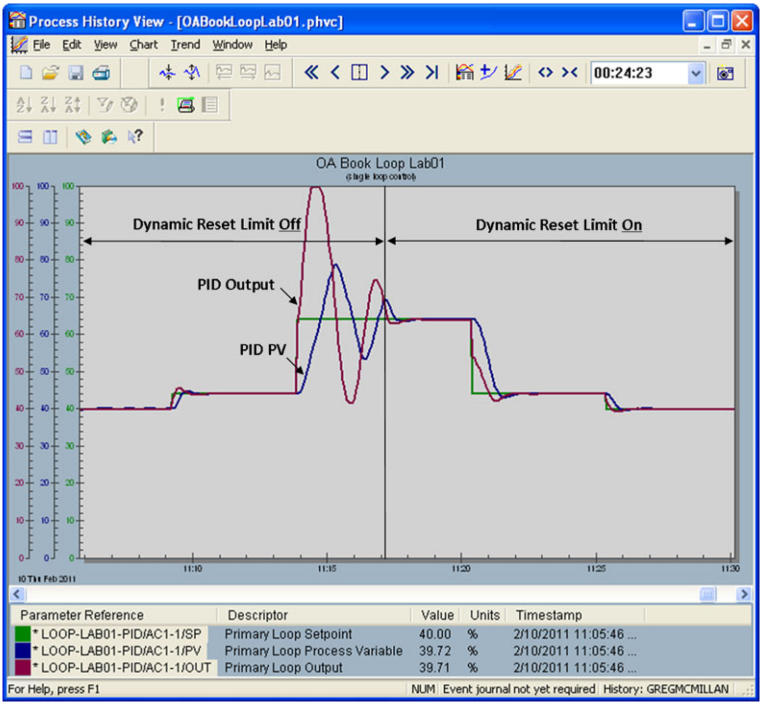
Fig. 14.7 Enhanced PID eliminates limit cycles from deadband and threshold sensitivity or resolution by suspending integral action when there is no update

The stroke or speed for external reset needs to be a primary process variable. Ideally, the valve should be speeded up by the use of volume boosters on the positioner output [18, 19, 24]. The constraints on VFD ramp rate must be more intelligently set. For most pumps and fans with properly sized motors, the speed ramp rate can be quite fast (e.g., 10% per second) without motor overload. Figure 14.8 shows that the burst of oscillations for large setpoint changes in a loop with a slow final control element are eliminated by dynamic reset limit option.

14.2.12 Slow Secondary Loop Response

If the secondary loop is slower than the primary loop, we have a problem similar to that discussed in the section on slow final control elements. For large changes in the setpoint or disturbance of the primary loop, oscillations develop. The solution is the same: dynamic reset limit should be used. In this case, the external reset is the process variable of the secondary loop.

Ideally, the secondary loop should be 4 times faster than the primary loop for maximum secondary disturbance rejection and minimal oscillations from interactions between the loops. If the secondary loop cannot be made faster, the primary



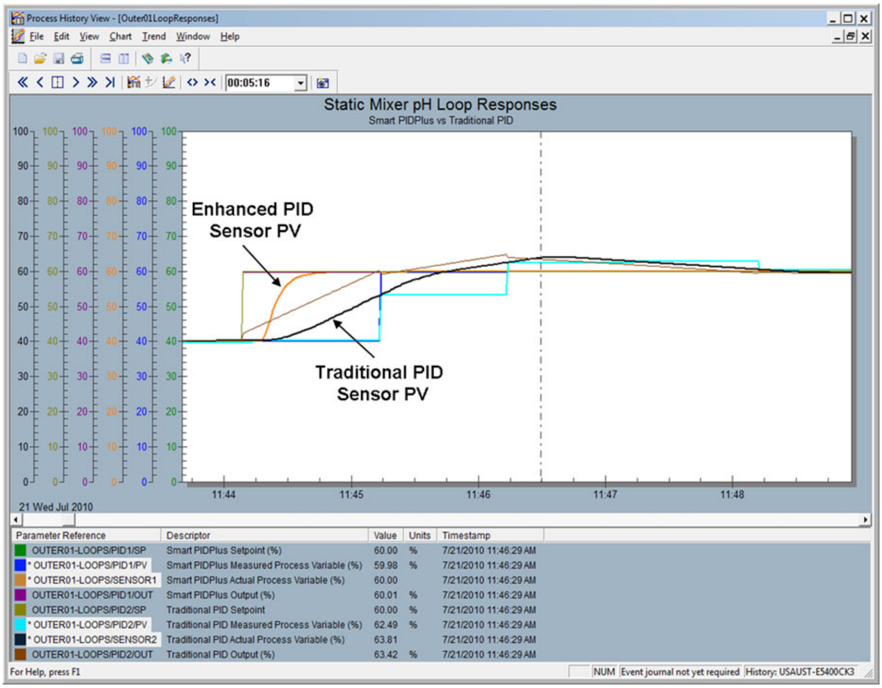
Parameter Reference	Descriptor	Value	Units	Timestamp
🟩 * LOOP-LAB01-PID/AC1-1/SP	Primary Loop Setpoint	40.00	%	2/10/2011 11:05:46 ...
🟦 * LOOP-LAB01-PID/AC1-1/PV	Primary Loop Process Variable	39.72	%	2/10/2011 11:05:46 ...
🟥 * LOOP-LAB01-PID/AC1-1/OUT	Primary Loop Output	39.71	%	2/10/2011 11:05:46 ...

Fig. 14.8 Dynamic reset limit option eliminates the oscillations for a slow valve or VFD that occur for large changes in a PID setpoint or load

loop may need to be slowed down. If wireless measurements with update rates larger than the 63% response time are used, the default update rate of the secondary measurement should be 4 times faster than the default update rate of the primary measurement per the cascade rule that the secondary loop is sufficiently faster than the primary loop to prevent interaction between the loops. For a cascade control of static mixer pH to reagent flow using the enhanced PID, a default update time of 60 seconds for pH and 16 seconds for flow provided exceptional pH setpoint control. The PID gains were set equal to the inverse of the open loop gain [20, 21]. Figure 14.9 shows that a cascade loop of pH to reagent flow on a static mixer with the enhanced PID can provide nearly perfect setpoint control.

14.2.13 Large Wireless Update Times

If the wireless update time is larger than the 63% process response time without the update delay, the use of the enhanced PID provides excellent setpoint control by allowing the use of a PID gain that is the inverse of the open loop gain. The enhanced PID is stable without retuning as the wireless default update time is increased to


Process History View - [Outer01LoopResponses]
File Edit View Chart Trend Window Help
00:05:16
Static Mixer pH Loop Responses
Smart PIDPlus vs Traditional PID
100
90
80
70
60
50
40
30
20
10
0
Enhanced PID
Sensor PV
Traditional PID
Sensor PV
11:44
11:45
11:46
11:47
11:48
21 Wed Jul 2010
NUM
Event journal not yet required
History: USAUST-E5400CK3

Parameter Reference	Descriptor	Value	Units	Timestamp
🟩 OUTER01-LOOPS/PID1/SP	Smart PIDPlus Setpoint (%)	60.00	%	7/21/2010 11:46:29 AM
🟦 * OUTER01-LOOPS/PID1/PV	Smart PIDPlus Measured Process Variable (%)	59.98	%	7/21/2010 11:46:29 AM
🟧 * OUTER01-LOOPS/SENSOR1	Smart PIDPlus Actual Process Variable (%)	60.00		7/21/2010 11:46:29 AM
🟪 OUTER01-LOOPS/PID1/OUT	Smart PIDPlus Output (%)	60.01	%	7/21/2010 11:46:29 AM
■ OUTER01-LOOPS/PID2/SP	Traditional PID Setpoint	60.00	%	7/21/2010 11:46:29 AM
■ * OUTER01-LOOPS/PID2/PV	Traditional PID Measured Process Variable (%)	62.49	%	7/21/2010 11:46:29 AM
⬛ * OUTER01-LOOPS/SENSOR2	Traditional PID Actual Process Variable (%)	63.81		7/21/2010 11:46:29 AM
🟫 OUTER01-LOOPS/PID2/OUT	Traditional PID Output (%)	63.42	%	7/21/2010 11:46:29 AM

For Help, press F1

Fig. 14.9 Wireless PID with a default update rate of 60 seconds for pH and 16 seconds for flow provided exceptionally tight static mixer setpoint control

prolong battery life and valve packing life by increasing the time interval between communications. Larger wireless update time also reduces feedforward timing errors [26]. The deadtime introduced into the loop is on the average about half of the update time since the measurement result is at the beginning of the time interval [6, 30–32, 34]. The additional deadtime increases the ultimate limit to loop performance for unmeasured disturbance per (14.3) and (14.4).

14.2.14 Large Analyzer Cycle Times

A large analyzer cycle time offers a similar opportunity for the use of a wireless PID. The primary difference is that the analyzer cycle time is typically much larger than the wireless update time, which means the wireless PID is beneficial for much slower processes. For an analyzer cycle time larger than the 63% process response time (without the analyzer delay), the use of the enhanced PID provides excellent setpoint control by allowing the use of a PID gain that is the inverse of the open loop gain. The enhanced PID is stable without retuning when the cycle time is increased due to the addition of sample points, decrease in sample flow rate, or an increase in chromatograph column length. The deadtime introduced into the loop is on the average about 1.5 times the cycle time since the measurement result is at the end of the time interval [6, 32]. The additional deadtime increases the ultimate limit to performance for unmeasured disturbance per (14.3) and (14.4). In the enhanced PID setpoint response, the measured PV has the analyzer delay but the actual process does not have the delay. Thus, if the controller gain is the inverse of open loop gain, the rise time does not depend upon the additional delay introduced by the analyzer.

14.2.15 High Process Nonlinearity

The process gain and deadtime for inline blend composition, heat exchanger, vessel coil temperature, and jacket temperature control from the material and energy balance is inversely proportional to flow. For pH and conductivity, there is also a nonlinearity seen as the slope in the plot of pH and conductivity versus acid or base concentration. For column temperature, there is a nonlinearity that is the slope in the plot of temperature versus the manipulated flow to feed flow ratio. Finally, there is the gain nonlinearity from installed characteristic of the control valve. If the curves are known, the nonlinearity of pH, conductivity, and column temperature are best handled by a PID input signal characterization and the nonlinearity of the valve characteristic by PID output signal characterization. An adaptive controller is still needed to correct for the changes in the curves with operating conditions such as feed composition for pH and conductivity and pressure for the valve. For inline control, the tuning settings would be scheduled based on total flow. For split ranged control valves, there can be a huge difference in process dynamics if different valves and different types of streams are manipulated. In all cases, an adaptive controller that identifies the process dynamics and the scheduling of tuning as a function of a user selected variable is critical. The scheduling provides preemptive adjustment of the tuning based on recent or best test results.

14.2.16 High Process Deadtime

The total loop deadtime sets the ultimate limit to performance as shown in (14.3) and (14.4). The sources of process deadtime are transportation delays, mixing delays, dissolution times, reaction times, thermal lags, and volumes in series (e.g., column trays). The sources of automation system deadtime are threshold sensitivity and resolution limits and backlash for a ramping signal, actuator lags, sensor lags, transmitter damping, analyzer cycle times, signal filters, module execution time, and digital update time. Except for digital delays and signal lags, the deadtime is variable. A simple addition of a deadtime block in the external reset limit path and the enabling of the dynamic reset limit option provide deadtime compensation with performance equivalent to a Smith Predictor. The improvement is only seen if the controller tuning is made faster. The integral time can be decreased to half of the total loop deadtime if the total loop deadtime is accurately set. The improvement and robustness is greatest for a loop where the process time constant is greater than the total loop deadtime, which is counter to common belief that deadtime compensators are most useful for deadtime dominant loops [23, 24]; see Sect. 14.2.6 on deadtime compensators for more detail.

The best option for dealing with high process deadtime is feedforward control with an accurate measurement of the disturbance and an accurate gain and timing for the feedforward correction.

14.2.17 Feedforward Timing Errors

If the feedforward correction does not arrive at the same time and the same place in the process as the disturbance, an irregular, oscillatory, and confusing response can occur. If the feedforward signal arrives too soon, an inverse response can occur. If the feedforward signal arrives too late, a second disturbance is created. A wireless measurement can help prevent the inverse response and second disturbance if the elapsed time of the wireless is reset upon activation of the feedforward and the default update rate is longer than the time to steady state. However, the additional delay of the wireless update rate could cause instability unless the traditional PID is detuned. An enhanced PID for wireless will prevent the instability without retuning regardless of how much the default update rate is increased to extend wireless battery life or ride out feedforward timing errors.

14.2.18 Split Range Nonlinearities and Discontinuities

The stick-slip is greatest, the change in the slope of the installed characteristic is greatest, and the change in process dynamics is most abrupt at the split range point. The split range can be between multiple secondary loops in cascade control or between multiple final control elements for single loop control. The split ranged final control elements are typically control valves but can be variable speed drives and heaters. When the demand for either final control element is minimal, there tends to be continual oscillation across the split range point. The oscillations can be a limit cycle from resolution or threshold sensitivity limits or from tuning settings that are too fast for the drastic change in the valve gain and in process dynamics. These oscillations introduce variability into the process and wear out valve packing from excessive movement and internal flow elements (plugs, discs, seals, and seating surfaces) from high velocities. When opposing media are manipulated, such as heating and cooling for temperature control and acid and base reagents for pH control, the oscillations cause a loss in process efficiency from an increase in energy and raw material use. The selection of the best split range point, adaptive tuning, and the enhanced PID algorithm can eliminate these oscillations.

The split range point is traditionally set at 50%. To compensate for the secondary loop or final control element and process gain nonlinearity from split range control, the split range point should be set per (14.21a) instead of 50% so that the ratio of the controller output spans for each split ranged secondary loop or final control element is equal to the ratio of the product of the process and secondary or element gains. If you consider the useable range of the PID output to be 0–100% and a gap (G) could be set at the split range point, you also have (14.21b) for the first span (S_1). If you solve (14.21b) for the second span (S_2) and substitute the result into (14.21a), you end up with (14.21c) for the first span in terms of element and process gains [7]. Since these gains change with operating conditions and PID tuning also depends upon the open loop time constant and deadtime, adaptive tuning is still beneficial. The subscript "v" for the final control element gain is used because most split ranged outputs go to valves.

$$S_{x1} = \frac{K_{v1}K_{p1}}{K_{v2}K_{p2}} S_{x2}, \tag{14.21a}$$

$$S_{x1} = 100 - G - S_{x2}, \tag{14.21b}$$

$$S_{x1} = \frac{K_{v1}K_{p1}}{K_{v1}K_{p1} + K_{v2}K_{p2}}(100 - G). \tag{14.21c}$$

Adaptive tuning can identify the changes in the final element and process dynamics. The tuning settings can be scheduled based on controller output with the regions more closely spaced around the split range point to deal with the discontinuities. If the best split range point is selected, the identification task is easier and the time to get a high quality model is shorter.

The gap at the split range point is a method to prevent temporary inadvertent manipulation of the other final control element. However, integral action in the PID or in the process will eventually cause a transition to the other element. Consequently, the benefit is mostly in terms of preventing noise from causing an unnecessary crossing of the split range point. The consequences of noise are better minimized by transmitter damping or signal filtering. For wireless measurements, attenuation of noise by transmitter damping is preferred to prevent unnecessary updates decreasing battery life and reducing the enhanced PID performance.

Integral deadband is used to help prevent limit cycles and to slow down the transition between split ranged valves. However, integral deadband slows down the response to an upset and complicates the tuning. User-selection of the integral deadband is difficult because high variability and nonlinearity of dynamics and stick-slip is the norm at the split range point. The enhanced PID requires no adjustment to prevent limit cycles and unnecessary transitions.

The PID enhanced with a wireless measurement has the patience to wait out short term transitions and discontinuities. If the damping in the transmitter and the threshold sensitivity for an update are set to prevent transmission of noise, the enhanced PID accomplishes the objective of the split range gap and integral deadband. The ability to deal with discontinuities is especially significant since there are no preformulated PID parameters or options. For example, a measurement default update rate of 30 seconds for jacket temperature can reduce unnecessary excursions across the split range point and overreaction to steam shock in addition to improving the loop's setpoint response for vessel temperature cascade control. If the default update rate is larger than the process response time, the PID tuning becomes independent of process deadtime and time constant. The PID gain can be simply set equal to the inverse of the open loop gain (K_o).

Split range control has often been used to achieve greater control valve rangeability by throttling a small valve in parallel with a large valve when the demand for flow is low. For a low flows, the small valve starts to open and the big valve freezes at minimum position to prevent riding the seat of the big valve which is the point of greatest stick-slip and wear. However, unless the big valve threshold sensitivity, resolution, and backlash in percent of stroke is much less than the small valve, the limit cycles at high flow demand are much larger. A better solution is to eliminate split range control and use instead valve position control where the process PID manipulates the small valve and a valve position control (VPC) PID manipulates the big valve. Traditionally, integral only control is used in the VPC with an integral time that is 10× the product of the process PID gain and reset time to eliminate interaction. However, this action is sometimes too slow for large disturbances. For measured load upsets, feedforward control should be added. Here the feedforward signals for the process PID and VPC PID would be intelligently limited to keep the small valve in a good a throttle range and avoid unnecessarily positioning of the big valve with its larger stick-slip.

The enhanced PID with a large update trigger level for the small valve position can be used for VPC instead of an integral only controller. The enhanced PID for VPC suppresses limit cycles from stick-slip and backlash and reduces interaction with the process PID while offering more aggressive action for large disturbances.

The crossing back and forth and discontinuity of the split range control can also be reduced for opposing media by the temporary transition to VPC instead of completely closing the valve not being throttled by the process PID. In this case, the VPC provides a smooth transition by a small opening of the valve normally shut by split range control in order to keep the valve manipulated by the process PID from dropping below some minimum throttle position. There is a small temporary waste of energy or raw materials while the opposing media valves are both open but this may be less than the loss of efficiency from the crossing back and forth and discontinuity of the split range point. The enhanced PID can be used for this VPC to reduce interactions and provide more aggressive tuning.

Valve position control is also used for the prioritization of the manipulation of multiple flows (multiple PID outputs) and for decreasing energy use and increasing production rate. Here again, the enhanced PID is beneficial.

14.2.19 Multiple Inputs and Outputs

When there are several process variables (PID inputs) with limits that need to be honored, a PID for each constraint is setup for override control. Each override PID has a process variable that needs to be constrained as its controlled variable. The output of the multiple PID goes to a signal selector that determines which output is used to manipulate a secondary loop setpoint or final control element. While we think of override control as a multiple input single output application, the output could be split ranged between multiple secondary loops or elements. The split range block can provide simultaneous besides the customary sequential manipulation of secondary loops or elements.

The principle problem with override control is the tuning, windup, and timely transition of the PID controllers. Each PID must be tuned to account for the different process and measurement dynamics of the controlled variables. The PID algorithm must not windup when not selected and must smoothly takeover control when controlled variable is above and below its setpoint for a high and low limit respectively. Implementation of the integral mode by literally integrating the error between the setpoint and controlled variable has caused the output of unselected controllers to walk off to low and high outputs despite the use of special options such as integral tracking. The positive feedback implementation of the integral mode inherently prevents this problem. Also, the use of the process variable of the secondary loop or the readback of valve position or speed for the final control element provides a smooth transition and prevents the override controller output from changing faster than the secondary loop or final control element can respond. If the proper external reset signal is used and the back calculate output of each function block is properly connected to the back calculate input of the upstream block, the most common cause of a disruptive transitions is the loss of status and continuity in the back calculate path. The loss is typically caused by the introduction of special logic by the user to try and accomplish what is automatically achieved by the proper setup of the override control system. Any special application requirements should be implemented via standard options in the standard PID, "Signal Selector", and "Splitter" blocks rather than by the introduction of calculations or sequences.

Users may desire that the override PID be selected when the process variables crosses the override PID setpoint. Since there is deadtime in all loops just by virtue of the delays and lags in the automation system (final control element, measurement, and controller), the process variable will overshoot the setpoint if correction is delayed to crossing the setpoint. The override setpoint in theses cases will need to be further away from the safety instrumentation system (SIS) trip or relief device setting. A way to eliminate the uncertainty of the setpoint bias needed to prevent activation if the SIS or relief device is to tune the override PID for a negligible overshoot by making the proportional mode contribution greater than the integral mode contribution in the PV approach to the override setpoint.

There may be multiple PID outputs because of alternate resources of fuel, utilities, raw materials, and reagent. Often, one of the resources is more abundant or less expensive. A common example is waste or recycled streams versus purchased fuel, heat, steam, reagent, and reactants. Valve position control (VPC) is used to maximize the use of the more abundant or lower cost resource by increasing its flow until the purchased resource manipulated by the process PID is at a minimum throttle position for minimum stick-slip and maximum valve life. As with other VPC applications, an enhanced PID eliminates limit cycles and reduces interaction and valve wear.

14.2.20 Energy Use and Production Rate Optimization

Valve position control (VPC) is used optimize energy use by decreasing the pressure of boilers and compressors and increasing the temperature of refrigeration units and cooling towers by working the furthest open user valve towards its maximum throttle position. The controlled variable for the VPC is the high signal selector output of user valve positions. The VPC PID output is the pressure or temperature setpoint of the utility.

Similarly VPC can be used to optimize production rate by increasing feed rate to columns, crystallizers, evaporators, and reactors by working the furthest open utility and feed valve towards its maximum throttle position. The controlled variable for the VPC is the high signal selector output of utility or feed valve positions. The VPC PID output is the feed flow setpoint.

To minimize interactions an enhanced PID can be used with a threshold sensitivity setting large enough to ignore insignificant changes in user, utility, or feed valve position that is the output of the high signal selector.

14.2.21 Process Interactions

The two predominant methods of reducing PID interaction are separation of the dynamics and decoupling. The separation of dynamics is accomplished by tuning the PID so that the closed loop time constant (Lambda) of the fighting PID loops are a factor of 10 different. The faster loop is made faster by a higher PID gain (smaller Lambda) and, if necessary, the slower loop is made slower by a lower PID gain (larger Lambda). As the Lambda is successively increased, integral only control is approached. If the loops have similar speeds, the most important loop is made faster. The enhanced PID and wireless measurements can be used to reduce interaction. The wireless trigger level setting is larger for the less important PID.

Decoupling can be achieved by adding a feedforward signal of one PID output to the other PID output. Half decoupling, the use of a feedforward for just one of two interacting PID, is sufficient to break interaction. The PID output from the least important loop is applied as feedforward to the more important loop. If there is no significant difference in the relative importance of the PID, then the feedforward is applied to the PID loop with the greatest deadtime.

14.2.22 Communication and Component Failures

Measurements or final control elements can fail to update because there is a loss of communication or a component failure to last value. Communication failures can occur in bus systems due to link failures and in wireless systems due to low battery power. Sensor failures to last value occur for cracked or coated pH glass electrodes and plugged DP impulse lines. Final element failure to the last value occurs when a control valve does not move due to excessive stiction from high temperature, binding or occlusion of the internal control element (e.g., seat and plug) from the build up of solids and coatings, or the last position air failure of piston actuators.

When a failure to update occurs, a conventional PID will continue to ramp its output based on the inevitable error between the controlled variable and the setpoint. If the PID error is large or the duration of the failure is long, the output will hit its output limit. When the failure is corrected and an update occurs, the traditional PID considers entire change in measurement to have occurred in the last execution of the PID. The result is a large step in the output from proportional action and a bump from derivative action.

The enhanced PID suspends integral action when there is no update. As a result the PID output stays at the last value before the failure which is the least disruptive action to the process. When the measurement or valve responds or the communication is restored, the enhanced PID makes a small proportional and derivative mode and gradual integral mode correction based on difference between the current and the last known value. The enhanced PID is able to ride out update failures without overreaction.

14.2.23 Batch Profile Optimization

Batch columns, crystallizers, evaporator, and reactors have a key concentration and or temperature endpoint. Temperature endpoints are measured online but concentration endpoints are often based on lab analysis and thus depend upon repeatable conditions. The path of the key measurement from start to finish of the batch crystallization, reaction, or separation process is a profile. As with most optimization opportunities, the first objective in batch control is to make the profile repeatable so that the batch cycle time and endpoint are consistent and differences can be tracked down to changes in raw material inputs, utility systems, instrumentation, or operator actions. Since the achievement of a repeatable batch is an important step that should not be bypassed, the use of the best batch, "Golden Batch", as the objective instead of matching the average of a representative set of batches is counter productive [4]. If the batch profile is controlled, variability is transferred from the profile to manipulated process inputs (flows). The result is a repeatable profile than can be optimized and a narrowing of the problem to analysis of the changes in flow. The process knowledge gained from data analytics can be used to optimize the profile.

The measurement, control and optimization of the profile by a PID can provide advanced batch control. However, since the actual and desired profile response is only in one direction, typically an increase in production concentration, the standard PID algorithm cannot make corrections in both directions. A flattening of the profile and a possible windup from integral action can occur if the setpoint profile outpaces the actual profile. If the rate of change of temperature or concentration is used as the controlled variable, the PID response is in both directions and smooth. The set point is now the slope of the optimum profile [4, 16].

For batch profile slope control, there is a translation of controller tuning settings from the use of measurement rate of change as the controlled variable. Proportional action is now derivative action, integral action is now proportional action, and derivative action is acceleration or second derivative action. Often the optimization of the profile simply consists of maximizing the slope [4, 16]. The maximum slope corresponds to the maximum biological product formation rate, crystallization rate, evaporation rate, chemical reaction rate, and separation rate.

14.2.24 Plantwide Feedforward Control

A plant can go to a new production rate within minutes by plantwide feedforward control. If all of the key flows (feed, utility, recycle, and vent flows) are set to the values on the Process Flow Diagram (PFD), the plant can move to a new production rate without waiting on composition, level, pressure, pH, and temperature loops to respond.

The PFD values may not be a simple ratio of the new to old production rate although this could be a good starting point. Better would be sets of flows captured at different operating rates and ambient conditions (e.g., summer versus winter operation) for the flow feedforwards. Process PID loops would as usual trim through feedback correction the feedforwards. The concept can be expanded to product grade changes and fed-batch operation for faster and more adaptive transitions and batches.

If a plant can respond quickly to changes in raw material supply and market demands, the inventories can be reduced. If a plant can increase rates when utility costs are low (e.g., off peak time rates) and ambient temperatures are low for cooling water, the energy efficiency can be increased.

14.2.25 Impact of Threshold Sensitivity and Noise on Performance

The equations presented in Sect. 14.2.1 on loop performance can be expanded to include the effect of wireless and valve threshold sensitivity and measurement noise.

For a step change in an unmeasured disturbance that would cause an open loop error of (E_o), the minimum peak error (E_x) (14.22) is proportional to the ratio of the total loop deadtime to the 63% response time (T_{63}). The minimum integrated error (E_i) (14.23) is approximately the peak error multiplied by the total loop deadtime that is the original deadtime (θ_o) plus the additional deadtime from a wireless measurement (θ_w) and a control valve (θ_v). The 63% response time (14.24) is simply the sum of the original loop deadtime plus the wireless measurement deadtime and the open loop time constant (τ_o), which is hopefully the primary process time constant.

Note that the valve deadtime from stiction and backlash does not appear in the 63% open loop step response time because the step change in controller output must be larger than the valve resolution, threshold sensitivity, and deadband to create a response. Consequently, on open loop step response trend charts, the deadtime from valve stiction and backlash is not seen. However, in closed loop operation, the controller output is ramping for an unmeasured disturbance and the time the controller takes to ramp through the valve resolution, threshold sensitivity, and deadband is additional deadtime

$$E_x = \frac{\theta_o + \theta_w + \theta_v}{T_{63}} E_o, \tag{14.22}$$

$$E_i = \frac{(\theta_o + \theta_w + \theta_v)^2}{T_{63}} E_o, \tag{14.23}$$

$$T_{63} = \theta_o + \theta_w + \tau_o. \tag{14.24}$$

The additional deadtime from a wireless measurement (14.25) is the smallest of the deadtimes from the wireless "default update rate" (update time interval) for periodic reporting (ΔT_w), and the wireless "trigger level" (threshold sensitivity) for exception reporting (S_w). The deadtime from periodic reporting ($\theta_{\Delta T}$) (14.26) is one half of the update time interval. The deadtime from exception reporting (θ_S) (14.27a) is one half of the threshold sensitivity setting divided by the maximum rate of change of the % process variable $(\Delta\mathrm{PV}/\Delta t)_{\max}$. Half of the interval and threshold sensitivity settings are used because on the average the disturbance starts halfway in the interval and halfway in the sensitivity.

The rate of change is a maximum during the beginning of the disturbance before the control loop has had any effect. This maximum rate of change is dictated by the size of the disturbance and process dynamics. The maximum ramp rate (14.27b) can be approximated by a near integrator gain multiplied by the equivalent change in controller for the disturbance. The equivalent change in controller output is the open loop error (E_o) divided by the open loop gain (K_o). The near integrator gain (14.27c) is the open loop gain divided by the open loop time constant (τ_o). The substitution of (14.27c) into (14.27b) cancels out the open loop gains giving the maximum ramp rate (14.27d) as simply the open loop error divided by the open loop time constant. The substitution of (14.27d) into (14.25) yields (14.27e) where we see that the deadtime from the threshold sensitivity setting decreases as the size of the disturbance increases and the open loop time constant decreases, both of which cause a faster rate of change

$$\theta_w = \min(\theta_{\Delta T}, \theta_S), \tag{14.25}$$

$$\theta_{\Delta T} = 0.5\Delta T_w, \tag{14.26}$$

$$\theta_S = \frac{0.5 S_m}{(\Delta\mathrm{PV}/\Delta t)_{\max}}, \tag{14.27a}$$

$$(\Delta\mathrm{PV}/\Delta t)_{\max} = K_i(E_o/K_o), \tag{14.27b}$$

$$K_i = \frac{K_o}{\tau_o}, \tag{14.27c}$$

$$(\Delta \mathrm{PV}/\Delta t)_{\max} = \frac{E_o}{\tau_o}, \tag{14.27d}$$

$$\theta_S = \frac{0.5 S_m \tau_o}{E_o}. \tag{14.27e}$$

Similarly, the deadtime from valve threshold sensitivity (S_v) can be estimated (14.28a) as half of the sensitivity divided by the maximum rate of change of the controller output that is initially mostly due to the proportional mode. This rate of change of controller output (14.28b) is the rate of change of the process variable multiplied by the controller gain (K_c). The controller gain (14.28c) can be approximated as the Ziegler–Nichols gain multiplied by a detuning factor (K_x) to provide robustness and a smoother response. However, the gain should not inflict disturbances from fluctuations in the PID output from exceeding the control valve threshold sensitivity. An enhanced PID noise band expressed as measurement threshold sensitivity (S_m) can reduce this noise allowing a higher controller gain. The substitution of (14.28c) into (14.28b) cancels out the open loop time constant (14.28d). The substitution of (14.28d) into (14.28a) yields (14.28e) where the deadtime is proportional to the product of the open loop gain and original deadtime and inversely proportional to the product of the detuning factor and open loop error

$$\theta_v = \frac{0.5 S_v}{(\Delta \mathrm{CO}/\Delta t)_{\max}}, \tag{14.28a}$$

$$(\Delta \mathrm{CO}/\Delta t)_{\max} = K_c (\Delta \mathrm{PV}/\Delta t)_{\max}, \tag{14.28b}$$

$$K_c = \min\left[\frac{K_x \tau_o}{K_o \theta_o}, \frac{S_v}{\max[(N_m - S_m), 0.002]}\right], \tag{14.28c}$$

$$(\Delta \mathrm{CO}/\Delta t)_{\max} = \frac{K_x E_o}{K_o \theta_o}, \tag{14.28d}$$

$$\theta_v = \frac{0.5 S_v K_o \theta_o}{K_x E_o}. \tag{14.28e}$$

14.3 Unit Operation Examples

The process industry is quite diverse in terms of production methods. While each industry tends to see itself as unique, similarities emerge when viewed on a conceptual level. The solutions previously described offer a tool palette that addresses most of the industrial opportunities despite widely different processes. Examples of these solutions are offered for some of the most important unit operations in process industry. While not detailed, the use of the enhanced PID for valve position control can in general offer an easy, fast, and inexpensive increase in production rate of continuous and fed-batch unit operations.

14.3.1 Biological Reactors

Bioreactors are used for the production of most of new pharmaceuticals that are proteins too complex to be produced by chemical processes. Most of the older biological processes use genetically engineered fungal and bacterial cells. As biopharmaceutical proteins become more sophisticated, genetically engineered mammalian cells, such as Chinese hamster ovaries (CHO) cells are increasingly used. Biopharmaceutical batch processes are predominantly batch because of the concern for the buildup of toxins and genetically deficient cells with continuous processes and the need to get new drugs to market quickly.

The important loops for bioreactors are pH, temperature, and dissolved oxygen. In general, the allowable controller gain is high-limited only by the measurement noise since the ratio of deadtime to time constant is small for a well designed system. The high controller gain in combination with the extremely slow disturbances translates to exceptionally tight PID control. Most users do not see much of an advantage from improved tuning for well designed systems. The main concern with these systems is getting to temperature setpoint as fast as possible with essentially no overshoot at the start of the batch. When this fast start is important, the fast setpoint response techniques such as smart bang-bang control can be used [13]. If there is no concern about batch cycle time and low temperatures do not cause seed cell degradation, a PID structure of integral mode on error and proportional and derivative modes on PV is used to eliminate overshoot [24].

Often the automation and bioreactor system design is less than ideal. If the piping and sparger for the injection of oxygen or air for dissolved oxygen control and addition of carbon dioxide for pH control are not completely separated, the dissolved oxygen and pH loops interact. In this case, a half decoupler and an enhanced PID can reduce the interaction. Often variable frequency drives create electromagnetic interference and ground potentials that cause spikes in pH measurements that disrupt the dissolved oxygen and pH loops. Wireless pH transmitters have been found to eliminate the spikes commonly seen in bioreactor loops [28].

Many bioreactor loops do not use sliding stem (globe) control stem valves because they are not suitable for sanitary and sterilization-in-place (SIP) service. The alternative final control elements used may have a poor resolution or sensitivity limit and backlash. The enhanced PID can eliminate the associated limit cycles.

There remains a largely unrealized opportunity for PID control of substrates, such as glucose. Online analyzers can provide glucose concentration measurements every 4 to 6 hours depending upon the number of bioreactors serviced by the analyzer and the auto sampler system design. For a bioreactor batch that takes 1 to 2 weeks, this sample time is fast enough to do closed loop control if an enhanced PID is used to deal with the analyzer cycle time and resolution limits.

Total cell concentration can be measured by turbidity sensors. The same analyzer used for glucose may be able to provide cell count and diameter and the concentration of precursors to cell death. Smart dielectric measurements can indicate what fraction of the cells still have their membranes intact and thus are viable cells for product formation. Consequently, the viable cell concentration profile can be controlled and optimized by the use of an enhanced PID with a controlled variable that is the rate of change of viable cell concentration and whose output is the glucose concentration and pH setpoints. The manipulation of the pH setpoint presents an additional challenge because of the narrow peak in the plot of cell growth and death rate versus pH and the reversal of process gain. The pH optimization can be limited to one side of the peak by the online computation of an inferential measurement of the change of viable cells per change in pH. It is conceivable that this inferential measurement (Δ cells/Δ pH) could be used as the controlled variable in a loop to eliminate the reversal of process sign problem. Here, the theoretical optimum PID setpoint of zero at the peak could be trimmed by the cell profile PID.

The online measurement of product concentration is more problematic, but given an inferential measurement is developed, online batch product concentration profile control is possible by the same type of PID used for viable cell concentration profile control. The optimum glucose and pH for product formation rate can be different than for optimum cell growth rate. The cell concentration profile PID may be used early in the batch during the pre-exponential and early exponential growth phases. The product concentration profile control would take over in the last half of the batch when product formation rate becomes significant.

14.3.2 Chemical Reactors

Chemical reactors set the stage for the production of bulk chemicals, intermediates, petrochemicals, polymers, pharmaceutical chemicals, and specialty chemicals. If the reaction is not producing the right product, not much else matters. Pharmaceutical and specialty chemicals use batch reactors while the other products predominantly use continuous reactors.

The Temperature PID is the most important controller since the reaction rate is often an exponential function of temperature via the Arrhenius Equation. The deadtime to time constant ratio for well mixed continuous reactors and the product of deadtime and integrating process gain for batch reactors are incredibly small (<0.001). The result is a permissible PID gain much larger than users are accustomed to (>50) and exceptionally tight temperature control. Since the most important task of the utility system is to satisfy the demands of the temperature PID, the transfer of variability by the PID from the reactor temperature to its utility system is maximized. The exception is where the utility system is used by a more important reactor in a more critical portion of its operation. The main limit to performance is the accuracy of the temperature sensor and the threshold sensitivity and resolution of the final control elements. Premium Resistant Temperature Detectors (RTD) and sliding stem valves with digital positioners should be used to allow the full capability of the PID to be realized [24–26].

Cascade control where the reactor temperature is the primary PID whose output is the setpoint of a jacket or coil temperature PID is used to linearize the reactor temperature loop and to provide fast compensation of disturbances to the heat transfer media [12]. Since the reactor temperature PID gain is exceptionally high there is a significant risk the output of this primary PID will change faster than the secondary PID can respond. For large disturbances or large setpoint changes, this cascade control system can burst into oscillations. The use of the dynamic reset limit in the positive feedback implementation of integral action will prevent this problem. The external reset signal is the controlled variable of the secondary PID (jacket or coil temperature). For final control elements that are slower than the secondary loop PID, which can occur for large control valves or variable frequency drives with speed rate of change limits, the dynamic reset limit is extended to the secondary loop with an external reset signal of valve position or speed.

The reactant feed PID controllers are important to keep the reactants in the optimum ratio since the reaction rate and completeness depends on the concentration of reactants. Coriolis meters with their exceptionally accurate mass flow and density measurement can be used to provide a component mass balance and inferential measurement of feed composition, respectively [24–26]. A PID that has the optimum ratio of component mass as it controlled variable can slowly trim the ratio of reactants. Alternately, for exothermic reactions, a PID whose controlled variable is cooling rate and thus reaction rate can be used to trim the feed ratio.

If the utility systems are a constraint to increasing production rate, override controllers with integral only action or the enhanced PID algorithm can be used as valve position controllers (VPC). The controlled variable of the VPC is the utility system PID output, and the VPC setpoint is the maximum throttle position of the utility valve or maximum speed of a variable frequency pump. The output of each VPC goes to a low signal high signal selector whose output becomes the ratioed setpoint for the feed PID. These feed PID should be tuned with the same closed loop time constant (Lambda) so that changes in feed rate are coordinated and the total component masses in the reactor are maintained in the desired ratio.

The temperature and concentration profile of a batch reactor can be controlled by the use of a PID whose controlled variable is the rate of change of temperature or concentration with time as discussed for biological reactors [16]. However, the prevention of too fast of an increase in reaction rate for extremely exothermic reactors is more critical than in biological reactors because the exponential increase in reaction rate can through positive feedback create a runaway response. For runaway reactors, there is a window of PID gains where too low of a gain can result in a runaway besides the more familiar case of too high of a gain causing a growing oscillation. Since the permissible gain is much higher than most users realize, the more common mistake is a PID gain dangerously close to the low gain limit. Maximizing derivative action is also important to prevent a runaway. The PID rate time should be set equal to the sum of the thermowell and heat transfer surface lags. If the controlled variable is rate of change of temperature, then the integral mode is effectively the proportional mode and the proportional mode is effectively the derivative mode in terms of the temperature response. Thus, there is no integral mode to force the temperature to a specific temperature value, which is consistent with the objective of the progression of the reaction rate in the batch being most important. However, there is an important consideration in the setting of the integral time for runaway reactions. If temperature rather temperature rate of change is used as the controlled variable, the reset time should be increased by a factor of 10 to prevent a runaway because of the extreme sensitivity of a process with positive feedback to the integral mode. Some severely exothermic polymerization reactors use proportional-derivative PD controllers (structure 4) for reactor temperature control because integral action has no sense of direction and promotes overshoot by delaying cooling until the temperature has crossed setpoint [24].

14.3.3 Crystallizers

Batch crystallizers are an important unit operation in pharmaceutical and specialty chemical processes. To minimize the formation of small seed crystals that can result in excessive fines and coating (frosting) in the beginning of the batch and to promote crystal growth and size toward the end of the batch, an optimum cooling curve and consequently temperature profile is used [27]. In the early part of the batch, the batch temperature is slowly lowered. The decrease in batch temperature is accelerated as the batch progresses to provide an exponential increase in cooling that is maximum at the near the end of the batch. Since crystallization releases heat, the PID can increase besides decrease the temperature and a temperature profile can be the setpoint for a PID whose controlled variable is crystallizer temperature. The use of temperature rate of change with time as the controlled variable is not necessary but might reduce batch time and prevent flattened sections or bumps in the temperature profile by emphasizing a continual decrease in temperature.

14.3.4 Distillation Columns

Column temperature at a given pressure provides an inferential measurement of column composition. The location in the column chosen for temperature control is the point where the change in temperature is the largest and most symmetrical for an increase and decrease in the flow manipulated by the temperature PID. The most effective scheme is the manipulation of distillate flow if this flow is not too small and the overhead receiver cross-sectional area is not too large. By manipulation of distillate flow, the reflux flow manipulated by receiver level control provides a degree of internal reflux control by changes in overhead vapor and hence reflux for changes in column loading and wall temperature. However, this scheme depends upon a change in distillate flow immediately translating to an appreciable change in reflux flow. If the overhead receiver area is too large, the change in level may be within the noise band or below the threshold sensitivity limit of the level measurement. The receiver level controller must have a very high gain. Normally, the product of deadtime and integrating process gain is so small the only limit to setting the overhead receiver level controller gain is measurement noise. However, users may not realize this and be close to the low gain limit that causes slow rolling oscillations besides poor temperature control. The oscillations may be incorrectly attributed to high of a PID gain and the PID gain may be decreased making the problem worse. If the distillate flow is too small (the case where a small amount of impurity is driven off the top), the temperature manipulates the steam flow or the bottoms flow. If the bottoms flow is manipulated, a sump level PID manipulates the steam flow. In this scheme, level control is poor from an inverse response in the sump level due to shrink from the collapse of bubbles and the swell from the formation of bubbles [27].

The deadtime to time constant ratio for column temperature loops is typically about 0.1 because the largest source of deadtime is the equivalent delay that results from a large number of interacting time constants from trays in series. The maximum temperature controller gain is about 4 times the inverse of the open loop gain. Typical temperature controller gains range from 2 to 20 for narrow and wide span temperature ranges, respectively, if the control point exhibits reasonable sensitivity of temperature to a change in composition. While the deadtime to time constant ratio is good for control, the absolute magnitude of the deadtime is large for tall columns (columns with a large number of actual or equivalent trays) [27].

The use of feedforward control for temperature and level control of columns has proven to be beneficial because of the large deadtime value and the moderate controllers gains used. The most frequent disturbance is a feed change. Flow feedforward control is used where the reflux, distillate, steam, and bottoms flow setpoints are ratioed to the feed flow and corrected by a feedforward summer in the temperature and level PID. The timing is complex. The corrective action of the steam or reflux flow must arrive at the temperature control point at the same time as the feed disturbance. Dynamic compensation is normally used to improve the timing of the feedforward where a delay is added to a feedforward correction that arrives too soon and a lead is added to compensate for a lag in the feedforward correction that is greater than the lag in the disturbance. An enhanced PID could help prevent oscillations from improper feedforward timing and eliminate limit cycles from measurement and valve threshold sensitivity and resolution limits and backlash. Since the rate of change of temperature is extremely slow, the signal-to-noise ratio could be improved by the use of wireless measurements with a default update rate chosen to be slow enough to insure the change in signal is more reflective of a change in true temperature than resolution limits or noise. The use of portable wireless integral mounted temperature transmitters would also help finding the optimum temperature control point. Similarly, portable wireless integral mounted pressure transmitters would enable finding the trays that are flooding as column rates as pushed. As in the reactor feed maximization, valve position controllers for each limiting valve capacity could be used as override controllers whose largest output is used to push the column feed rate higher.

“Ratio” blocks are used to multiply the feed flow by the desired ratio. The “Ratio” block also has the actual ratio. The feedback correction by the temperature or level PID is done by a “Bias/Gain” block that acts a feedforward summer. The desired and actual ratio are displayed for the operator. For startup until column temperature and traffic has reached the normal operating range, the temperature PID is put in manual and the operator sets the desired ratio for pure flow ratio control.

14.3.5 Evaporators

In an evaporator, often a pressure PID manipulates the vapor flow out the top and a level controller manipulates the liquid discharge flow out the bottom. The steam flow sets the evaporation rate. The concentration of the product can be controlled by the use of a Coriolis meter to measure product density in a recirculation line or in the discharge line if there is always a discharge flow. The product density controller manipulates the steam flow. A “Ratio” block can be used to multiply the steam flow by the desired ratio of steam to feed flow. If a Coriolis meter is also used on the feed flow, the desired ratio of steam to feed flow can be computed from the actual feed density and the desired product density. The product density controller adds a feedforward correction to the output of the ratio block by means of a “Bias/Gain” block. For startup until evaporator temperature and vapor flow has reached the normal operating range, the density PID is put in manual and the operator sets the desired steam to feed ratio for pure flow ratio control.

14.3.6 Neutralizers

The pH control of neutralizers can be particularly challenging due to the extreme nonlinearity and sensitivity of the pH measurement as a result of the exponential relationship between pH and hydrogen ion activity. The changes in controller gain and the rangeability and threshold sensitivity of the final element needed are extraordinary. PID gain changes of 1000 to 1 and a final element rangeability requirement of 10,000 to 1 are possible with strong acid and base systems. The precision of the final element may determine the number of neutralization stages necessary to keep waste streams in compliance with environmental regulations. The limit cycle amplitude from threshold sensitivity and resolution limits can be extremely large due to amplification by the steep slope of the titration curve. The 7 pH value for a broken electrode or wire and the failure to last value of a coated electrode are insidious [7].

Signal linearization, adaptive tuning, split range control, valve position control (VPC), and the enhanced PID are solutions to be considered. Signal linearization can translate the controlled variable from pH to reagent demand per the titration curve. Adaptive tuning can correct for changes in the titration curve and process dynamics. Split ranged control can enable neutralization with both acids and bases. A VPC can adjust a large (coarse) valve to keep a small (fine) valve in a good throttle range for pH control. A VPC can also maximize the use of waste and low cost reagents. Portable wireless pH transmitters can optimize the control location and eliminate spikes from EMI. Finally, the enhanced PID can eliminate oscillations from split range point discontinuities and from valve backlash, threshold sensitivity, and resolution, reduce interactions between the small and large valves, prevent overreaction to pH electrode failures, and extend wireless battery life [21, 24, 28].

14.4 Conclusion

The PID has been the predominant method of feedback and feedforward control for the process industry since the inception of automation systems. The PID provides a consistent and convenient interface for the operator. The algorithm is optimal for unmeasured disturbances and unknown dynamics, the common case in manufacturing. Intelligence can be added to expand the capability through adaptive tuning, deadtime compensation, feedforward control, signal linearization, and smart bang-bang logic. Key capabilities, such as the "dynamic reset limit", "measurement threshold sensitivity setting", and an enhanced PID developed for wireless, can eliminate oscillations and improve efficiency (reduce raw materials, utilities, and waste), flexibility, operability, maintainability, profitability, and safety, that determines compliance and competitiveness. The innovative use of the developing and expanding capability of the PID is the key to sustainable manufacturing.

References

1. Astrom, K., Hagglund, T.: Advanced PID Control. ISA, Research Triangle Park (2006)
2. Blevins, T., McMillan, G., Wojsznis, W., Brown, M.: Advanced Control Unleashed. ISA, Research Triangle Park (2003)
3. Bohl, A., McAvoy, T.: Linear feedback vs. time optimal control. II. The regulator problem. Ind. Eng. Chem. Process Des. Dev. **15**(1) (1976)
4. Boudreau, M., McMillan, G.: New Directions in Bioprocess Modeling and Control. ISA, Research Triangle Park (2008)
5. Harriott, P.: Process Control. McGraw Hill, New York (1964)
6. McMillan, G.: Tuning and Control Loop Performance. ISA, Research Triangle Park (1994)
7. McMillan, G.: Advanced pH Measurement and Control. ISA, Research Triangle Park (2003)
8. McMillan, G.: Models Unleashed. ISA, Research Triangle Park (2004)
9. McMillan, G.: What is your flow control valve telling you. Control Design, May 2004
10. McMillan, G., Sowell, M., Wojsznis, P.: The next generation—adaptive control takes a leap forward. Chemical Processing, Sept. 2004
11. McMillan, G.: Good Tuning: A Pocket Guide. ISA, Research Triangle Park (2005)
12. McMillan, G.: Life's a batch. Control, May 2005
13. McMillan, G.: Full throttle batch and startup response. Control, May 2006
14. McMillan, G.: Improve control loop performance. Chemical Processing, October 2007
15. McMillan, G.: Effect of sample delay on standard PID tuning and loop performance. Advanced Application Note 5, Emerson, Austin (2008). http://www.modelingandcontrol.com/repository/AdvancedApplicationNote005.pdf
16. McMillan, G.: Unlocking the secret profiles of batch reactors. Control, July 2008
17. McMillan, G.: Is wireless control ready for prime time? Control, July 2009
18. McMillan, G.: Essentials of Modern Measurements and Final Elements in the Process Industry, 2nd edn. ISA, Research Triangle Park (2010)
19. McMillan, G.: Centrifugal and Axial Compressor Control. Momentum Press, New York (2010)
20. McMillan, G.: Wireless—Overcoming challenges of PID control & analyzer applications. InTech, July–August 2010
21. McMillan, G.: DeltaV v11 PID enhancements for wireless. Emerson white paper, Austin, August 2010
22. McMillan, G.: How to get the most out of your PID. In: ISA Automation Week, Houston, October 2010

23. McMillan, G.: Deminar #10 Deadtime compensator myth buster. Emerson website entry and screencast recording, Austin, October 2010. http://www.modelingandcontrol.com/2010/10/review_of_deminar_10_-_deadtim.html
24. McMillan, G.: Exceptional opportunities in process control. ISA short course, Saint Louis, 8 December 2010. http://www.modelingandcontrol.com/2010/12/exceptional_opportunities_in_p_18.html
25. McMillan, G.: Measurement and valve errors—Sources and impact. Emerson website entry, Austin, 9 December 2010. http://www.modelingandcontrol.com/2010/12/measurement_and_valve_errors_-.html
26. McMillan, G.: Star performers in minimizing life cycle costs. Emerson website entry, Austin, 16 December 2010. http://www.modelingandcontrol.com/2010/12/star_performers_in_minimizing.html
27. McMillan, G.: Advanced Temperature Measurement and Control. ISA, Research Triangle Park (2011)
28. McMillan, G., Baril, R.: pH measurement and control. Chemical Engineering, August 2010
29. McMillan, G., Dasani, S., Jagadeesan, P.: Adaptive level control. Control, February 2010
30. Shinksey, F.: The effect of scan period on digital control loops. InTech, June 1993
31. Shinskey, F.: Feedback Control for Industrial Processes. McGraw Hill, New York (1994)
32. Shinskey, F.: Process Control Systems. McGraw Hill, New York (1996)
33. Shinskey, F.: The power of external reset-feedback. Control, May 2006
34. Shinskey, F.: Maximizing control-loop performance. Control, May 2010
35. Studebaker, P.: Sustainable plant. PutmanMedia (2011). http://www.sustainableplant.com/
36. Wojsznis, W., Blevins, T., Wojsznis, P.: Adaptive feedback/feedforward PID controller. In: ISA Conference, Houston (2003)

Part IV
Non-standard Approaches to PID

Chapter 15
Fractional-Order PID

Blas M. Vinagre and Concepción A. Monje

15.1 Introduction

It is usual in undergraduate courses of feedback control to introduce the *basic control actions* and their effects on the controlled system behavior in the frequency domain. Thus, we know that these actions are the proportional, the derivative, and the integral ones, and their main effects on the controlled system behavior are [1]:

- To increase the speed of the response and to decrease the steady-state error and relative stability for the proportional action;
- To increase the relative stability and the sensitivity to noise for the derivative action;
- To eliminate the steady-state error and to decrease the relative stability for the integral action.

The positive effects of the derivative action (increase of the relative stability) can be observed in the frequency domain by introducing the $\pi/2$ phase lead, and the negative ones (increase of the sensitivity to high-frequency noise) by increasing the gain with a slope of 20 dB/dec. For the integral action, the positive effects (elimination of steady-state errors) can be deduced by the infinite gain at zero frequency (integral effect), and the negative ones (decrease of the relative stability) by the $\pi/2$ phase lag introduced. Considering this, it is quite natural to conclude that, by introducing more general control actions of the form s^{α}, $1/s^{\alpha}$, $\alpha \in \mathbb{R}^{+}$, we could achieve more satisfactory compromises between their positive and negative effects, and we could develop more powerful and flexible design methods to satisfy the controlled system specifications by combining these actions.

B.M. Vinagre (✉)
University of Extremadura, Campus Universitario, 06006 Badajoz, Spain
e-mail: bvinagre@unex.es

C.A. Monje
University Carlos III of Madrid, Avenida Universidad 30, 28911 Leganés, Madrid, Spain
e-mail: cmonje@ing.uc3m.es

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_15, © Springer-Verlag London Limited 2012

These generalized operators, arising in a quite natural way in the frequency domain, lead us to the definition of the differential and integral operators of arbitrary order, which are the fundamental operators of the *Fractional Calculus (FC)*, a generalization of the classical calculus to orders of integration and differentiation not necessarily integer. The use of this mathematical tool in different techniques and methodologies of feedback control, the *Fractional-order Control (FOC)*, has received growing attention in the last decades, and the application of the fractional-order operators to the *PID* algorithm gives us the *Fractional-order PID (FoPID)*, one of the subjects deserving more interest in *FOC*.

In this chapter, after introducing the fundamental definitions of the *FC*, the *FoPID* will be studied in both the frequency and the time domains, and the structures, tuning rules, and ways for its implementation will be revised and commented, as well as their practical applications.

15.2 Fractional Calculus and Fractional-Order Systems

15.2.1 Brief Historical Overview of Fractional Calculus

Students of mathematics, sciences, and engineering know the differential operators d/dx, d^2/dx^2, etc., but probably few of them ponder over whether it is necessary for the order of differentiation to be an integer. Why not take a rational, fractional, irrational, or even a complex number? At the very beginning of integral and differential calculus, in a letter to L'Hôpital in 1695, Leibniz himself raised the question for a non-integer-order derivative, and this question was the origin of an ongoing topic for more than 300 years which now is known as *Fractional Calculus*, a generalization of ordinary differentiation and integration to arbitrary (non-integer) order.

The earliest theoretical contributions to the field were made by Euler and Lagrange in the eighteenth century, and the first systematic studies seem to have been made at the beginning and middle of the nineteenth century by Liouville, Riemann, and Holmgren. It was Liouville who expanded functions in series of exponentials and defined the nth-order derivative of such a series by operating term-by-term as if n were a positive integer. Riemann proposed a different definition that involved a definite integral and was applicable to power series with non-integer exponents. Grünwald and Krug were the first to unify the results from Liouville and Riemann. Grünwald did it by returning to the original sources and adopting as a starting point the definition of a derivative as the limit of a difference quotient and arriving at definite-integral formulas for the nth-order derivative. Krug, working through Cauchy's integral formula for ordinary derivatives, showed that Riemann's definite integral had to be interpreted as having a finite lower limit, while Liouville's definition corresponded to a lower limit of $-\infty$.

The first application of *FC* was made by Abel in 1823. He discovered that the solution of the integral equation for the tautochrone problem could be obtained via an integral in the form of a derivative of order $1/2$. Later in the nineteenth century, important stimuli to the use of *FC* were provided by Boole's development of symbolic methods for solving linear differential equations with constant coefficients, or the operational calculus of Heaviside developed to solve certain problems in electromagnetic theory, such as transmission lines. In the twentieth century, contributions have been made to both the theory and the applications of *FC* by very well known scientists such as Weyl and Hardy (properties of differintegrals), Erdély (integral equations), Riesz (functions of more than one variable), Scott Blair (rheology), or Oldham and Spanier (electrochemistry and general transport problems).

In the closing decades of the last century, there was a continuing growth of the applications of *FC* mainly promoted by the engineering applications in the fields of feedback control, systems theory, and signals processing.

The interested reader can find good surveys of the history of *FC* in [30, 47, 53].

15.2.2 Fractional-Order Operators: Definitions and Properties

In the literature, there are several definitions for fractional integrals and derivatives (see [61]). Among them, the most important ones for the purpose of this chapter are presented next.

For any $\alpha \in \mathbb{R}^+$, *Riemann–Liouville's definition of the fractional-order integral*, that can be seen as a natural consequence of Cauchy's formula for repeated integrals, can be expressed as

$$\mathscr{I}_c^{\alpha} f(t) \triangleq \frac{1}{\Gamma(\alpha)} \int_c^t (t-\tau)^{\alpha-1} f(\tau)\, d\tau, \quad t > c,\ \alpha \in \mathbb{R}^+. \tag{15.1}$$

When we deal with dynamic systems, it is usual that $f(t)$ is a causal function of t, and so, in what follows, the definition of the fractional-order integral to be used will consider 0 as the lower limit of the integral.

The definition in (15.1) cannot be used for the fractional-order derivative by direct substitution of α by $-\alpha$ because we have to proceed carefully in order to guarantee the convergence of the integrals involved in the definition and to preserve the properties of the ordinary derivative of integer order. After some subtle mathematical considerations, and introducing the positive integer m so that $m-1 < \alpha < m$, *Riemann–Liouville's definition of the fractional-order derivative* of order $\alpha \in \mathbb{R}^+$ has the following form:

$${}_R\mathscr{D}^{\alpha} f(t) \triangleq \mathscr{D}^m \mathscr{I}^{m-\alpha} f(t) = \frac{d^m}{dt^m}\left[\frac{1}{\Gamma(m-\alpha)} \int_0^t \frac{f(\tau)}{(t-\tau)^{\alpha-m+1}}\, d\tau\right], \tag{15.2}$$

where $m-1 < \alpha < m$, $m \in \mathbb{N}$.

An alternative definition of the fractional-order derivative was introduced by Caputo as

$${}_C\mathscr{D}^{\alpha} f(t) \triangleq \mathscr{I}^{m-\alpha} \mathscr{D}^m f(t) = \frac{1}{\Gamma(m-\alpha)} \int_0^t \frac{f^{(m)}(\tau)}{(t-\tau)^{\alpha-m+1}}\, d\tau, \tag{15.3}$$

where $m-1<\alpha<m, m\in\mathbb{N}$.

There are the following relations between these two definitions in (15.2) and (15.3):

$$ {}_R\mathscr{D}^{\alpha} f(t) = {}_C\mathscr{D}^{\alpha} f(t) + \sum_{k=0}^{m-1} \frac{t^{k-\alpha}}{\Gamma(k-\alpha+1)} f^{(k)}\big(0^+\big), \tag{15.4}$$

$$ {}_R\mathscr{D}^{\alpha}\left(f(t) - \sum_{k=0}^{m-1} f^{(k)}\big(0^+\big)\frac{t^k}{k!}\right) = {}_C\mathscr{D}^{\alpha} f(t). \tag{15.5}$$

Due to its importance in applications, we will consider here Grünwald–Letnikov's definition, based on the generalization of the backward difference. This definition has the form

$$ \mathscr{D}^{\alpha} f(t)|_{t=kh} = \lim_{h\to 0}\frac{1}{h^{\alpha}}\sum_{j=0}^{k}(-1)^j\binom{\alpha}{j} f(kh-jh). \tag{15.6}$$

The Laplace integral transform is a fundamental tool in systems and control engineering. For this reason, we will give the equivalents of the defined fractional-order operators in the Laplace domain. These equivalents are:

$$ \mathscr{L}\big[\mathscr{I}^{\alpha} f(t)\big] = s^{-\alpha}F(s), \tag{15.7}$$

$$ \mathscr{L}\big[{}_R\mathscr{D}^{\alpha} f(t)\big] = s^{\alpha}F(s) - \sum_{k=0}^{m-1} s^k\big[{}_R\mathscr{D}^{\alpha-k-1} f(t)\big]_{t=0}, \tag{15.8}$$

$$ \mathscr{L}\big[{}_C\mathscr{D}^{\alpha} f(t)\big] = s^{\alpha}F(s) - \sum_{k=0}^{m-1} s^{\alpha-k-1} f^{(k)}(0). \tag{15.9}$$

15.2.3 Fractional-Order Systems

15.2.3.1 Models

Based on the operators introduced previously, the equations for a continuous-time dynamic system of fractional order can be written as follows:

$$ H\big(\mathscr{D}^{\alpha_0\alpha_1\alpha_2\cdots\alpha_m}\big)(y_1,y_2,\ldots,y_l) = G\big(\mathscr{D}^{\beta_0\beta_1\beta_2\cdots\beta_n}(u_1,u_2,\ldots,u_k)\big), \tag{15.10}$$

where y_i, u_i are functions of time and $H(\cdot)$,$G(\cdot)$ are the combination laws of the fractional-order derivative operator. For the linear time-invariant single-variable case, the following equation would be obtained:

$$\begin{aligned} &a_n \mathscr{D}^{\alpha_n} y(t) + a_{n-1} \mathscr{D}^{\alpha_{n-1}} y(t) + \cdots + a_0 \mathscr{D}^{\alpha_0} y(t) \\ &\quad = b_m \mathscr{D}^{\beta_m} u(t) + b_{m-1} \mathscr{D}^{\beta_{m-1}} u(t) + \cdots + b_0 \mathscr{D}^{\beta_0} u(t). \end{aligned} \tag{15.11}$$

If in the previous equation all the orders of derivation are integer multiples of a base order α, that is, $\alpha_k = k\alpha$, $\beta_k = k\alpha$, $\alpha \in \mathbb{R}^+$, the system will be of *commensurate order*, and (15.11) becomes

$$\sum_{k=0}^{n} a_k \mathscr{D}^{k\alpha} y(t) = \sum_{k=0}^{m} b_k \mathscr{D}^{k\alpha} u(t). \tag{15.12}$$

If in (15.12) $\alpha = 1/q$, $q \in \mathbb{Z}^+$, the system will be of *rational order*.

In the case of discrete-time systems (or discrete equivalents of continuous-time systems), we can obtain models of the form

$$\begin{aligned} &a_n \Delta_h^{\alpha_n} y(t) + a_{n-1} \Delta_h^{\alpha_{n-1}} y(t) + \cdots + a_0 \Delta_h^{\alpha_0} y(t) \\ &\quad = b_m \Delta_h^{\beta_m} u(t) + b_{m-1} \Delta_h^{\beta_{m-1}} u(t) + \cdots + b_0 \Delta_h^{\beta_0} u(t), \end{aligned} \tag{15.13}$$

where Δ_h^{γ} denotes the difference operator with step size h and order γ.

Applying the Laplace transform to (15.11) with zero initial conditions, or the Z transform to (15.13), the input–output representations of the fractional-order systems can be obtained. In the case of continuous models, a fractional-order system will be given by a transfer function of the form

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \cdots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \cdots + a_0 s^{\alpha_0}}. \tag{15.14}$$

In the case of discrete-time systems, the discrete-time transfer function will be of the form

$$G\big(z^{-1}\big) = \frac{b_m (\omega(z^{-1}))^{\beta_m} + b_{m-1} (\omega(z^{-1}))^{\beta_{m-1}} + \cdots + b_0 (\omega(z^{-1}))^{\beta_0}}{a_n (\omega(z^{-1}))^{\alpha_n} + a_{n-1} (\omega(z^{-1}))^{\alpha_{n-1}} + \cdots + a_0 (\omega(z^{-1}))^{\alpha_0}}, \tag{15.15}$$

where $(\omega(z^{-1}))$ is the Z transform of the operator Δ_h^1, or, in other words, the discrete equivalent of the Laplace operator, s.

As can be seen in the previous equations, a fractional-order system has an irrational-order transfer function in the Laplace domain or a discrete transfer function of unlimited order in the Z domain, since only in the case of $\alpha_k \in \mathbb{Z}$, there will be a limited number of coefficients $(-1)^l \binom{\alpha_k}{l}$ different from zero. Because of this, it can be said that a fractional-order system has an unlimited memory or is infinite-dimensional, and obviously the systems of integer order are just particular cases.

In the case of a commensurate-order system, the continuous-time transfer function is given by

$$G(s)=\frac{\sum_{k=0}^{m} b_k (s^\alpha)^k}{\sum_{k=0}^{n} a_k (s^\alpha)^k}. \tag{15.16}$$

15.2.3.2 Dynamic Behavior and Stability

In general, the study of the stability of fractional-order systems can be carried out by studying the solutions of the differential equations that characterize them. An alternative way is the study of the transfer function of the system (15.14). To carry out this study, it is necessary to remember that a function of the type

$$a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \cdots + a_0 s^{\alpha_0}, \tag{15.17}$$

with $\alpha_i \in \mathbb{R}^+$, is a multi-valued function of the complex variable s whose domain can be seen as a Riemann surface [25, 82] of a number of sheets which is finite only in the case of $\forall i$, $\alpha_i \in \mathbb{Q}^+$, the principal sheet being defined by $-\pi < \arg(s) < \pi$. In the case of $\alpha_i \in \mathbb{Q}^+$, that is, $\alpha = 1/q$, q being a positive integer, the q sheets of the Riemann surface are determined by

$$s = |s|e^{j\phi}, \quad (2k+1)\pi < \phi < (2k+3)\pi,\ k = -1, 0, \ldots, q-2. \tag{15.18}$$

Correspondingly, the case of $k = -1$ is the *principal sheet*. For the mapping $w = s^\alpha$, these sheets become the regions of the plane w defined by

$$w = |w|e^{j\theta}, \quad \alpha(2k+1)\pi < \theta < \alpha(2k+3)\pi. \tag{15.19}$$

This mapping is illustrated in Figs. 15.1 and 15.2 for the case of $w = s^{1/3}$. Figure 15.1 represents the Riemann surface that corresponds to the transformation introduced above, and Fig. 15.2 represents the regions of the complex plane w that correspond to each sheet of the Riemann surface. These three sheets correspond to

$$k = \begin{cases} -1, & -\pi < \arg(s) < \pi \text{ (the principal sheet)}, \\ 0, & \pi < \arg(s) < 3\pi \text{ (sheet 2)}, \\ 1\ (= 3-2), & 3\pi < \arg(s) < 5\pi \text{ (sheet 3)}. \end{cases}$$

Thus, an equation of the type

$$a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \cdots + a_0 s^{\alpha_0} = 0, \tag{15.20}$$

which in general is not a polynomial, will have an infinite number of roots, among which only a finite number of them will be on the principal sheet of the Riemann surface. It can be said that the roots which are in the secondary sheets are related to time domain solutions (or responses) that are always monotonically decreasing

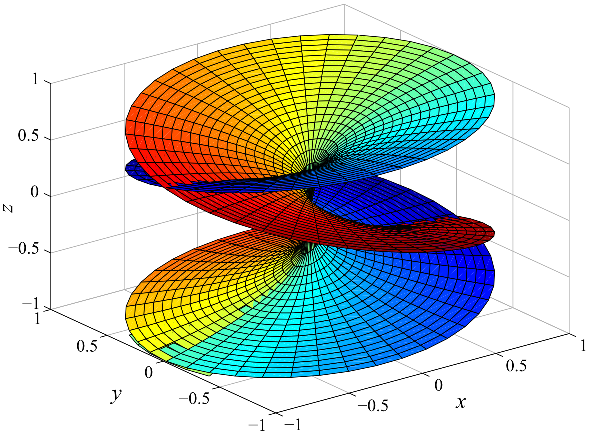


Fig. 15.1 Riemann surface for $w = s^{1/3}$

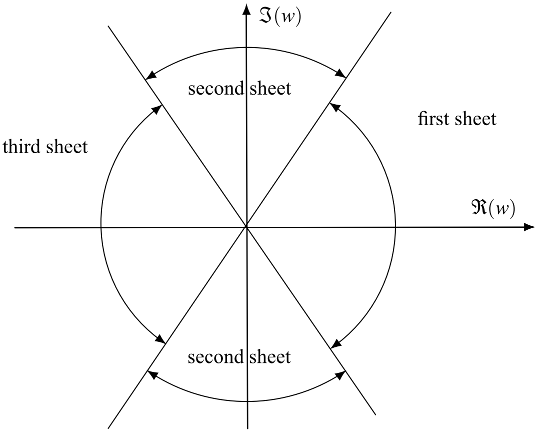


Fig. 15.2 w-plane regions corresponding to the Riemann surface for $w = s^{1/3}$

functions (they go to zero without oscillations when $t \to \infty$), and only the roots in the principal sheet are responsible for a different dynamics: damped oscillation, oscillation of constant amplitude, or oscillation of increasing amplitude with monotonic growth.

In general, it can be said that a fractional-order system, with an irrational-order transfer function $G(s) = P(s)/Q(s)$, is bounded-input bounded-output (BIBO) stable if and only if the following condition is fulfilled (see [46] for more details):

$$\exists M, \quad \left|G(s)\right| \leq M, \quad \forall s \quad \Re(s) \geq 0. \tag{15.21}$$

The previous condition is satisfied if all the roots of $Q(s) = 0$ in the principal Riemann sheet, not being roots of $P(s) = 0$, have negative real parts. For the case of commensurate-order systems, whose characteristic equation is a polynomial of the complex variable $\lambda = s^{\alpha}$, the stability condition is expressed as

$$\left|\arg(\lambda_i)\right| > \alpha\frac{\pi}{2}, \tag{15.22}$$

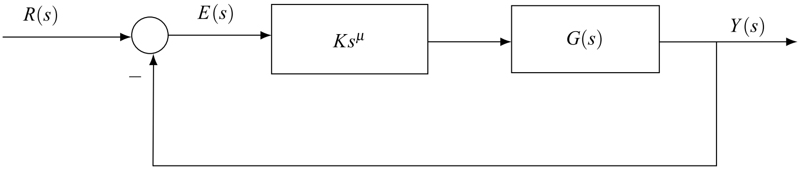


Fig. 15.3 Block diagram of a closed-loop system with fractional-order control actions

where λ_i are the roots of the characteristic polynomial in λ. For the particular case of $\alpha = 1$, the well known stability condition for linear time-invariant systems of integer order is recovered:

$$\left|\arg(\lambda_i)\right| > \frac{\pi}{2}, \quad \forall \lambda_i / Q(\lambda_i) = 0. \tag{15.23}$$

Nowadays we can find interesting studies on the stability of fractional-order systems. There are even some attempts to develop polynomial techniques, either Routh or Jury type, to analyze their stability. Of course, we can always use the geometric techniques of complex analysis based on Cauchy's argument principle, since they inform us about the number of singularities of the function within a rectifiable curve by observing the evolution of the function argument through this curve. For more details about the stability of fractional-order systems, see [12, 31, 37, 46, 58, 79].

15.3 Fractional-Order Control

15.3.1 Generalized Fractional-Order Control Actions

Starting from the block diagram of Fig. 15.3 (see, e.g., [2, 10]), the effects of the generalized basic control actions of type Ks^{μ} for $\mu \in [-1, 1]$ will be described in this section. The basic control actions traditionally considered will be particular cases of this general case, in which $\mu = 0$ for the proportional action, $\mu = -1$ for the integral action, and $\mu = 1$ for the derivative action.

As is known, the main effects of the integral action are those that make the system slower, decrease its relative stability, and eliminate the steady-state error for inputs for which the system has a finite error.

These effects can be observed in different domains. In the time domain, the effects on the transient response consist of the decrease of the rise time and the increase of the settling time and the overshoot. In the complex plane, the effects of the integral action consist of a displacement of the root locus of the system towards the right half-plane. Finally, in the frequency domain, these effects consist of an increase of -20 dB/dec in the slopes of the magnitude curves and a decrease of $\pi/2$ rad in the phase plots. In the case of a fractional-order integral, that is, $\mu \in (-1, 0)$, the selection of the value of μ needs consideration of the effects mentioned above. In the

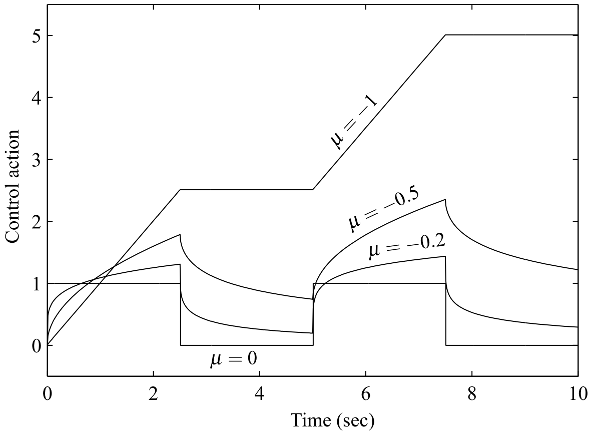


Fig. 15.4 Integral control action for a square error signal and $\mu = 0, -0.2, -0.5, -1$

time domain, the effects of the control action can be studied considering the effects of this action on a squared error signal. If the error signal has the form

$$e(t) = \sum_{k=0}^{N} (-1)^k u_0(t - kT), \quad k = 0, 1, 2, \ldots, N, \tag{15.24}$$

where $u_0(t)$ is the unit step, its Laplace transform is

$$E(s) = \sum_{k=0}^{N} (-1)^k \frac{e^{-kTs}}{s}. \tag{15.25}$$

Thus, the control action, as shown in the block diagram of Fig. 15.3, will be given by

$$\begin{aligned} u(t) &= \mathscr{L}^{-1}\{U(s)\} = \mathscr{L}^{-1}\left\{ K \sum_{k=0}^{N} (-1)^k \frac{e^{-kTs}}{s^{1-\mu}} \right\} \\ &= K \sum_{k=0}^{N} \frac{(-1)^k}{\Gamma(1-\mu)} (t - kT)^{-\mu} u_0(t - kT). \end{aligned} \tag{15.26}$$

Figure 15.4 shows the function $u(t)$ for the values $\mu = 0, -0.2, -0.5, -1$; $T = 30$; $N = 4$. As can be observed, the effects of the control action on the error signal vary between the effects of a proportional action ($\mu = 0$, square signal) and an integral action ($\mu = -1$, straight lines curve). For intermediate values of μ, the control action increases for a constant error, which results in the elimination of the steady-state error, and decreases when the error is zero, resulting in a more stable system.

In the complex plane, the root locus of the system with the control action is governed by

$$1 + Ks^{\mu} G(s) = 0, \tag{15.27}$$

or by the following equivalent conditions for the magnitude and phase:

$$|K| = \frac{1}{|s^{\mu}||G(s)|}, \tag{15.28}$$

$$\arg\bigl[s^{\mu}G(s)\bigr] = (2n+1)\pi, \quad l = 0, \pm 1, \pm 2, \ldots. \tag{15.29}$$

Taking into account that

$$s = |s|e^{j\phi} \quad \Longrightarrow \quad s^{\mu} = |s|^{\mu}e^{j\mu\phi}, \tag{15.30}$$

the conditions of magnitude and phase can be expressed by

$$|K| = \frac{1}{|s|^{\mu}|G(s)|}, \tag{15.31}$$

$$\arg\bigl[s^{\mu}G(s)\bigr] = \arg\bigl[G(s)\bigr] + \mu\phi = (2n+1)\pi, \quad l = 0, \pm 1, \pm 2, \ldots. \tag{15.32}$$

The selection of the value of $\mu \in (-1, 0)$ affects the displacement of the root locus towards the right half-plane and the values of K to reach the magnitude condition. In the frequency domain, the magnitude curve is given by

$$20\log\bigl|s^{\mu}G(s)\bigr|_{s=j\omega} = 20\log\bigl|G(j\omega)\bigr| + 20\mu\log\omega, \tag{15.33}$$

and the phase one by

$$\arg\bigl[s^{\mu}G(s)\bigr]_{s=j\omega} = \arg\bigl[G(s)\bigr] + \mu\frac{\pi}{2}. \tag{15.34}$$

By continuously varying the value of μ between -1 and 0, it is possible to introduce a constant increment in the slopes of the magnitude curve that varies between -20 dB/dec and 0 dB/dec, and to introduce a constant lag in the phase plot that varies between $-\pi/2$ rad and 0 rad.

Regarding the derivative action, it is known that it increases the stability of the system and tends to emphasize the effects of noise at high frequencies. In the time domain, a decrease in the overshoot and the settling time is observed. In the complex plane, the derivative action produces a displacement of the root locus of the system towards the left half-plane. In the frequency domain, this action produces a constant phase lead of $\pi/2$ rad and an increase of 20 dB/dec in the slopes of the magnitude curves. Following a procedure similar to that for the integral action, it is easy to prove that all these effects can be weighted by the selection of the order of the derivative action, that is, by selecting $\mu \in (0, 1)$. In the time domain, the effects of the derivative control action can be studied considering the effects of this action on a trapezoidal error signal given by

$$e(t) = t\,u_0(t) - t(t-T)\,u_0(t-T) - t(t-2T)\,u_0(t-2T) + t(t-3T)\,u_0(t-3T),$$

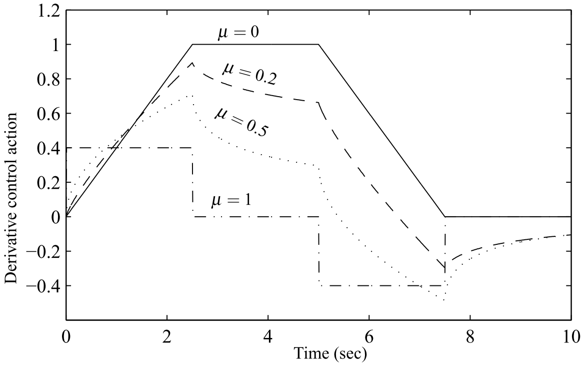


Fig. 15.5 Derivative control action for a trapezoidal error signal and $\mu = 0, 0.2, 0.5, 1$

whose Laplace transfer function is

$$E(s) = \frac{1}{s^2} - \frac{e^{-Ts}}{s^2} - \frac{e^{-2Ts}}{s^2} + \frac{e^{-3Ts}}{s^2}. \tag{15.35}$$

Therefore, and according to Fig. 15.3, the control action will be given by

$$\begin{aligned} u(t) &= \mathscr{L}^{-1}\{U(s)\} = \mathscr{L}^{-1}\left\{K\left(\frac{1}{s^{2-\mu}} - \frac{e^{-Ts}}{s^{2-\mu}} - \frac{e^{-2Ts}}{s^{2-\mu}} + \frac{e^{-3Ts}}{s^{2-\mu}}\right)\right\} \\ &= \frac{K}{\Gamma(2-\mu)}\big\{t^{1-\mu}u_0(t) - (t-T)^{1-\mu}u_0(t-T) \\ &\quad - (t-2T)^{1-\mu}u_0(t-2T) + (t-T)^{1-\mu}u_0(t-3T)\big\}. \end{aligned} \tag{15.36}$$

The effects of the control action on the error signal are shown in Fig. 15.5 and vary between the effects of a proportional action ($\mu = 0$, trapezoidal signal) and a derivative action ($\mu = 1$, square signal). For intermediate values of μ, the control action corresponds to intermediate curves. It must be noted that the derivative action is not zero for a constant error and the growth of the control signal is more damped when a variation in the error signal occurs, which implies a better attenuation of high-frequency noise signals.

In the frequency domain, the magnitude curve is given by (15.33) and the phase plot by (15.34). As can be observed, by varying the value of μ between 0 and 1, it is possible to introduce a constant increment in the slopes of the magnitude curve that varies between 0 dB/dec and 20 dB/dec, and to introduce a constant delay in the phase plot that varies between 0 rad and $\pi/2$ rad.

15.3.2 State-of-the-Art in FOC

Maybe the first sign of the potential of FOC, though without using the term "fractional," emerged with Bode (see [7, 8]). A key problem in the design of a feedback amplifier was to devise a feedback loop so that the performance of the closed loop was invariant to changes in the amplifier gain. Bode presented an elegant solution to this robust design problem, which he called the *ideal cutoff characteristic*, nowadays known as *Bode's ideal loop transfer function*, whose Nyquist plot is a straight line through the origin, giving a phase margin invariant to gain changes. Clearly, this ideal system is, from our point of view, a fractional-order integrator with transfer function $G(s) = (\omega_{cg}/s)^{\alpha}$, known as *Bode's ideal transfer function*, where ω_{cg} is the gain crossover frequency and the constant phase margin is $\varphi_m = \pi - \alpha\pi/2$. This frequency characteristic is very interesting in terms of robustness of the system to parameter changes or uncertainties, and several design methods have made use of it. In fact, the fractional-order integrator can be used as an alternative reference system for control [80].

This first step towards the application of FC in control led to the adaptation of the FC concepts to frequency-based methods. The frequency response and the transient response of the non-integer-order integral (in fact, Bode's loop ideal transfer function) and its application to control systems were introduced by Manabe [45], and more recently in [4].

In the 1980s and 1990s, Oustaloup and Podlubny presented important studies on *FOC* strategies, which established the effective starting point of FC in automatic control applications. In particular, Oustaloup [54] proposed the CRONE (French abbreviation for *Commande Robuste d'Odre Non Entier*, meaning Non-integer Order Robust Control) method for the control of dynamic systems, having now three generations of CRONE controllers (see [35]). With respect to Podlubny's work, he introduced the generalization of the traditional *PID* controller to non-integer orders, namely the PI$^{\lambda}$D$^{\mu}$ controller, where λ and μ are the orders of the integrator and differentiator, respectively [62]. Podlubny also demonstrated the better response of these kinds of controllers in comparison with the classical ones, especially to control fractional-order systems.

During the last decades, further research activities to define new effective tuning methods for fractional order controllers have been proposed in the literature as an extension of the classical control theory, mainly for traditional *PID* controllers due to their widespread industrial use. A classification of these tuning techniques is presented in [77]. These tuning methods are mainly based on techniques in the frequency domain but also on optimizing certain performance indices or providing the controlled system with extra specifications given by the additional tuning parameters of fractional order controllers with respect to classical ones.

FC also extends to other kinds of control strategies different from *PID* ones. For instance, a robust sliding-mode control is developed for a coupled tank [26], an inverted pendulum [27], and a DC-DC buck converter [32]; in [65, 66] fractional order generalized predictive controllers are designed to control the velocity and position of a servomotor; fuzzy fractional and adaptive *PID* controllers are investigated in [3] and [28], respectively; tuning methods for the synthesis of robust fractional-order controllers based on Quantitative Feedback Theory (QFT) are proposed in [14, 33], and [15, 75] and [59] propose the design of fractional order controllers by minimizing H_2 and H_∞ norms; fractional gain scheduled controllers are suggested to control irrigation canals with variable transport delay and time constant [36]. Furthermore, the potential of the applications of variable-order fractional controllers are discussed in [69] and [78], referring to physical experimental and simulation studies, respectively, and in [71], focusing on the compensation of the effects of time-varying network delays.

The range of applications of FOC is wide: car active suspensions, electrical circuits, hydraulic actuators, autonomous vehicles, flexible manipulators, rigid robots, irrigation canals, servomotors, and so on. Refer to [49] and [23] for a comprehensive and current review of FOC and its applications, as well as for information on how to simulate fractional order systems and controllers with MATLAB and realization techniques.

Although the range of design techniques and applications presented here is quite far from aiming at completeness, it is clear that a renewed interest has been devoted to FOC, becoming an important research topic. As a matter of fact, it is important to remark that, among the 160 papers presented in the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10), held in October 2010, more than 40 were in the area of FOC (see [64]).

15.4 The Fractional-Order PID Controller

15.4.1 Definitions and Characteristics

This section introduces a more generalized structure for the classical integer-order *PID* controller, keeping the simplicity of its formulation and making use of the generalized derivative and integral control actions described above. In order to show the characteristics and possibilities of the application of the so-called fractional-order PID controller (*FoPID*), a comparison with the standard *PID* will be given in the frequency domain.

The classical *PID* controller can be considered as a particular form of lead–lag compensation in the frequency domain. Its transfer function can be expressed as

$$C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s, \tag{15.37}$$

or

$$C(s) = k\frac{(s/\omega_c)^2 + 2\delta_c s/\omega_c + 1}{s}, \tag{15.38}$$

with $\omega_c = \sqrt{K_i/K_d}$, $\delta = K_p/(2\sqrt{K_i K_p})$, $k = K_i$. Another form can be

$$C(s) = k\frac{(s+a)(s+b)}{s}. \tag{15.39}$$

Therefore, the contributions of the controller depend on one of the following:

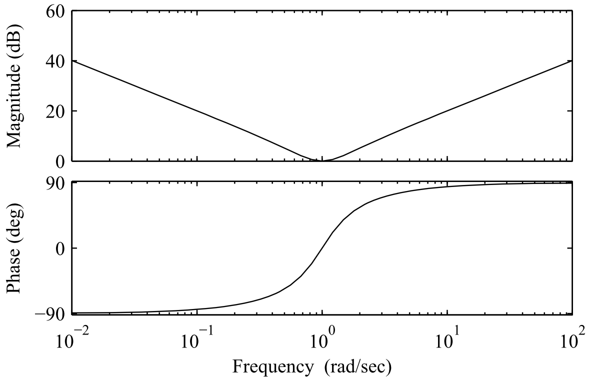


Fig. 15.6 Frequency response of the classical *PID* controller with $K_p = K_i = K_d = 1$

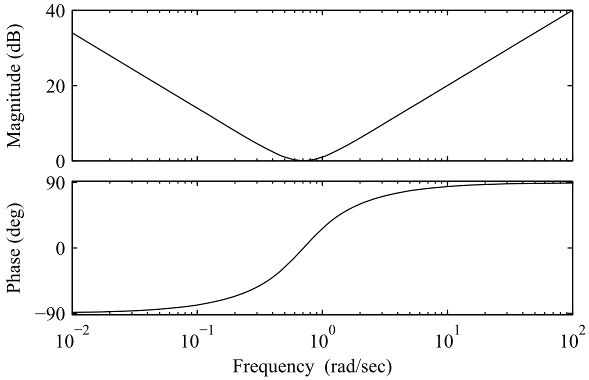


Fig. 15.7 Frequency response of the classical *PID* controller with $K_p = 1$, $K_i = 0.5$, $K_d = 1$

- Gains K_p, K_i, K_d.
- Gain k and parameters ω_c, δ_c.
- Gain k and location of zeros a and b.

In the frequency response of the controller, the selection of these gains or parameters is equivalent to the selection of the position, smoothness, and minimum value of the magnitude curve and the slope of the phase plot of the controller at the frequency of this minimum. However, at high and low frequencies the values of the slopes of the magnitude curve and the values of the contributions in phase are fixed. This is illustrated in Fig. 15.6 for $K_p = K_i = K_d = 1$ and Fig. 15.7 for $K_p = 1$, $K_i = 0.5$, $K_d = 1$. Comparing these two figures, it is observed that both the value and position of the magnitude minima and the inflection point of the phase plot are modified by the value of K_i, whilst the slopes of the magnitude curves and the asymptotic values of the phase plots remain the same.

In 1994, a patent [41] proposed a new three term controller, the TID controller, in which the proportional term of the *PID* was replaced by a “tilt” term with transfer function of the form $s^{\frac{1}{n}}$. The argument of the author for introducing this new term in the place of the proportional one was that the resulting transfer function of the controller approximated more closely an optimal transfer function denoted as *Bode’s*

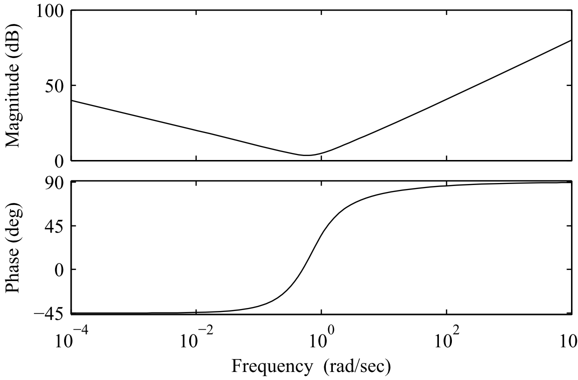


Fig. 15.8 Frequency response of the *FoPID* controller with $k = 1$, $\omega_f = 1$, $\delta_f = 1$, and $\lambda = \mu = 0.5$

ideal compared to the conventional *PID*, and this fact allowed for simpler tuning, better disturbance rejection, and better robustness to plant variations. Though it can be considered as a seminal work, this controller has received little attention. Thus, we will concentrate on the *FoPID* controller.

The integro-differential equation defining the control action of a *FoPID* controller is given by

$$u(t) = K_p e(t) + K_i \mathscr{D}^{-\lambda} e(t) + K_d \mathscr{D}^{\mu} e(t). \tag{15.40}$$

Applying Laplace transform to this equation with null initial conditions, the transfer function of the controller can be expressed by

$$C_f(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu = k \frac{(s/\omega_f)^{\lambda+\mu} + s\delta_f s^\lambda/\omega_f + 1}{s^\lambda}. \tag{15.41}$$

Figure 15.8 shows the frequency response of this controller for $k = 1$, $\omega_f = 1$, $\delta_f = 1$, and $\lambda = \mu = 0.5$.

As can be observed, this fractional-order controller allows us to select both the slope of the magnitude curve and the phase contributions at both high and low frequencies.

In a graphical way, the control possibilities using a *FoPID* controller are shown in Fig. 15.9, extending the four control points of the classical *PID* to the range of control points of the quarter-plane defined by selecting the values of λ and μ (shown as the shaded area in Fig. 15.9).

15.5 Tuning Methods

15.5.1 Introduction

It is important to realize that there is a very wide range of control problems and, consequently, also a need for a wide range of design techniques. There are already

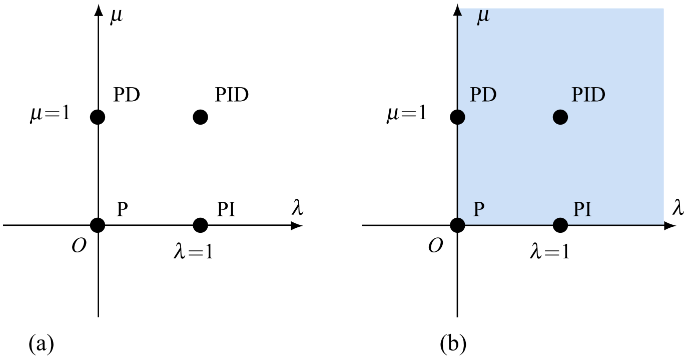


Fig. 15.9 *FoPID* vs classical *PID*: from points to plane: (**a**) integer-order controller and (**b**) fractional-order controller

many tuning methods available in the literature for fractional $\text{PI}^{\lambda}\text{D}^{\mu}$ controllers of the form

$$C(s) = K_p + \frac{K_i}{s^{\lambda}} + K_d s^{\mu}. \tag{15.42}$$

Since this kind of controller has five parameters to tune ($K_p, K_d, K_i, \lambda, \mu$), up to five design specifications for the controlled system can be met, that is, two more than in the case of a conventional *PID* controller, where $\lambda = 1$ and $\mu = 1$. It is essential to study which specifications are more interesting as far as performance and robustness are concerned, since it is the aim to obtain a controlled system robust to uncertainties of the plant model, load disturbances, and high-frequency noise. All these constraints will be taken into account in the tuning technique in order to take advantage of the introduction of the fractional orders.

During the last decades, further research activities to define new effective tuning methods for fractional-order controllers have been proposed as an extension of the classical control theory, mainly for traditional *PID* controllers due to its widespread industrial use. Some analytical methods, concerning phase and gain margins, flat phase, or dominant poles, can be found in [20, 39, 40, 44, 79], as well as some optimization-based methods in [13, 51] and tuning rules in [5, 6, 19, 76]. Recently, tuning methods for *FoPID* controllers based on AI tools such as Adaptive Genetic Algorithms [16] and Particle Swarm Optimization [83], as well as Fuzzy fractional-order PID controllers [3], have been proposed.

Here we will concentrate our attention on some methods that were the basis of many others that have appeared in the literature (see [49, 64, 77] for a review).

15.5.2 Analytical Methods

15.5.2.1 Phase and Gain Margin-Based Methods

In [79], a method is proposed that sets $\lambda = \mu$, imposes a specified gain margin φ_m at a gain crossover frequency ω_{cg}, and a specified gain margin G_m at a phase crossover frequency ω_{cp}. By solving the set of four nonlinear equations

$$K_p + \left(\frac{K_i}{\omega_{cg}^\lambda} + K_d \omega_{cg}^\lambda\right) \cos\frac{\lambda\pi}{2} = \frac{\cos(-\pi + \varphi_m - \arg(G(j\omega_{cg})))}{|G(j\omega_{cg})|}, \tag{15.43}$$

$$\left(-\frac{K_i}{\omega_{cg}^\lambda} + K_d \omega_{cg}^\lambda\right) \sin\frac{\lambda\pi}{2} = \frac{\sin(-\pi + \varphi_m - \arg(G(j\omega_{cg})))}{|G(j\omega_{cg})|}, \tag{15.44}$$

$$K_p + \left(\frac{K_i}{\omega_{cp}^\lambda} + K_d \omega_{cp}^\lambda\right) \cos\frac{\lambda\pi}{2} = \frac{\cos(\pi - \arg(G(j\omega_{cp})))}{G_m|G(j\omega_{cp})|}, \tag{15.45}$$

$$\left(-\frac{K_i}{\omega_{cp}^\lambda} + K_d \omega_{cp}^\lambda\right) \sin\frac{\lambda\pi}{2} = \frac{\sin(\pi - \arg(G(j\omega_{cp})))}{G_m|G(j\omega_{cp})|}, \tag{15.46}$$

the four parameters of the controller, K_p, K_i, K_D, λ, can be obtained.

In [11], the authors propose choosing $\lambda = \mu > 1$ and specifying a phase margin φ_m at a crossover frequency ω_{cg}. This allows us to freely choose one of the controller gains (K_p, K_i, or K_d) and then to determine the other two from the above equations. In [84], the authors further develop the former set of equations for $\lambda \neq \mu$ and plants of the form

$$G(s) = \frac{1}{a_1 s^\alpha + a_2 s^\beta + a_3}. \tag{15.47}$$

In [44], a similar idea is developed, but for fractional PI controllers and first-order plus integration plants.

A new type of specification was introduced in [20] for fractional PI controllers. This three-parameters controller is tuned by combining gain and phase margin specifications with a condition of flat phase around the frequency at which the sensitivity circle touches the Nyquist curve, that is, the condition

$$\frac{d\arg(C(j\omega)G(j\omega))}{d\omega} = 0 \tag{15.48}$$

must be fulfilled.

In [74], the authors show that, by using the Internal Model Control technique, *FoPID* controllers can by obtained. For example, if in Fig. 15.10 the plant $G(s)$ is of the form

$$G(s) = \frac{K}{1 + s^\alpha T} e^{-Ls} \tag{15.49}$$

with $G^*(s)$ being an inverse or pseudoinverse of $G(s)$, $G'(s)$ a model for the plant, and $F(s)$ a filter for reducing the effect of model mismatch and pseudoinversion,

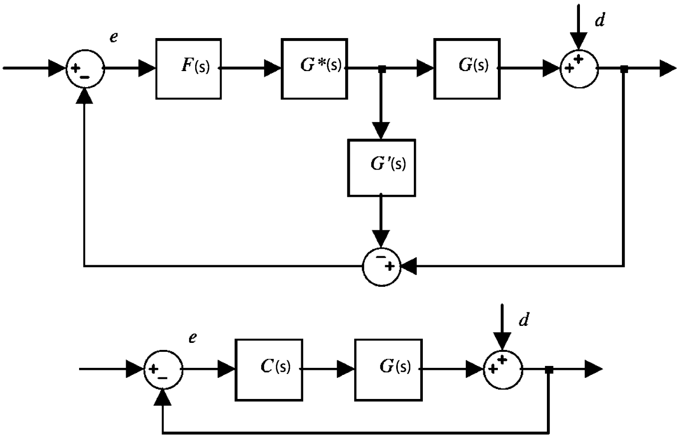


Fig. 15.10 Block diagram for IMC (*top*) and equivalent block diagram (*bottom*)

the resulting controller

$$C(s) = \frac{F(s)G^*(s)}{1 - F(s)G^*(s)G'(s)}, \tag{15.50}$$

can be reduced to a *FoPID*.

15.5.3 Optimization-Based Methods

15.5.3.1 F-Migo Algorithm

A method for tuning fractional PI controllers of the form

$$C(s) = K_p + \frac{K_i}{s^\alpha} \tag{15.51}$$

can be found in [49] with the objective of optimizing the effects of load disturbance with a constraint on the maximum load disturbance-to-output sensitivity M_s. It imposes maximum values of

$$M_s = \max_{0<\omega<\infty} \left|S(j\omega)\right|, \qquad M_p = \max_{0<\omega<\infty} \left|T(j\omega)\right|, \tag{15.52}$$

these functions being the sensitivity and complementary sensitivity functions defined respectively as

$$S(s) = \frac{1}{1 + C(s)G(s)}, \qquad T(s) = \frac{C(s)G(s)}{1 + C(s)G(s)}. \tag{15.53}$$

In this case, the optimization problem is to maximize K_i in order to obtain the controller parameters that guarantee that the closed-loop system is stable and the Nyquist plot of the loop transfer function lies outside the circle with center at $s = -C$ and radius R, C and R being defined by

$$C = \frac{M_s - M_s M_p - 2M_s M_p^2 + M_p^2 - 1}{2M_s(M_p^2 - 1)}, \tag{15.54}$$

$$R = \frac{M_s + M_p - 1}{2M_s(M_p^2 - 1)}. \tag{15.55}$$

The algorithm can be summarized as follows:

- Let the plant to be controlled, $G(s)$, be stable.
- Choose the fractional order α at which you want to find a controller for the system.
- By using a Newton–Raphson technique, solve the equation

$$h(\omega) = 2R\left(\left[C\frac{b}{r} + R\right]\left[\frac{r'}{r} - \frac{\alpha}{\omega}\right] - C\left(\frac{b}{r}\right)'\right) = 0, \tag{15.56}$$

  where r and b are defined in the expression

$$G(j\omega) = a(\omega) + jb(\omega) = r(\omega)e^{j\phi(\omega)}. \tag{15.57}$$

- Calculate the controller gains as

$$K_i = -\frac{R\omega^\alpha}{r\sin\gamma} - \frac{Cb\omega^\alpha}{r^2\sin\gamma}, \tag{15.58}$$

$$K_p = R\frac{\cos\gamma}{\sin\gamma} + \frac{Cb}{r^2}\frac{\cos\gamma}{\sin\gamma} - \frac{Ca}{r^2}, \tag{15.59}$$

  where $\gamma = \pi\alpha/2$.
- Check if the values of M_s and M_p are satisfactory and if the loop is stable.
- Repeat the procedure for the next fractional order and compare the results.

15.5.3.2 Monje's Method

In [51], the objective was to design a fractional-order controller so that the system fulfills different specifications regarding robustness to plant uncertainties, load disturbances, and high-frequency noise. For that reason, specifications related to phase margin, sensitivity functions, and robustness constraints are considered in this design method, due to their important features regarding performance, stability, and robustness. Of course, other kinds of specifications can be met, depending on the particular requirements of the system. Therefore, the design problem is formulated as follows:

- **Phase margin φ_m and gain crossover frequency ω_{cg} specifications**. The equations that define the phase margin and the gain crossover frequency are

$$\left|C(j\omega_{cg})G(j\omega_{cg})\right|_{\mathrm{dB}} = 0 \text{ dB}, \qquad \arg\bigl(C(j\omega_{cg})G(j\omega_{cg})\bigr) = -\pi + \varphi_m. \tag{15.60}$$

- **Robustness to variations in the gain of the plant**. The constraint considered in this case is [21]:

$$\left.\frac{d\arg(F(s))}{d\omega}\right|_{\omega=\omega_{cg}} = 0. \tag{15.61}$$

  This condition forces the phase of the open-loop system $F(s) = C(s)G(s)$ to be flat at ω_{cg}, and so to be almost constant within an interval around ω_{cg}. It means that the system is more robust to gain changes and the overshoot of the response is almost constant within a gain range, also known as the *iso-damping property* of the time response.
- **High-frequency noise rejection**. A constraint on the complementary sensitivity function $T(j\omega)$ can be established:

$$\begin{gathered}\left|T(j\omega) = \frac{C(j\omega)G(j\omega)}{1 + C(j\omega)G(j\omega)}\right|_{\mathrm{dB}} \le A \text{ dB},\\ \forall\omega \ge \omega_t \text{ rad/s} \Rightarrow \left|T(j\omega_t)\right|_{\mathrm{dB}} = A \text{ dB},\end{gathered} \tag{15.62}$$

  with A the desired noise attenuation for frequencies $\omega \ge \omega_t$ rad/s.
- **To ensure a good output disturbance rejection**. A constraint on the sensitivity function $S(j\omega)$ can be defined:

$$\begin{gathered}\left|S(j\omega) = \frac{1}{1 + C(j\omega)G(j\omega)}\right|_{\mathrm{dB}} \le B \text{ dB},\\ \forall\omega \le \omega_s \text{ rad/s} \Rightarrow \left|S(j\omega_s)\right|_{\mathrm{dB}} = B \text{ dB},\end{gathered} \tag{15.63}$$

  with B the desired value of the sensitivity function for frequencies $\omega \le \omega_s$ rad/s (desired frequency range).
- **Steady-state error cancellation**. Properly implemented, a fractional-order integrator of order $k + \alpha$, $k \in \mathbb{N}$, $0 < \alpha < 1$, is, for steady-state error cancellation, as efficient as an integer-order integrator of order $k + 1$ [2].

The design problem is based on solving the system of five nonlinear equations (given by the corresponding design specifications) and five unknown parameters K_p, K_d, K_i, λ, μ. However, the complexity of this set of nonlinear equations is very significant, specially when fractional orders of the Laplace variable s are introduced, and finding the solution is not trivial. In fact, a nonlinear optimization problem must be solved, in which the best solution of a constrained nonlinear equation has to be found. MATLAB optimization toolbox has been used in this work to reach the best solution out with the minimum error. The function used for this purpose is called `fmincon()`, which finds the constrained minimum of a function of several variables. It solves problems of the form $\min_{\boldsymbol{x}} f(\boldsymbol{x})$ subject to $\boldsymbol{C}(\boldsymbol{x}) \le 0$, $\boldsymbol{C}_{eq}(\boldsymbol{x}) = 0$,

$\boldsymbol{x}_m \leq \boldsymbol{x} \leq \boldsymbol{x}_M$, where $f(\boldsymbol{x})$ is the function to minimize; $\boldsymbol{C}(\boldsymbol{x})$ and $\boldsymbol{C}_{eq}(\boldsymbol{x})$ represent the nonlinear inequalities and equalities, respectively (nonlinear constraints); $\boldsymbol{x}$ is the minimum sought; $\boldsymbol{x}_m$ and $\boldsymbol{x}_M$ define a set of lower and upper bounds on the design variables $\boldsymbol{x}$.

In this particular case, the specification in (15.60) referring to the magnitude of the open-loop system is taken as the main function to minimize, and the rest of specifications (15.60)–(15.63) are taken as constraints for the minimization, all of them subject to the optimization parameters defined within the function `fmincon`. The success of this design method depends mainly on the initial starting values (initial conditions) considered for the parameters of the controller.

15.5.4 Tuning Rules

The use of tuning rules based on the previous knowledge of the plant transfer function parameters is a very common design method for *PID* controllers since the publication of the seminal paper by Ziegler and Nichols [85]. In the case of *FoPID* controllers, several researcher have contributed to this area.

Following the reaction curve method for plants with S-shaped step response with transfer function of the form

$$G(s) = \frac{Ke^{-Ls}}{1+sT}, \tag{15.64}$$

in [74, 76] the authors propose several sets of tuning rules conceived from *FoPIDs* tuned as in [50] for obtaining specifications on minimum order (for reference tracking), phase and gain crossover frequencies, phase and gain margins, phase flatness, high frequency noise rejection, and low frequency output disturbance rejection. With these rules, the gains and orders of the controllers are obtained by using polynomials depending on the plant parameters (K, L, T) as

$$K_p = a_0 + a_1 L + b_1 \left(\frac{T}{K}\right) + a_2 L^2 + b_2 \left(\frac{T}{K}\right)^2 + c_1 \left(\frac{TL}{K}\right). \tag{15.65}$$

From the results of a numerical optimization method, in [19] tuning rules are proposed for fractional PI controllers obtaining the controller gains and orders as functions of the plant parameters. Also for fractional PIs, alternative rules are proposed in [5] starting from a similar numerical method.

Following the ultimate cycle Ziegler–Nichols method, in [76] three sets of tuning rules are proposed. Two of them are for achieving the specifications mentioned above (see [50, 51]), and the third one is specific for plants with integration.

15.5.5 Auto-tuning

As is known, there is a wide variety of auto-tuning methods for traditional *PID* controllers. Some of them aim in some way at tuning the robustness of the controlled system [70], for example, forcing the phase of the open-loop system to be flat around the gain crossover frequency so that the system is robust to gain variations [21, 22]. However, the complexity of the equations relating the parameters of the controller increases when some kinds of robustness constraints are required for the controlled system. The implementation of these types of auto-tuning methods for industrial purposes will be really complex since, in general, industrial devices, such as a PLC, cannot solve sets of complex nonlinear equations.

For that reason, an auto-tuning method for fractional-order PI$^\lambda$D$^\mu$ controllers based on the relay test was proposed in [48, 51], that allows the fulfillment of robustness constraints for the controlled system by simple relations among the parameters of the controller, simplifying the later implementation process.

The final aim is to find a method out to auto-tune a fractional-order PI$^\lambda$D$^\mu$ controller formulated as

$$C(s) = K_c x^\mu \left(\frac{\lambda_1 s + 1}{s}\right)^\lambda \left(\frac{\lambda_2 s + 1}{x\lambda_2 s + 1}\right)^\mu. \tag{15.66}$$

The parameters of this controller can be related to those of the standard PI$^\lambda$D$^\mu$ controller given by

$$C_{\mathrm{std}}(s) = K_p \left(1 + \frac{1}{T_i s}\right)^\lambda \left(1 + \frac{T_d s}{1 + T_d s/N}\right)^\mu \tag{15.67}$$

by carrying out some calculations in (15.67), obtaining the following transfer function:

$$C_{\mathrm{std}}(s) = \frac{K_p}{(T_i)^\lambda} \left(\frac{T_i s + 1}{s}\right)^\lambda \left(\frac{T_d(1 + 1/N)s + 1}{1 + T_d s/N}\right)^\mu. \tag{15.68}$$

Comparing (15.66) and (15.67), the relations obtained are $T_i = \lambda_1$, $K_p = k'(\lambda_1)^{-\lambda}$, $N = (1 - x)/x$ and $T_d = \lambda_2(1 - x)$.

As can be observed, this controller has two different parts given by the following equations:

$$\mathrm{PI}^\lambda(s) = \left(\frac{\lambda_1 s + 1}{s}\right)^\lambda, \tag{15.69}$$

$$\mathrm{PD}^\mu(s) = K_c x^\mu \left(\frac{\lambda_2 s + 1}{x\lambda_2 s + 1}\right)^\mu. \tag{15.70}$$

Equation (15.69) corresponds to a fractional-order PI$^\lambda$ controller and (15.70) to a fractional-order lead compensator that can be identified as a PD$^\mu$ controller plus a noise filter. In this method, the fractional-order PI$^\lambda$ controller will be used to cancel the slope of the phase of the plant at the gain crossover frequency ω_{cg}. This way, a flat phase around the frequency of interest is ensured. Once the slope is canceled, the PD$^{\mu}$ controller will be designed to fulfill the design specifications of gain crossover frequency ω_{cg} and phase margin φ_m, following a robustness criterion based on the flatness of the phase curve of this compensator. This way, the resulting phase of the open-loop system will be the flattest possible, ensuring the maximum robustness to plant gain variations.

15.6 Implementation Methods and Computational Tools

15.6.1 Implementation Methods

Once the *PID* controllers have been tuned, the final step for applying them is to find suitable methods for implementing the fractional order operators. These methods have to fulfill several requirements; among them are approximation techniques for obtaining causal, stable, and minimum phase integer-order approximations of the operators in continuous or discrete domains; selection of the adequate frequency range for the approximations; algorithms that take into account issues such as computer memory requirements, computational load, complexity of devices, and others. Furthermore, for the particular case of the fractional integral operator, it is a usual requirement to preserve the integral effect for low frequencies.

The reader can find many methods in the literature for implementing fractional-order operators, both in the continuous and discrete domains. In the continuous domain, the techniques are mainly based on obtaining a rational approximation for the operator s^{α}, $-1 < \alpha < 1$ (there is no sense in approximating $\text{abs}(\alpha) > 1$, because the integer part can be implemented without approximation), and can be classified in the following categories:

- Methods based on mathematical techniques for rational approximation of functions (see [43, 63, 79, 81]). These techniques are based on the approximation of an irrational function, $G(s)$, by a rational one defined by the quotient of two polynomials in the variable s.
- Methods based on the explicit recursive location of poles and zeros of the rational approximation (see [18, 54–56] and references therein).
- Methods based on frequency domain identification, that is, on finding a rational integer-order system whose frequency response fits that of the fractional-order operator.

Due to the nature of the operator s^{α}, $-1 < \alpha < 1$ in the complex plane, almost all these methods, when adequate, lead to rational approximations with poles and zeros interlaced along the negative real axis.

In the discrete domain, there are many proposed techniques, too. They can be divided into two categories:

- Indirect techniques based on the discretization of the rational functions obtained by continuous approximations (see [43, 79], and references therein).
- Direct methods based on:
  - The discretization of the operator s by $s \simeq \omega(z^{-1})$ and then the approximation of a so-called generating function of the form $\omega^{\alpha}(z^{-1})$ corresponding to s^{α} (see [42, 67, 81]).
  - Time domain discretization of the fractional integral and derivative by using numerical techniques (see [38] for the foundations of these techniques).

There are other many proposed techniques that the reader can find in the literature for discrete approximations but, in any case, it is important to remark that, as well as in the continuous case, rational approximations are better and so, in the discrete domain, IIR filters are usually better than FIR filters.

After having found a rational function approximation, the next step is to find a way for implementing it. If the rational function is continuous, we have to look for analog realizations, and we have several options: traditional analog circuits (see [17, 55, 63]) or programmable circuit devices such as switched capacitors or field programmable analog arrays (FPAA) (see [13, 68]). If the rational function is discrete, we can implement the functions by using a computer or any other microprocessor or microcontroller-based device, as well as specific devices as field programmable gate arrays (FPGA) (see [68]).

Finally, we can find some dedicated devices called fractances that can implement the fractional behavior without using approximations [9].

15.6.2 Computational Tools

Almost every researcher on fractional calculus has his/her own computer tools to simulate and design *FOC* systems. Here we only mention the most complete and general of them available for the interested reader. These will be briefly discussed in chronological order of appearance for free acquisition.

The *Ninteger* toolbox [72, 73]. This toolbox for SISO systems implements non-integer controllers both in the frequency and the discrete time domains. Over 30 formulas are available for approximating a non-integer-order derivative on-line. Structures such as non-integer *PIDs* are directly available and second and third-generation CRONE controllers may be found in the software. There are functions for finding norms, identifying models, and plotting frequency diagrams. A graphical interface allows us to choose parameters interactively and to check what the performance will be. This toolbox also includes a Simulink library.

The *CRONE* toolbox [24, 57]. This toolbox, developed by the CRONE team at the University of Bordeaux some years ago for implementing the CRONE strategy, also deals with multiple-input multiple-output plants and includes a graphical interface, but just recently became free and easily available.

The set of tools in [49]. This very recent set of tools includes functions for: implementation of fractional-order operators, both continuous and discrete ones;

frequency response fitting of fractional-order controllers; sub-optimal approximation of fractional-order transfer functions (FOTFs); computation of fractional-order derivatives, integrals and special functions; analytical solutions of linear fractional-order differential equations; MATLAB objects for FOTFs; modeling using FOTFs; stability analysis; time and frequency domain analysis; norm evaluation of FOTFs; block diagrams; optimal controllers design; and others.

Nowadays, one can also find interactive tools for the design of *FoPIDs*. In [29, 60], the authors present a Sysquake interactive software tool for the loop-shaping design of *FoPID* controllers. In particular, this tool allows us to determine automatically the controller parameters by mapping a point of the process Nyquist plot to a point of the loop transfer function Nyquist plot. In this context, constraints on the gain or phase margins or on the maximum sensitivity can be effectively considered. The effects of changing user-chosen parameters can be interactively verified both in the time and the frequency domains.

15.7 Summary and Perspectives

In this chapter, a general overview of the *FoPID* controller has been given. Starting from the mathematical foundations of the fractional integral and differential operators, we have presented the generalization to non-integer orders of the basic control actions, their effects on the controlled system behavior, and the possible advantages of combining them in a closed-form controller, the *FoPID* controller. Some ideas and examples for the controller tuning and implementation, as well as computational tools for the controller design and the fractional-order systems simulation, have been reviewed.

Of course, the main purpose of this general overview was not to be exhaustive (many aspects has been reviewed, but many more can be discussed and explained). By presenting this essential information, the purpose was to stimulate the interest of the readers in the field of fractional-order control, in particular of *FoPID* control, and to make them aware of the fact that much more has to be done in different aspects of this emerging field. Theoretical aspects such as studies in robust stability of closed loops with *FoPIDs* need to be developed, as well as stability tests for fractional-order systems. If we think of the large number of tuning rules for *PID* controllers proposed since the work by Ziegler and Nichols, of the interest they have even nowadays for practical and industrial applications, it is not difficult to infer that, by allowing the orders of integral and derivative parts to be non integer, we can expand both this field of research and the range of the practical applications of the results. Regarding the practical implementations for effective industrial applications, though interesting results have been obtained in [52, 68], or [34], more suitable, robust, and portable algorithms and techniques would be welcome.

References

1. Åström, K.J., Murray, R.M.: Feedback Systems: An Introduction for Scientists and Engineers. Princeton University Press, Princeton (2008)
2. Axtell, M., Bise, E.M.: Fractional calculus applications in control systems. In: Proc. of the IEEE 1990 Nat. Aerospace and Electronics Conf., pp. 563–566 (1990)
3. Barbosa, R.: On linear fuzzy fractional PD and PD + I controllers. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10) (2010)
4. Barbosa, R.S., Tenreiro, J.A., Ferreira, I.M.: A fractional calculus perspective of PID tuning. In: Proceedings of the ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, USA (2003)
5. Bhambhani, V., Chen, Y.Q., Xue, D.: Optimal fractional order proportional integral controller for varying time-delay systems. In: Proceedings of the 17th IFAC World Congress, pp. 4910–4915 (2008)
6. Bhaskaran, T., Chen, Y.Q., Xue, D.: Practical tuning of fractional order proportional and integral controller (1): Tuning rule development. In: Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE'07) (2007)
7. Bode, H.: Relations between attenuation and phase in feedback amplifier design. Bell Syst. Tech. J. **19**, 421–454 (1940)
8. Bode, H.: Network Analysis and Feedback Amplifier Design. Van Nostrand, Princeton (1945)
9. Bohannan, G.W.: Analog fractional order controller in temperature and motor control applications. J. Vib. Control **14**(9–10), 1487–1498 (2008)
10. Calderón Godoy, A.J.: Control fraccionario de convertidores electrónicos de potencia tipo buck (in Spanish). PhD thesis, University of Extremadura (2002)
11. Caponetto, R., Fortuna, L., Porto, D.: A new tuning strategy for a non integer PID controller. In: Proceedings of the 1st IFAC Workshop on Fractional Differentiation and its Applications (FDA'04) (2004)
12. Caponetto, R., Dongola, G., Fortuna, L., Petráš, I.: Fractional Order Systems. Modeling and Control Applications. World Scientific, Singapore (2010)
13. Caponetto, R., Dongola, G., Gallo, A., Giustolisi, G.: Fractional order differ-integral operator in integrated technology. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10), Badajoz, Spain (2010)
14. Cervera, J., Baños, A.: Automatic loop shaping in QFT using CRONE structures. J. Vib. Control **14**(9–10), 1513–1529 (2008)
15. Cervera, J., Baños, A., Monje, C.A., Vinagre, B.M.: Tuning of fractional PID controllers by using QFT. In: Proceedings of the 32nd Annual Conference of the IEEE Industrial Electronic Society (2006)
16. Chang, L.Y., Chen, H.C.: Tuning of fractional PID controller using adaptive genetic algorithm for active magnetic bearing system. WSEAS Trans. Syst. **8**(1), 158–167 (2009)
17. Charef, A., Idiou, D.: Design of analog variable fractional order differentiator. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10) (2010)
18. Charef, A., Sun, H.H., Sao, Y.Y., Onaral, B.: Fractal systems as represented by singularity functions. IEEE Trans. Autom. Control **37**(9), 1465–1470 (1992)
19. Chen, Y.Q., Bhaskaran, T., Xue, D.: Practical tuning rule development for fractional order proportional and integral controllers. J. Comput. Nonlinear Dyn. **3**, 021403 (2008)
20. Chen, Y.Q., Dou, H., Vinagre, B.M., Monje, C.A.: A robust tuning method for fractional order PI controllers. In: Proceedings of the 2nd IFAC Workshop on Fractional Differentiation and its Applications (FDA'06) (2006)
21. Chen, Y.Q., Hu, C.H., Moore, K.L.: Relay feedback tuning of robust PID controllers with isodamping property. In: Proceedings of the 42nd IEEE Conference on Decision and Control, Hawaii, USA, pp. 347–352 (2003)
22. Chen, Y.Q., Moore, K.L., Vinagre, B.M., Podlubny, I.: Robust PID controller autotuning with a phase shaper. In: Proceedings of the First IFAC Workshop on Fractional Differentiation and Its Application, pp. 162–167. ENSEIRB, Bordeaux (2004)

23. Chen, Y.Q., Petráš, I., Xue, D.: Fractional order control—a tutorial. In: Proceedings of the 2009 American Control Conference (ACC'09), pp. 1397–1411 (2009)
24. CRONE Team: The CRONE toolbox for MATLAB (2010). http://www.ims-bordeaux.fr/CRONE/toolbox/
25. Cuadrado, M., Cabanes, R.: Temas de Variable Compleja. Servicio de Publicaciones de la ETSIT UPM, Madrid (1989)
26. Delavari, H., Ranjbar, A., Ghaderi, R., Salehi, E.: Robust fractional sliding mode control for a coupled tank. In: Proceedings of the 3rd IFAC Workshop on Fractional Differentiation and Its Applications (FDA'08) (2008)
27. Delavari, H., Faieghi, M.R., Ranjbar, A.: Fractional-order sliding mode controller for inverted pendulum. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10) (2010)
28. Delavari, H., Ghaderi, R., Ranjbar, A., HosseinNia, S., Momani, S.: Adaptive fractional PID controller for robot manipulator. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10) (2010)
29. Dormido, S., Pisoni, E., Visioli, A.: An interactive tool for loop-shaping design of fractional-order PID controllers. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10) (2010)
30. Dugowson, S.: Les différentielles métaphysiques: histoire et philosophie de la généralisation de l'ordre de dérivation. PhD thesis, University of Paris (1994)
31. Dzieliński, A., Sierociuk, D.: Stability of discrete fractional order state-space systems. J. Vib. Control **14**, 1543–1556 (2008)
32. HosseinNia, S.H., Vinagre, B.M.: Direct boolean integer and fractional order SMC of switching systems: Application to a DC-DC buck converter. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10) (2010)
33. Kalla, R., Nataraj, P.S.V.: Synthesis of fractional-order QFT controllers using interval constraint satisfaction technique. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10) (2010)
34. Lanusse, P., Sabatier, J.: PLC implementation of CRONE controller. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10), Badajoz, Spain (2010)
35. Lanusse, P., Sabatier, S., Moreau, X., Oustaloup, A.: Robust control. Technical report, Tutorial Workshop on Fractional Calculus Applications in Automatic Control and Robotics, 41st IEEE Conference on Decision and Control (CDC'02) (2002)
36. Lavado, F., Tejado, I., Vinagre, B.M.: Gain scheduled fractional order PI controller for irrigation canals with variable dynamics. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10) (2010)
37. Li, Y., Chen, Y.Q., Podlubny, I.: Mittag-Leffler stability of fractional order nonlinear dynamic systems. Automatica **45**(8), 1957–1959 (2009)
38. Lubich, C.H.: Discretized fractional calculus. SIAM J. Math. Anal. **17**(83), 704–719 (1986)
39. Luo, Y., Chen, Y.Q.: Fractional order proportional derivative controller for a class of fractional order systems. Automatica **45**(10), 2446–2450 (2009)
40. Luo, Y., Chen, Y.Q., Wang, C.Y., Pi, Y.G.: Tuning fractional order proportional integral controllers for fractional order systems. J. Process Control **20**(7), 823–831 (2010)
41. Lurie, B.J.: Three-parameter tunable tilt-integral-derivative (TID) controller. U.S. Patent US5371670, 6 December 1994
42. Maione, G.: Concerning continued fractions representations of noninteger order digital differentiators. IEEE Signal Process. Lett. **13**(12), 725 (2006)
43. Maione, G.: Approximation of the fractional operator s^{ν} using Thiele's continued fractions. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10), Badajoz, Spain (2010)
44. Maione, G., Lino, P.: New tuning rules for fractional PI$^{\alpha}$ controllers. Nonlinear Dyn. **49**, 251–257 (2007)
45. Manabe, S.: The non-integer integral and its application to control systems. Denki Gakkaishi **6**(3–4), 83–87 (1961)

46. Matignon, D.: Stability properties for generalized fractional differential systems. In: Proceedings of the Colloquium Fractional Differential Systems: Models, Methods and Applications, Paris, pp. 145–158 (1998)
47. Miller, K.S., Ross, B.: An Introduction to the Fractional Calculus and Fractional Differential Equations. Wiley, New York (1993)
48. Monje, C.A.: Design methods of fractional order controllers for industrial applications. PhD thesis, University of Extremadura (2006)
49. Monje, C.A., Chen, Y.Q., Vinagre, B.M., Xue, D., Feliu, V.: Fractional-order Systems and Controls. Fundamentals and Applications. Springer, Berlin (2010)
50. Monje, C.A., Vinagre, B.M., Chen, Y.Q., Feliu, V., Lanusse, P., Sabatier, J.: Proposals for fractional PID tuning. In: Proceedings of 1st IFAC Workshop on Fractional Differentiation and its Applications, Bordeaux, France (2004)
51. Monje, C.A., Vinagre, B.M., Feliu, V., Chen, Y.Q.: Tuning and auto-tuning of fractional order controllers for industry applications. Control Eng. Pract. **16**(7), 798–812 (2008)
52. Monje, C.A., Vinagre, B.M., Santamaría, G.E., Tejado, I.: Auto-tuning of fractional order PID controller using PLC. In: Proceedings of the 14th IEEE International Conference on Emerging Technologies and Factory Automation, Palma de Mallorca, Spain (2009)
53. Oldham, K.B., Spanier, J.: The Fractional Calculus. Theory and Applications of Differentiation and Integration of Arbitrary Order. Dover, New York (2006)
54. Oustaloup, A.: La Commade Crone: Commande Robuste D'ordre Non Entier. Hermès, Paris (1991)
55. Oustaloup, A.: La Dérivation Non Entière. Hermès, Paris (1995)
56. Oustaloup, A., Levron, F., Nanot, F., Mathieu, B.: Frequency-band complex noninteger differentiator: Characterization and synthesis. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. **47**, 25–40 (2000)
57. Oustaloup, A., Melchior, P., Lanusse, P., Cois, O., Dancla, F.: The CRONE toolbox for MATLAB. In: Proceedings of the IEEE International Symposium on Computer-Aided Control System Design (CACSD 2000), Anchorage, USA, pp. 190–195 (2000)
58. Petráš, I.: Stability of fractional-order systems with rational orders. Fract. Calc. Appl. Anal. **12**(3), 269–298 (2009)
59. Petráš, I., Hypiusova, M.: Design of fractional-order controllers via H_∞ norm minimisation. Sel. Top. Model. Control **3**, 50–54 (2002)
60. Pisoni, E., Visioli, A., Dormido, S.: An interactive tool for fractional order PID controllers. In: Proceedings of the 35th Annual Conference of IEEE Industrial Electronics (IECON'09), pp. 1468–1473 (2009)
61. Podlubny, I.: Fractional Differential Equations. An Introduction to Fractional Derivatives, Fractional Differential Equations, Some Methods of Their Solution and Some of Their Applications vol. 198. Academic Press, San Diego (1998)
62. Podlubny, I.: Fractional-order systems and $pi^\lambda d^\mu$-controllers. IEEE Trans. Autom. Control **44**(1), 208–214 (1999)
63. Podlubny, I., Petráš, I., Vinagre, B., O'Leary, P., Dorčak, L.: Analogue realizations of fractional-order controllers. Nonlinear Dyn. **29**, 281–296 (2002)
64. Podlubny, I., Vinagre, B.M., Chen, Y.Q., Feliu, V., Tejado, I. (eds.): Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10)
65. Romero, M., Tejado, I., Vinagre, B.M., de Madrid, A.P.: Fractional-order generalized predictive control of a servomotor plant. In: Proceedings of the 3rd IFAC Workshop on Fractional Differentiation and Its Applications (FDA'08) (2008)
66. Romero, M., Tejado, I., Vinagre, B.M., de Madrid, A.P.: Position and velocity control of a servo by using GPC of arbitrary real order. In: New Trends in Nanotechnology and Fractional Calculus Applications, pp. 369–376. Springer, Berlin (2009)
67. Romero, M., de Madrid, A.P., Mañoso, C.: Discrete approximations to fractional-order operators based on Chebyshev polynomials. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10), Badajoz, Spain (2010)

68. Santamaría, G.E., Valverde, J.V., Pérez-Aloe, R., Vinagre, B.M.: Microelectronic implementations of fractional-order integrodifferential operators. J. Comput. Nonlinear Dyn. **3**(2), 021301 (2008)
69. Sheng, H., Sun, H., Coopmans, C., Chen, Y.Q., Bohannan, G.W.: Physical experimental study of variable-order fractional integrator and differentiator. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10), 2010
70. Tan, K.K., Huang, S., Ferdous, R.: Robust self-tuning PID controller for nonlinear systems. J. Process Control **12**(7), 753–761 (2002)
71. Tejado, I., Vinagre, B.M., Chen, Y.Q.: Fractional gain and order scheduling controller for networked control systems with variable delay. Application to a smart wheel. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10) (2010)
72. Valério, D.: Ninteger (2008). http://www.mathworks.cn/matlabcentral/fileexchange/8312
73. Valério, D., da Costa, J.S.: Ninteger: a non-integer control toolbox for MATLAB. In: Proceedings of the 1st IFAC Workshop on Fractional Differentiation and its Applications IFAC Workshop on Fractional Differentiation and its Applications (FDA'06), Bordeaux, France (2004)
74. Valério, D., da Costa, J.S.: Tuning of fractional PID controllers with Ziegler-Nichols type rules. Signal Process. **86**(10), 2771–2784 (2006)
75. Valério, D., da Costa, J.S.: Tuning of fractional controllers minimising H_2 and H_∞ norms. Acta Polytech. Hung. **3**(4), 55–70 (2006)
76. Valério, D., da Costa, J.S.: Tuning rules for fractional PIDs. In: Fractional Calculus: Theoretical Developments and Applications in Physics and Engineering, pp. 463–476. Springer, Berlin (2007)
77. Valério, D., da Costa, J.S.: A review of tuning methods for fractional PIDs. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10) (2010)
78. Valério, D., da Costa, J.S.: Variable order fractional controllers. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10) (2010)
79. Vinagre, B.M.: Modelado y control de sistemas dinámicos caracterizados por ecuaciones íntegro-diferenciales de orden fraccional (in Spanish). PhD thesis, Spanish Open University (UNED) (2001)
80. Vinagre, B.M., Monje, C.A., Calderón, A.J., Chen, Y.Q., Feliu, V.: The fractional integrator as a reference function. In: Proceedings of the First IFAC Workshop on Fractional Differentiation and Its Applications, pp. 150–155. ENSEIRB, Bordeaux (2004)
81. Vinagre, B.M., Podlubny, I., Hernández, A., Feliu, V.: Some approximations of fractional order operators used in control theory and applications. Fract. Calc. Appl. Anal. **3**(3), 231–248 (2000)
82. Westerlund, S., Ekstam, L.: Capacitor theory. IEEE Trans. Dielectr. Electr. Insul. **1**(5), 826–839 (1994)
83. Zamani, M., Karimi-Chartemani, M., Sadati, N., Parmiani, M.: Design of a fractional order PID controller for an AVR using particle swarm optimization. Control Eng. Pract. **17**(12), 1380–1387 (2009)
84. Zhao, C., Xue, D., Chen, Y.Q.: A fractional order PID tuning algorithm for a class of fractional order plants. In: International Conference on Mechatronics and Automation, Niagara Falls (2005)
85. Ziegler, J.G., Nichols, N.B.: Optimum settings for automatic controllers. Trans. Am. Soc. Mech. Eng. **64**, 759–768 (1942)

Chapter 16
Event-Based PID Control

José Sánchez, Antonio Visioli, and Sebastián Dormido

16.1 Introduction

It is well known that in some processes, a small stationary control error or smooth oscillations of the process output around the set-point does not constitute hard design constraints; however, the reduction of the information exchanged between the agents that take part in the control loop (sensors, controllers, actuators) is one of the tightest requirements. With these demands, one of the most convenient strategies is the use of event-based sampling and control approaches. In recent years, event-based sampling and control techniques have drawn special attention from several research groups. Indeed, the reduction in information flow is a relevant issue, especially when there are constraints on the communication rate (for instance, when data are exchanged in a distributed control system by wired or wireless networks). In these situations, cutting down the traffic load is a key issue because the more traffic there is, the higher the possibility of losing data and experiencing stochastic time delays. This prevents the occurrence of large latencies and delay fluctuations, but the CPU utilization is also reduced. A well-known assertion in communication networks states that a reduction in information flow is always welcome, especially if the network is a generic one, such as the Internet, where the channel is shared by many applications. In any case, a framework in which a sheer reduction in the exchanged traffic is an essential issue is in wireless networks, particularly when using

J. Sánchez (✉) · S. Dormido
Departamento de Informática y Automática, UNED, C/ Juan del Rosal 16, 28040 Madrid, Spain
e-mail: jsanchez@dia.uned.es

S. Dormido
e-mail: sdormido@dia.uned.es

A. Visioli
Dipartimento di Ingegneria dell'Informazione, University of Brescia, Via Branze 38,
25213 Brescia, Italy
e-mail: antonio.visioli@ing.inibs.it

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_16, © Springer-Verlag London Limited 2012

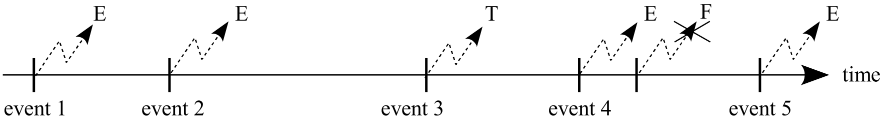


Fig. 16.1 *E* represents events triggered due to the asynchronous logical condition that becomes true at certain instants in time. *T* represents a forced event when $t_{\text{without}} \geq t_{\max}$. *F* represents a situation when the logical condition is true, but the time elapsed from the last event, the number 4, is less than $t_{\min}$

battery-powered or limited computational power devices. In these cases, the greater the reduction in information flow, the higher the decrease in computing operations and transmissions and thus the longer the lifetime of batteries becomes.

Before describing the application of the event-based paradigm to PID control, it is important to explain an event from a control-engineering point of view. In everyday life, an event is something that happens at a certain point in time. If this concept is implemented in a digital control system, it must be translated into a language that can be interpreted by computers. From a computer-based point of view, an event is an incident that occurs when some Boolean or logical condition becomes true. An example of a generic event-based condition is represented as

$$\text{(logical condition IS true) OR } (t_{\text{without}} \geq t_{\max}). \tag{16.1}$$

The *logical condition* detects whether something happens at a certain time, and thus it is considered the asynchronous term of the expression. In the process control field, the logical condition is composed of Boolean operations, where the variables are the signals that the sensor receives from the process or the control action produced by a controller. Otherwise, a function of the variables can be used (for instance, an estimate, the derivative, or an integration). The right-hand side of the event-based condition is a synchronous safety condition that forces the triggering of the event. This is commonly used because there are situations in which the logical condition can never be true and, therefore, no events can occur. For this time-based condition, an event is raised when the elapsed time without triggering the event, t_{without}, is higher than $t_{\max}$.

Also, there are situations in which it is necessary to include a second synchronous safety condition to guarantee a minimum inter-event time.

The new event-based condition can be described as

$$\big(\text{(logical condition IS true) AND } (t_{\text{without}} \geq t_{\min})\big) \text{ OR } (t_{\text{without}} \geq t_{\max}). \tag{16.2}$$

This new synchronous condition avoids the triggering of an infinite number of events within a finite interval $t_{\min}$. A graphical example is shown in Fig. 16.1. Because the triggering of an event in process control means, in general, activating a controller to calculate one or several control actions depending on the controller strategy, an excessive number of events would generate a high number of control actions that could damage the actuators or saturate the communication channels in the control loop, producing delays and data drop-outs.

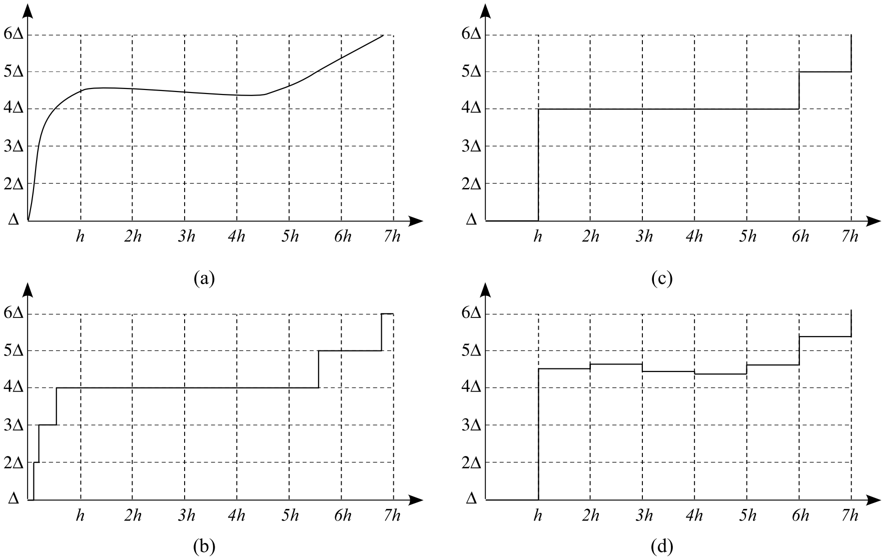


Fig. 16.2 (**a**) Continuous signal. (**b**) Event-based sampled signal obtained by the continuous evaluation of the logical condition. (**c**) Event-based sampled signal obtained with the "compound architecture". (**d**) Resulting time-based signal

Strictly speaking, to be able to detect the exact point in time when a logical condition becomes true, it should be continuously evaluated in real time and *ad infinitum*. However, a continuous evaluation of the condition implies a continuous sampling of the variables or functions involved in the logical expression; but current digital control systems implement discrete synchronous sampling strategies. For this reason, practical event-based control approaches are implemented to sample variables and evaluate event-based conditions as fast as possible. This is called the *compound approach* or the *fast sampling approach*, where the asynchronous events are presynchronized by fast periodic sampling [21]. However, to determine in a simulation the exact instant in time when an event occurs is not a difficult problem because when the event arises, it is possible to stop the simulation and, for example, to go backwards to make approximations.

Figure 16.2 presents the result of the evaluation of a simple event-based condition using a continuous signal obtained, for instance, from a sensor incorporated into an industrial process. In this example, the asynchronous logical condition to evaluate is

$$\left|y(t_{\text{act}}) - y(t_{\text{last}})\right| \geq \Delta \tag{16.3}$$

where $y(t_{\text{act}})$ is the current value of a process output $y(t)$, $y(t_{\text{last}})$ is the previous true-condition value of $y(t)$, and Δ is known as *threshold* or *delta*, which assumes a value smaller than the dynamic range of $y(t)$. This is expressed as $\Delta < |y_{\max} - y_{\min}|$. The resultant signals in Fig. 16.2 are obtained by applying the signal values that make the logical condition true to a zero order hold (ZOH). The

Table 16.1 Logical condition used in event-based sampling

Logical condition	True when
$\int_{t_{\text{last}}}^{t_{\text{act}}} \lvert y(t) - y(t_{\text{last}})\rvert \, dt \geq \Delta$	The integration of the difference between the current value of the signal $y(t_{\text{act}})$ and its value the last time that the condition was true $y(t_{\text{last}})$ is greater than Δ
$\lvert\hat{y}(t_{\text{act}}) - y(t_{\text{act}})\rvert \geq \Delta$	The difference between the prediction of the signal $\hat{y}(t_{\text{act}})$ and its current value $y(t_{\text{act}})$ is greater than Δ
$\int_{t_{\text{last}}}^{t_{\text{act}}} \lvert\hat{y}(t) - y(t)\rvert \, dt \geq \Delta$	The integral of the absolute value of the difference between the current value of the signal and its prediction from the last time that the condition was true t_{last} to the current time t_{act} is greater than Δ
$\int_{t_{\text{last}}}^{t_{\text{act}}} [y(t) - y(t_{\text{last}})]^2 \, dt \geq \Delta$	The energy of the difference between the current value of the signal and its value last time that the condition was true is greater than Δ
$\lvert\dot{y}(t_{\text{act}}) - \dot{y}(t_{\text{last}})\rvert \geq \Delta$	The difference between the derivative of the current signal $\dot{y}(t_{\text{act}})$ and its derivative the last time that the condition was true $\dot{y}(t_{\text{last}})$ is greater than Δ
$\lVert\hat{X}(t_{\text{act}}) - X(t_{\text{act}})\rVert \geq \Delta$	The difference between the model $\hat{X}(t_{\text{act}})$ and the real process $X(t_{\text{act}})$ is greater that Δ

signal in Fig. 16.2(b) is obtained when the logical condition is evaluated continuously. Figure 16.2(c) presents the result when a compound approach is applied with a sampling period of h. Figure 16.2(d) is the signal obtained with the classical time-triggered sampling paradigm. The process of generating a discrete-time signal by evaluating an event-based condition over a continuous-time signal is commonly known in control literature as *event-based sampling*. Other names found in control literature are variable-frequency sampling [8], adaptive sampling [24], deadband sampling [25], Lebesgue sampling [3], send-on-delta sampling [21], and level-crossing sampling [20].

In the example shown in Fig. 16.2, the number of samples obtained by the time-based approach is seven versus the three and six samples that its two event-based counterparts produce. For the logical condition of the example, a total absence of events occurs for a steady-state signal or when the oscillating range of the signal is below Δ. This situation can be avoided by adding a synchronous safety condition, though there are logical conditions based on integral criteria that can be used to avoid it. A summary of the event-based sampling strategies found in the bibliography is shown in Table 16.1.

In continuing to analyze the resulting signals in Fig. 16.2, it is clear that the key factor is the establishment of the parameter Δ to determine the sampling efficiency. There are studies from the 1970s related to the study of the merits of event-based sampling strategies [11, 15] and to the analytical design of the different event-based samplers [33]. Recent analytical studies on the send-on-delta sampling effectiveness against time-based sampling and other event-based criteria can be found in [2, 20, 22, 23]. Experimental comparisons based on models can be found in [21].

How does the event-based sampling approach work in a real framework, such as the automatic control of a process? In the majority of digital control systems, the actions of the main control agents (sensors, controllers, actuators) are ruled by a shared clock signal. This results in synchronized actions over a sampling interval h_{nom} that includes the sampling of the process by the sensor, the activation of the controller, and the application of the control action to the actuator. This is the basis for sampled control systems. However, if the event-based paradigm is applied, different events would independently cause the control agents to perform their corresponding actions in an asynchronous manner. Now, a different event-based condition could be associated with each agent to detect a change in a system variable or a function of the system variable. The simplest example of this situation is when a sensor detects a change in a signal (the event) and sends a sample of the output process to the controller (the action).

In the following section, a general classification of the event-based control strategies found in the literature is established to introduce the reader to the material covered in the rest of the chapter. The other sections are dedicated to explaining the evolution of event-based PID control, starting from the first published literature and progressing to the latest advances available at the moment this chapter was written (e.g., a two-degree-of-freedom (2DOF) event-based PI controller). It should be noted that to the authors' knowledge, there are no industrial implementations of event-based PID controllers, and there is only a small number of publications with results supported by real processes [17, 30, 31].

16.2 Classification of Event-Based Control Strategies

Research and publications regarding event-based sampling and control have become more widespread in recent years, especially those related to wireless networked control systems. Considering that any agent involved in a control loop (sensor, controllers, actuators) can work with events, some authors attempt to classify event-based control approaches by introducing an appropriate generic model for SISO systems [37, 38]. In these classifications, the agents can own input events to receive information (for example, the controller from the sensor or from the operator, the actuator from the controller) and output events to activate the transmission of information (a sample from the sensor to the controller or the control action to the actuator from the controller), and controllers can have different control algorithms.

However, to provide a more basic scheme of the differences between the event-based PID approaches and the other event approaches, a simplified structure of the system with event-based control presented in [2] is shown in Fig. 16.3 and described in the following paragraphs. The system consists of a generic process, an event detector, and a control input generator (the observer is included in the control generator). The event detector sends data (process output or state vector) to activate the control generator when an event occurs. The control input generator is inactive between two consecutive events and is activated by an invocation of the event detector (it is also possible to activate it by a time signal, but this would involve an observer

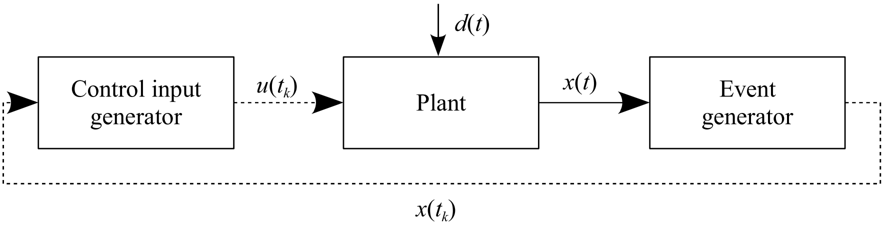


Fig. 16.3 Simplified block diagram of a control system based on an event-based approach. The transmission from the event detector is always event-based. The control input generator can be activated by time or when the event detector sends a sample

to estimate the plant state when no new information is available). Therefore, the control input generator works in a closed loop only at the events (feedback). Similarly, once the control input generator has produced the new control action, there are two possible conditions that can be used to send the data to the actuator: when they are available or when a logical condition located at its output is true, that is, an event-based solution.

Depending on the event-based condition located in the event detector and the control algorithm included in the control input generator, we can distinguish three generic categories. In the first, feedback control actions are computed when some variable varies more that a certain value Δ (see Table 16.1 for different event-based conditions). Thus, the control input generator produces a new control action every time that $x(t_k)$ up or down crosses the pre-calculated levels $(\ldots, -3\Delta, -2\Delta, -\Delta, 0, \Delta, 2\Delta, 3\Delta, \ldots)$. The control law employed in this category is, in most of the cases, a PID with variable sampling period [1, 7, 9, 10, 26, 28, 30–32, 35], but there are approaches based on state feedback [12, 13, 16, 19, 27, 38]. In the second group, the control action is set to the maximum/minimum while $x(t)$ stays out of a certain dead band around the set-point y_{sp} or a state trajectory X_s, that is, while $|x(t_{\text{act}}) - y_{sp}| \geq \Delta$ or $\|X(t_{\text{act}}) - X_s\| \geq \Delta$; at the very instant $x(t)$ re-enters into the band, a new event arises, and the control generator produces a feedforward control signal to move the process asymptotically to the reference value or to the trajectory [2, 5, 17, 18]. The feedforward control signal is obtained by means of a generalized hold that is reset every time x re-enters the band. Finally, it is possible to define a third group of event-based approaches, where one impulse action is generated to bring the process to the set-point value or to the origin when x crosses the limits of the dead band; the rest of the time the control action is zeroed [3, 6, 14, 29]. To summarize, the first group maintains a constant control law over the non-uniform sampling period delimited by two consecutive events, the second group is a variable control law, and the third group is an impulsive one.

Using the real-time system terminology, it is also possible to differentiate between *sporadic* and *aperiodic* event-based control. Aperiodic means that a minimum inter-arrival time between two consecutive events is not defined (that is, the second synchronous safety condition defined in Sect. 16.1 is not included in the event-based condition). The opposite situation is the sporadic control, where a minimum inter-event period to avoid saturations is defined. Examples of sporadic event-based approaches are reported in [14] and [29].

Bearing in mind the very different types of event-based control policies that have been presented previously, the central issue of this chapter is now addressed. The following sections are focused on the description of event-based controllers based on the ISA-PID control law [4] (Fig. 16.4) as the control input generator. Its representation in the frequency domain is

$$\begin{aligned} U(s) = K_p \biggl(& \beta Y_{sp}(s) - Y(s) + \frac{1}{T_i s}\bigl(Y_{sp}(s) - Y(s)\bigr) \\ & + \frac{T_d s}{1 + \frac{T_d}{N_d} s}\bigl(\gamma Y_{sp}(s) - Y(s)\bigr)\biggr). \end{aligned} \tag{16.4}$$

For the digital implementation of this controller, the control law must be discretized. This is typically achieved using forward differences for the integral term (such that it is possible to precalculate the integral part for time t_{k+1} at time t_k) and using backward differences for the derivative part. Thus, by denoting the sampling interval as `hnom` and by assuming $\gamma = 0$, the following pseudo-code of this PID controller results:

```
function u=pid(ysp,y)
  % Calculate control variable
  up=Kp*(beta*ysp-y);
  ud=Td/(Nd*hnom+Td)*ud-Kp*Td*Nd/(Nd*hnom+Td)*(y-y_old);
  u=up+ui+ud;

  % Update states
  ui=ui+Kp/Ti*hnom*(ysp-y);
  y_old=y;
```

Note that the previous code is not optimized; for example, some coefficients of the control law can be pre-calculated to reduce the computation time. Also, the scope of the variables is considered global.

All of the event-based PID approaches analyzed in this chapter consider that communications between the sensor, the event detector, the control input generator, and the actuator are instantaneous; that is, the communication delays and the computation time are negligible. In a theoretical event-based approach, the event detector must continuously examine the sensor and evaluate the event-based condition to invoke the control input generator when it becomes true. As noted before, in practical digital implementations, the event detector is time-triggered with a fixed sampling period. Depending on this period, the event detector can perform a discrete or a quasi-continuous evaluation of the event-based condition (i.e., the compound approach). However, a higher sampling frequency induces higher computational cost.

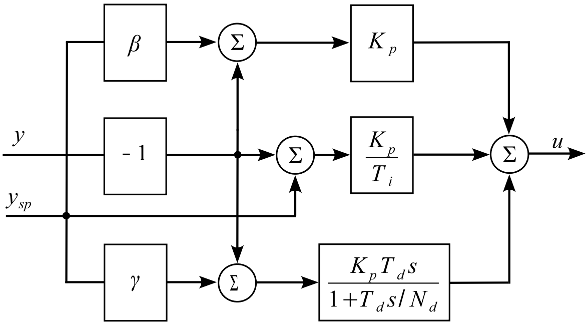


Fig. 16.4 Block diagram and simplified representation of a PID controller based on the ISA law

16.3 The First Event-Based PID Controller

The first event-based PID controller reported in the literature is described in [1]. The aim of this controller is to reduce the CPU time for control law computation, not to provide an overall reduction of the information exchanged in the control loop. Actually, as it occurs in practical implementations of event-based approaches, this controller is a hybrid approximation between a time and an event-based controller. The activation of the event generator is time-based with period h_{nom}, but the invocation of the control input generator is not. If the event-based condition is fulfilled, the control input generator is invoked to produce a new control action. Note that if the activation frequency of the event generator is increased to the maximum in a real implementation, this controller becomes a quasi-pure, event-based approach. According to the author, this event-based PID control structure can be distributed in a client–server architecture, where clients running periodical event detectors send samples to a server where a control generator produces the control action (according to the parameters and state information associated with each client).

As described before, at every nominal sampling period h_{nom} the controller reads the sample from the sensor, and the event detector evaluates the event-based condition with this information. If the condition is true, an event occurs, and the control input generator will calculate the control action using a PID law with variable sampling period. The logical condition is a function of the relative error. Thus, the condition is true when the absolute value of the difference between the current value of the control error $e(t_{\mathrm{act}})$ and the value of the control error $e(t_{\mathrm{last}})$ when the last event occurred is greater than a predefined threshold Δ_e. In other words, an error-based send-on-delta sampling is applied to the controller. Furthermore, the event detector includes a time-based safety condition to guarantee a maximum time-interval between consecutive events h_{max}. In this way, a new value of the control signal is forced to be calculated when the elapsed time $h_{\mathrm{act}} = t_{\mathrm{act}} - t_{\mathrm{last}}$ exceeds a given limit h_{max}. Formally, the overall event-based condition can be written as

$$\big(\left|e(t_{\mathrm{act}}) - e(t_{\mathrm{last}})\right| \geq \Delta_e\big) \text{ OR } (h_{\mathrm{act}} \geq h_{\mathrm{max}}). \tag{16.5}$$

In [12], the sampling pattern that produces the event-based condition is denoted as *locally non-uniform sampling*. The event-based condition uses a uniform sampling time h_{max} when the error does not change or oscillates smoothly but uses a

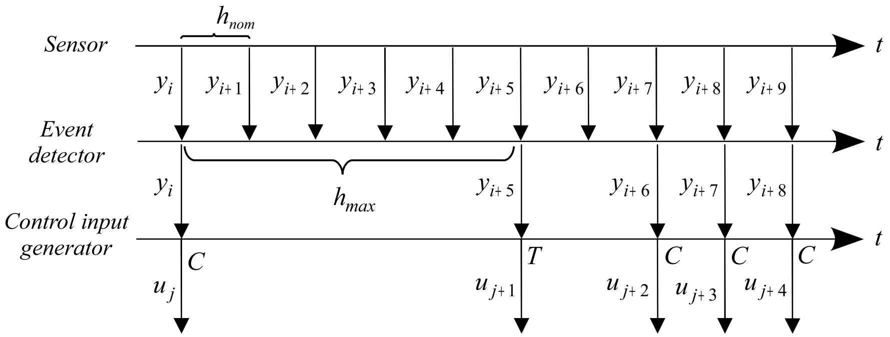


Fig. 16.5 Exchange of information between sensor, event detector, and control input generator in the event-based PID approach where $h_{\max} = 5h_{\text{nom}}$. C represents that control actions are calculated because the logical condition becomes true, and T represents that the control input generator acts due to the safety condition ($h_{\text{act}} \geq h_{\max}$)

non-uniform sampling period h_{act} if the current error variation maintains locally below Δ_e before the timer fixed to $h_{\max}$ triggers. It should be noted that the non-uniform sampling period h_{act} is always a multiple of h_{nom}. Figure 16.5 presents an example of the information exchanged between the control agents using the event-based condition defined in (16.5).

The pseudo-code for this PID controller would read as follows:

```
function u=event_detector(ysp,y)
  e=ysp-y;
  hact=hact+hnom;

  if (abs(e-es)>Delta_e || hact>=hmax)
    u=control_input_generator(ysp,y);
  end;


function u=control_input_generator(ysp,y)
  up=Kp*(beta*ysp-y);
  ud=Td/(Nd*hact+Td)*ud-Kp*Td*Nd/(Nd*hact+Td)*(y-y_old);
  u=up+ui+ud;

  % Update integral term
  ui=ui+Kp/Ti*hact*(ysp-y);

  y_old=y;
  hact=0;
  es=e;
```

The control input generator is executed only if the event is detected, thus saving computational resources if there is no need to update the control variable (for example, when the process is at the required steady-state). However, the computational time increases when an event occurs because the controller coefficients cannot be precalculated, as they depend on the current sampling interval h_{act}, namely, the time elapsed between the last and the current sampling time. Furthermore, two additional state variables (`es` and `hact`) and two additional parameters (`Delta_e` and `hmax`) are needed with respect to the standard time-based digital algorithm presented in (16.4).

16.4 Alternatives to Calculate the Integral Term

In the digital implementation of a PID controller, the control law is typically discretized using the forward Euler approximation for the integral [4, 40]. If t_k is the current sampling time, the integral control action to be applied in the next sampling time t_{k+1} is pre-calculated by the following expression

$$u_i(t_{k+1}) = u_i(t_k) + \frac{K_p}{T_i} h_{\mathrm{nom}} e(t_k). \tag{16.6}$$

As the nominal sampling period h_{nom} is constant, there is no problem in precalculating the contribution of the integral part for the next sampling time t_{k+1}. However, as pointed out in [10], one of the main drawbacks of the previous algorithm is that the integral action is pre-calculated without actually knowing the true length of the next sampling period because it is unknown a priori. In a discretized PID controller, the pseudocode to calculate $u_i(t_{k+1})$ is

```
ui=ui+Kp/Ti*hnom*(ysp-y);
```

where `hnom` is the nominal sampling period. In the event-based approach, this value is replaced by the time elapsed between the last two events, that is, h_{act}. In other words, the event-based algorithm substitutes h_{nom} with h_{act}, which might not be sensible.

A way to overcome this problem, according to [10], is to calculate the integral part by shifting the time instants of the integral time in the original expression. Such shifting produces the expression

$$u_i(t_k) = u_i(t_{k-1}) + \frac{K_p}{T_i}(t_k - t_{k-1}) e(t_{k-1}). \tag{16.7}$$

Then, taking into account that in the event-based approach $t_{k-1} = t_{\mathrm{last}}$, $t_k = t_{\mathrm{act}}$ and $h_{\mathrm{act}} = t_{\mathrm{act}} - t_{\mathrm{last}}$, the previous expression can be reformulated as

$$u_i(t_{\mathrm{act}}) = u_i(t_{\mathrm{last}}) + \frac{K_p}{T_i} h_{\mathrm{act}} e(t_{\mathrm{last}}). \tag{16.8}$$

In this way, the integral control action is not pre-calculated for the next activation of the controller but for the current one based on the available information. Another solution proposed in [10] and very similar to the previous approach is achieved by calculating the integral part using the backward approximation instead of the forward one. Thus, using backward differences, the integral part is expressed as

$$u_i(t_{\text{act}}) = u_i(t_{\text{last}}) + \frac{K_p}{T_i} h_{\text{act}} e(t_{\text{act}}). \tag{16.9}$$

The pseudo-code for the event-based PID controller based on backward approximation could be written as follows:

```
function u=event_detector(ysp,y)
  e=ysp-y;
  hact=hact+hnom;

  if (abs(e-es)>Delta_e || hact>=hmax)
    u=control_input_generator(ysp,y)
  end;


function u=control_input_generator(ysp,y)
  up=Kp*(beta*ysp-y);

  % Backward approximation of the integral term
  ui=ui+Kp/Ti*hact*(ysp-y);
  ud=Td/(Nd*hact+Td)*ud-Kp*Td*Nd/(Nd*hact+Td)*(y-y_old);
  u=up+ui+ud;

  y_old=y;
  hact=0;
  es=e;
```

16.5 Removing the Safety Synchronous Condition

In any case, it is obvious that removing the safety limit condition $h_{\text{act}} \geq h_{\max}$ would greatly simplify the algorithm, as there would be one parameter less to tune. However, despite removing the condition, there remains the problem that large overshoots can be generated because of excessive integral action if the period h_{act} between two successive events becomes large and a set-point value change is required after a long steady-state period or a disturbance appears. This matter has been discussed in [10], where three different solutions are introduced: saturation of the integral term, exponential forgetting factor of h_{act}, and a hybrid algorithm based on the two previous solutions. In the three solutions, the backward approximation of the integral part is considered.

The first consists of expressing the integral control action as

$$u_i(t_{\text{act}}) = u_i(t_{\text{last}}) + \frac{K_p}{T_i}\tilde{E} \tag{16.10}$$

where the product $\tilde{E}$ is obtained by

$$\tilde{E} = (h_{\text{act}} - h_{\text{nom}})\Delta_e + h_{\text{nom}} e(t_k) \tag{16.11}$$

when h_{act} is greater than $h_{\max}$. Actually, the safety condition in this approach has not been removed, but the error area is approximated better. The time interval between two consecutive events can be divided into two sections: the real steady-state interval and the detection interval. The first is equal to $(h_{\text{act}} - h_{\text{nom}})$ and the second is h_{nom}, which is the interval over which the error becomes greater than Δ_e due to a set-point change, and the event detector is activated. In the first interval, the error area fulfills the following expression

$$\int_{t_{\text{last}}}^{t_{\text{act}}-h_{\text{nom}}} e(t)\,dt \le (h_{\text{act}} - h_{\text{nom}})\Delta_e. \tag{16.12}$$

The previous expression is always true because otherwise $e(t) > \Delta_e$, which is a condition under which the detector would have been activated. The error area during the detection interval is bounded by

$$\int_{t_{\text{act}}-h_{\text{nom}}}^{t_{\text{act}}} e(t)\,dt \le h_{\text{nom}} e(t_{\text{act}}) \tag{16.13}$$

where $e(t_{\text{act}})$ is the error value that forces the activation of the event detector. A better approximation of the total error area between two consecutive events $\tilde{E}$ can be obtained by adding the two previous upper limits (16.12) and (16.13)

$$\tilde{E} = (h_{\text{act}} - h_{\text{nom}})\Delta_e + h_{\text{nom}} e(t_{\text{act}}) \le h_{\text{act}} e(t_{\text{act}}). \tag{16.14}$$

Applying this modification, the following pseudocode can be written:

```
function u=event_detector(ysp,y)
  e=ysp-y;
  hact=hact+hnom;

  if (abs(e-es)>Delta_e)
    u=control_input_generator(ysp,y)
  end;


function u=control_input_generator(ysp,y)
  up=Kp*(beta*ysp-y);
  % Calculate the new approximation of the error area
  if (hact>=hmax)
    E=(hact-hnom)*Delta_e+hnom*e;
  else
    E=hact*e;
  end;

  ui=ui+Kp/Ti*E;
  ud=Td/(Nd*hact+Td)*ud-Kp*Td*Nd/(Nd*hact+Td)*(y-y_old);
  u=up+ui+ud;

  y_old=y;
  hact=0;
  es=e;
```

The second technique involves replacing the true sampling interval h_{act} in the expression of the integral part by a small one h_{exp} obtained by an exponential function with a forgetting factor based on h_{act} and h_{nom}. In this way, the original value of h_{act} is reduced after a long steady-state time interval and, thus, so is the integral contribution. Formally, the current sampling interval h_{exp} to be employed in determining the integral action is defined as

$$h_{\mathrm{exp}} = h_{\mathrm{act}} e^{(h_{\mathrm{nom}} - h_{\mathrm{act}})}. \tag{16.15}$$

The pseudo-code corresponding to the second modification is:

```
function u=event_detector(ysp,y)
  e=ysp-y;
  hact=hact+hnom;

  if (abs(e-es)>Delta_e)
    u=control_input_generator(ysp,y)
  end;


function u=control_input_generator(ysp,y)
  up=Kp*(beta*ysp-y);

  % Apply exponential decreasing
  hexp=hact*exp(hnom-hact);
  ui=ui+Kp/Ti*hexp*e;

  ud=Td/(Nd*hact+Td)*ud-Kp*Td*Nd/(Nd*hact+Td)*(y-y_old);
  u=up+ui+ud;

  y_old=y;
  hact=0;
  es=e;
```

The best solution is, however, a hybrid one based on the previous two solutions because the exponential algorithm is not capable of reducing the overshoot appropriately if the steady-state time interval is not sufficiently long. For this reason, the hybrid algorithm uses the approximation of the integral error but replaces h_{act} by $h_{\exp}$. Therefore, when h_{act} is greater than $h_{\max}$, the error area is obtained by

$$\tilde{E} = (h_{\exp} - h_{\mathrm{nom}})\Delta_e + h_{\mathrm{nom}} e(t_{\mathrm{act}}). \qquad (16.16)$$

Thus, the following algorithm is produced:

```
function u=event_detector(ysp,y)
  e=ysp-y;
  hact=hact+hnom;

  if (abs(e-es)>Delta_e)
    u=control_input_generator(ysp,y)
  end;
```

```
function u=control_input_generator(ysp,y)
  up=Kp*(beta*ysp-y);

  if (hact>=hmax)
    hexp=hact*exp(hnom-hact);
    E=(hexp-hnom)*Delta_e+hnom*e;
  else
    E=hact*e;
  end;

  ui=ui+Kp/Ti*E;

  ud=Td/(Nd*hact+Td)*ud-Kp*Td*Nd/(Nd*hact+Td)*(y-y_old);
  u=up+ui+ud;

  y_old=y;
  hact=0;
  es=e;
```

16.6 Minimizing the Computational Requirements

The previous PID algorithm based on the hybrid approach produces increased computational effort, particularly because of the calculation of the exponential function (indeed, the algorithm with only the saturation of the product $\tilde{E}$ is much simpler). The use of a look-up table with the pre-calculated values of the function can be an effective means of reducing the computational cost. However, some improvements to this algorithm are described in [9] to reduce the computational cost and to achieve a performance similar to a conventional time-triggered PID controller.

The minimization of the computational cost implies minimizing the number of triggered events. The solution proposed in [9] is based on calculating the control signal only when the measurement is too far from the set-point value. In the original conception, the event-based condition is based on the relative error and uses a time-triggered event detector with h_{nom}. Now, the logical condition of the detector is based on the absolute error

$$\left|e(t_{\text{act}})\right| > \Delta_e. \tag{16.17}$$

To pinpoint when the event happens, the authors propose replacing the time-triggered detector by a continuous one to send a request at the very moment that the error exceeds the threshold. In real implementations, the continuous time event detector is employed by reducing the value of h_{nom} (the compound approach) as much as possible to simulate a continuous sampling.

With this new event-based condition the number of events is minimized during the steady-state intervals when the error remains below the threshold; however, during the transient period, the number of events will grow. To reduce the number of events, a new logical condition is introduced. It consists of adding a minimum inter-event time, that is, $h_{\text{act}} > h_{\text{min}}$, to the event detector. According to [9], this minimum time interval could be chosen as the discrete sampling period corresponding to a time-triggered controller. Another option is to find a compromise between the reduction of h_{nom} and the number of events produced.

Another improvement in this low-computation version of the event-based algorithm is the reduction of the absolute error during the steady-state intervals. With the current implementation, $e(t)$ remains below Δ_e, not producing control actions to reduce the error. To force a change in this situation, an extra event is sent to the control generator if nothing happens during the time h_{extra} elapsed from the last time an event was triggered. This extra event will be repeatedly forced while the error is greater than a desired minimum level e_{min}.

This low computational cost event-based PID controller can be implemented as follows:

```
function u=event_detector(ysp,y)
  e=ysp-y;
  hact=hact+hnom;
  hlast=hlast+hnom;
  if (abs(e)>Delta_e && hact>hmin)
    u=control_input_generator(ysp,y);
  elsif (hlast>hextra && abs(e)>emin)
    u=control_input_generator(ysp,y);
    hlast=0;
  end;


function u=control_input_generator(ysp,y)
  up=Kp*(beta*ysp-y);

  if (hact>=hmax)
    hexp=hact*exp(hnom-hact);
    E=(hexp-hnom)*Delta_e+hnom*e;
  else
    E=hact*e;
  end;

  ui=ui+Kp/Ti*E;
  ud=Td/(Nd*hact+Td)*ud-Kp*Td*Nd/(Nd*hact+Td)*(y-y_old);
  u=up+ui+ud;

  hact=0;
  y_old=y;
```

In this pseudo-code, it can be appreciated that the event detector is evaluated every h_{nom}, and it is a combination of logical conditions with two different purposes. The control input generator is invoked with different periodicity depending on the process dynamics (steady state or transients) and the time elapsed from the last event.

Simulations in [9] show that the behavior of this controller is better than that of the others described in this section, with a performance very close to a time-triggered PID counterpart using the same set of parameters. However, real experiments are not provided, and simulations are run without noise. An experimental study of different event-based strategies to control the liquid level in a single tank is provided in [30]. In that work, many algorithms are tested, but the overall conclusion is that the event-based PID approach described in [1] works sufficiently well.

16.7 The Event-Based PID Controller with Sticking Detection

The algorithm proposed in [34] is similar to that proposed in [1], though some differences should be pointed out (in addition to the use of backwards differences for the integral part). First, the event-based condition considered is

$$\left(\left(\left|e(t_{\mathrm{act}})-e(t_{\mathrm{last}})\right|>\Delta_e\right)\text{ AND }(h_{\mathrm{act}}\geq h_{\mathrm{min}})\right)\text{ OR }(h_{\mathrm{act}}\geq h_{\mathrm{max}}). \qquad (16.18)$$

Another difference is that the tuning rule given for the value of $h_{\max}$ is selected equal to one or two times h_{settling}, namely, the settling time of the loop for the step response. The problem of the impact of the integral term due to a long period without events is solved by limiting the integration time $h_{\max I}$ to h_{rise}, namely, the process rise time for step response.

The approaches described in the previous section do not address the sticking problem [35] that can occur depending on the logical condition chosen in the send-on-delta sampling law. This phenomenon appears when the error signal changes slowly with its derivative tending to zero (for example, due to sluggish control actions or at the peaks of overshoots) such that the control loop achieves an equilibrium state (different from that required by the control specifications) with $|e(t_{\text{act}})| > \Delta_e$. As the condition $|e(t_{\text{act}}) - e(t_{\text{last}})| \leq \Delta_e$ remains true, the event detector does not act, and new control actions are not produced to remove the system from this state. The system will remain in this state until the safety limit condition becomes true (if it exists) or a disturbance or a new set-point value forces a change of the derivative.

In this context, the use of the time limit condition $h_{\text{act}} \geq h_{\max}$ would not help solve the problem if $h_{\max}$ assumes an excessively high value. The method proposed in [34] to resolve this problem is to generate an event depending on the sensor value derivative (or, equivalently, depending on the derivative of the control error). Thus, the controller will produce a new control action after a period $h_{\max S}$ if the error derivative is quite small, and its last variation tends to zero. This is expressed in [34] as

$$(h_{\text{act}} \geq h_{\max S}) \text{ AND } \left(\left|e(t_{\text{act}})\right| \geq \Delta_s\right) \text{ AND } \left(\left|\dot{e}(t_{\text{act}}) + \left(\dot{e}(t_{\text{act}}) - \dot{e}(t_{\text{last}})\right)\right| < \frac{\Delta_s}{2}\right), \tag{16.19}$$

where the recommendation is to fix $h_{\max S} = 2h_{\text{nom}}$. The following sketch of the algorithm is proposed:

```
function u=event_detector(ysp,y)

  e=ysp-y;
  hact=hact+hnom;
  error_der=(e-e_old)/hact;
  diff_error_der=error_der-error_der_old;

  if (hact>hmaxS &&
         abs(e)>Delta_s &&
             abs(error_der+diff_error_der)<Delta_s/2)
    u=control_input_generator(ysp,y);
  elsif(hact>=hmax || (abs(e-es)>Delta_e && hact>=hmin))
    u=control_input_generator(ysp,y);
  end;

  % Update states
  e_old=e;
  error_der_old=error_der;

function u=control_input_generator(ysp,y)
  up=Kp*(beta*ysp-y);

  % Limit the integration time
  if (hact>=hmaxI)
    hi=hmaxI;
  else
    hi=hact;
  end;

  ui=ui+Kp/Ti*hi*(ysp-y);
  ud=Td/(Nd*hact+Td)*ud-Kp*Td*Nd/(Nd*hact+Td)*(y-yold);

  u=up+ui+ud;

  % Update states
  yold=y;
  hact=0;
```

It is evident that the complexity of the event detector increases and more parameters must be tuned. In the previous description of the algorithm, it was considered that the event detector is periodically invoked every h_{nom}, and this value is sufficiently small to resemble a quasi-continuous sampling of the process. In this situation, the suggestion given in [34] is to select $\Delta_e = \Delta_s$ and $\Delta_s = (0.05\text{–}0.15)\Delta_{\min \text{SP}}$, where $\Delta_{\min \text{SP}}$ is the minimum set-point change, but fulfills $\Delta_s \le e_{\max \text{Steady}}/2$, where $e_{\max \text{Steady}}$ is the maximum tolerated steady-state error. In an actual event-based conception of the algorithm, the event detector and the control input generator should be independent entities working separately. In this distributed architecture, the event detector should check the logical condition continuously and send a signal to invoke the control input generator when necessary. If the previous algorithm is implemented in this way, the suggestion is set to $\Delta_e = 0$.

It should be noted that some of the logical conditions presented in Table 16.1 avoid the sticking because they use the integration of the signal and not its instantaneous value. Though the process output maintains the steady state, the integration of the error will finally produce the triggering of the event. Also, if the logical condition checks the absolute error, that is, $|e(t_{\text{act}})| \ge \Delta_e$, the sticking is also avoided.

16.8 Avoidance of Limit Cycles

Event-based controllers suffer from limit cycles because these controllers are a kind of quantized system with limited accuracy. During the steady state, limit cycles take the form of small oscillations around the set-point value, and their amplitude and frequency vary depending on the tuning parameters. A study of the emergence condition of limit cycles in a PI control loop with send-on-delta sampling is presented in [39].

A practical solution to reduce the probability and frequency of limit cycles in the event-based PID approach presented in [34] (see Sect. 16.7) is introduced in [35]. To detect the existence of the limit cycle, the control error is filtered with a low-pass filter and compared to the unfiltered control error. This condition is augmented with the dead zone on control error. Thus, the complete limit cycle avoidance condition can be expressed as

$$\left(\left|e_f(t_{\text{act}}) - e(t_{\text{act}})\right| < \Delta_s\right) \text{ AND } \left(\left|e(t_{\text{act}})\right| < \Delta_s\right) \tag{16.20}$$

where e_f is the filtered control error. When this condition becomes true, the error is set to zero. The error is filtered using the same mechanism applied to the derivative term.

Another contribution to this algorithm is a mechanism to approximate the integral term. This time the solution is to apply a high-pass filter to the integral term based on the backward approximation. Therefore, the integral term is expressed as

$$u_i(t_{\text{act}}) = u_i(t_{\text{last}}) + \frac{K_p}{T_i}\frac{h_{\text{act}}}{1+\frac{h_{\text{act}}}{N_i}}e(t_{\text{last}}). \tag{16.21}$$

The pseudo-code for the event-based control algorithm with the reduction of limit cycles and the filtering of the integral term is:

```
function u=event_detector(ysp,y)
  e=ysp-y;
  hact=hact+hnom;
  error_der=(e-eold)/hact;
  diff_error_der= error_der-error_der_old;

  if (hact>hmaxS &&
             abs(e)>Delta_s &&
               abs(error_der+diff_error_der)<Delta_s/2)
    u=control_input_generator(ysp,y);
  elsif (hact>=hmax || (abs(e-es)>Delta_e && hact>=hmin))
    u=control_input_generator(ysp,y);
  end;

  % Update states
  e_old=e;
  error_der_old=error_der;

function u=control_input_generator(ysp,y)
  up=Kp*(beta*ysp-y);
```

```
% Filter the error
ad=1/(1+Ne*hact);
ef=ad*ef+ad*Ne*(ysp-y);
if (abs(ef-e)<Delta_s && abs(e)<Delta_s)
  e=0;
end;

% Filter the integral
hf=hact/(1+hact/Ni);
ui=ui+Kp/Ti*hf*e_old;

% Filter the derivative
ud=Td/(Nd*hact+Td)*ud-Kp*Td*Nd/(Nd*hact+Td)*(y-y_old);

u=up+ui+ud;

% Update states
yold=y; hact=0;
```

Based on simulations, some suggestions are given in [35] to tune the filter coefficients of the integral term and the error. The recommended values are $N_e \in \{3, \dots, 10\}$ and $N_i = \frac{T_i}{h_{\text{nom}}}$.

16.9 Use of Simplified Predictors in Event-Based PID Control

In the previous approaches, the event detector was considered time-triggered (with a fast sampling period) and the control input generator event-based. Moreover, in the last sections, the creation of timing problems from the event-based nature of the control input generator was discussed; these problems include variable sampling periods, large approximation errors in the integral and derivative terms, and sticking. Such drawbacks reduce the quality of the control, and heuristics must be introduced in the control law to resolve them.

A possible configuration that eliminates the timing problems is that both elements are time triggered but both send the information to their respective adjacent neighbor when some event-based condition is fulfilled. Thus, the event-detector evaluates every h_s its event-based condition (for instance, $|y(t_{\text{act}}) - y(t_{\text{last}})| \geq \Delta_s$), and the control input generator produces every h_c a control action that is sent to the actuator.

As there are no communication delays and the computational time is negligible or inferior to the nominal sampling periods h_s and h_c, there are no timing problems in this approach. However, the problem with this configuration is that the control input generator could not receive the process state from the sensor every h_c, and this information must be obtained by employing an (Kalman, Luenberger) observer in the controller to predict the value of the process variable at each sampling interval

h_c. In this approach, the observer actually runs in an open loop between samples, and the process variable is estimated using the last received state. The state estimation is updated every time a new signal arrives from the sensor. It is evident that the effectiveness of the methodology heavily relies on the accuracy of the observer (which means that the design effort increases significantly) and that the presence of disturbances can imply a significant decrement in performance. A partial solution to these problems can be achieved by employing a reduced-order (first-order) observer, as in [36]. The idea is that the implementation of a time-based PID algorithm with a first order observer is very simple once a first-order dynamic model of the real process is obtained.

A similar approach can be implemented by applying a first-order state-space predictor of the plant to estimate the plant output in an open loop in the absence of updated information from the event detector. The simplified first-order predictor can be written as

$$\begin{aligned} \hat{x}(t_k) &= A_s\hat{x}(t_{k-1}) + b_s u(t_{\text{last}}), \\ \hat{y}(t_k) &= c_s\hat{x}(t_k) \end{aligned} \tag{16.22}$$

where A_s, b_s, and c_s are first order matrices and $u(t_{\text{last}})$ is the last control action sent to the actuator. In this approach, the predictor is part of the control input generator, and thus, it is evaluated every h_c with the available information. At the beginning of each evaluation, the arrival of a new value of the process output sent from the event detector is checked. If this occurs, the PID law calculates the control action. If there are no arrivals, the controller uses the last prediction of the process output. After that, the predictor is evaluated with the new information, and the process output prediction is ready for the next invocation of the control generator.

The use of the predictor demands a small h_c to reduce the estimation errors, though this produces a high number of control actions that can saturate the communication link or produce wearing in the actuators. To maintain low communication traffic, this approach introduces an event-based condition in the control input generator to evaluate whether it is worth transmitting the new control action to the actuator. The logical condition is

$$\left|u(t_{\text{act}}) - u(t_{\text{last}})\right| \geq \Delta_c. \tag{16.23}$$

Considering the fact that the event detector and the control input generator are periodically evaluated each h_s and h_c, respectively, a pseudo-code of this approach could be written as follows:

```
% Execute with period hs

function sensor_event_detector(y,ysp)
  if (abs(y-y_last)>=Delta_s)
    send_to_control_input_generator(y,ysp);
    y_last=y;
  end;
```

```
% Execute with period hc

function control_input_generator(y,ysp)
  if (no_new_y_from_event_detector)
    y=y_est;
  end;

  up=Kp*(beta*ysp-y);
  ud=Td/(Nd*hc+Td)*ud-Kp*Td*Nd/(Nd*hc+Td)*(y-y_old);
  u=up+ui+ud;

  controller_event_detector(u);

  ui=ui+Kp/Ti*hc*(ysp-y);
  y_old=y;

  % Predict the next process output
  x_est=y/cs;
  x_est=As*x_est+bs*u;
  y_est=cs*x_est;


function controller_event_detector(u)
  if (abs(u-u_last)>=Delta_c)
    send_to_actuator(u);
    u_last=u;
  end;
```

The algorithm does not include any safety condition to force the triggering of events or heuristics to detect sticking or to reduce the error during the stationary state. Some of the modifications introduced in previous sections could be included.

16.10 Separating the Event-Detectors for the P, I, and D Parts

All of the previous control approaches consider one event detector to activate the calculation of the overall control action. A strategy that employs two event detectors to activate the calculation of the control action in a PI controller (no derivative action is used) has been proposed in [28]. In this approach, there is an event detector based on the error signal and another event detector that analyzes the integral of the error. Therefore, there are P- and I-type events to activate the control input generator. When one of these two events occurs, the parameters to calculate the update of the control signal are sent to the control input generator.

The proportional action is produced in this approach in two situations: if the unsigned error signal has grown by an amount Δ_P with respect to the last time an event was triggered or if the error signal diminishes in magnitude since the last event detection, as desired, but has shot over the origin by at least Δ_P. The full P-type event-based condition that combines these two situations is expressed in [28] as

$$\big(\big|e(t_{\text{act}})\big| - \big|e(t_{\text{last}})\big| \geq \Delta_P\big) \text{ OR } \big(e(t_{\text{act}}) \cdot e(t_{\text{last}}) < 0 \text{ AND } \big|e(t_{\text{act}})\big| \geq \Delta_P\big). \quad (16.24)$$

The motivation to introduce two types of events is that P-type events on their own can result in steady-state errors and sticking phenomena. As noted before, logical conditions based on the integration of a signal can help to eliminate these problems. Regarding the integral part, an I-type event is triggered when the integral of the error changes by more than an amount Δ_I with respect to the last event detection, that is,

$$\left|\int_{t_{\text{last}}}^{t_{\text{act}}} e(t)\,dt\right| \geq \Delta_I. \quad (16.25)$$

It should be noted that t_{last} is updated to t_{act} when a new event takes place, regardless of the type. Every time a P- or I-type event arises, the control input generator produces an update of the proportional and integral terms and $t_{\text{last}} = t_{\text{act}}$. Between two subsequent events, the control signal is defined as a superposition of two simple waveforms, that is, a series of pulses whose amplitudes depend on the error signal and a piecewise constant waveform whose values depend on samples of the integral error signal. Formally, it is expressed as

$$u(t_{\text{act}}) = \begin{cases} -K_p e(t_{\text{last}}) - K_i \int_{t_{\text{last}}}^{t_{\text{act}}} e(t)\,dt & \text{for } t_{\text{act}} \leq t_{\text{last}} + \varepsilon(K_p, K_i), \\ -K_i \int_{t_{\text{last}}}^{t_{\text{act}}} e(t)\,dt & \text{for } t_{\text{act}} > t_{\text{last}} + \varepsilon(K_p, K_i) \end{cases} \quad (16.26)$$

where $K_i = K_p/T_i$ and $\varepsilon(K_p, K_i)$ is the pulse duration that determines the application of the proportional part, while the integral part is applied for the whole length of the sampling interval. It appears that in this case there are also many parameters to tune in the overall algorithm.

Considering the fact that the waveform generator code is invoked each h_{nom}, a pseudo-code for the previous approach could be written as follows:

```
function u=PI_waveform_generator(ysp,y)
  hact=hact+hnom;
  [up,ui]=event_detectors(ysp,y);

  if (hact>length_P_pulse)
    u=ui;
  else
    u=up+ui;
  end;
```

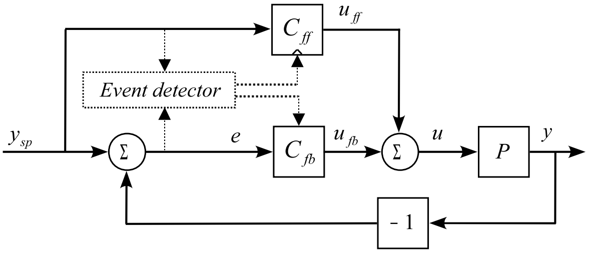


Fig. 16.6 Block diagram of the 2DOF event-based controller

```
function (up,ui)=event_detectors(ysp,y)
  e=ysp-y;
  IE=IE+hnom*e;

  % Detecting I-type events
  if (abs(IE)>=Delta_I)
    [up,ui]=control_input_generator(ysp,y)
  end;

  % Detecting P-type events
  if (abs(e)-abs(e_old))>=Delta_P ||
              (abs(e)>=Delta_P && (e*e_old<0))
    [up,ui]=control_input_generator(ysp,y)
  end;

function (up,ui)=control_input_generator(ysp,y)
  up=Kp*(beta*ysp-y);
  ui=ui+Kp/Ti*IE;
  e_old=e;
  y_old=y;
  hact=0;
  IE=0;
```

16.11 The 2DOF Event-Based PI Controller

Until now, the discussed event-based approaches have been based on a one-degree-of-freedom (1DOF) structure. In [31], a new event-based PI controller based on a 2DOF structure is used to cope with the set-point following and the load disturbance rejection tasks. The block diagram of the 2DOF event-based controller is shown in Fig. 16.6. The event detection logic of both controllers is located inside one block. In this case, the generation of the control actions u_{ff} and u_{fb} by the compensators C_{ff} and C_{fb} is triggered by state events obtained from the set-point value and the control error signal.

For the set-point following task, the solution is a design procedure that, by considering a first-order-plus-dead-time (FOPDT) model of the process and a pre-designed open-loop control action, produces an event-based feedforward controller C_{ff} that provides a required process variable transition with only two events. Once the process reaches the steady state, an event-based feedback PI controller C_{fb} is in charge of rejecting disturbances and maintaining the process inside a deadband around y_{sp}. C_{fb} starts calculating proportional and integral actions only when the process output moves outside the dead band, and it stops when the process output is again inside the band. As in the approach described in the previous section, the feedback control action is calculated due to the triggering of P- or I-type events.

As explained before, in this approach it is assumed that the process to be controlled exhibits FOPDT dynamics, namely, it can be modeled by the following transfer function

$$P(s) = \frac{K}{Ts+1} e^{-Ls}. \tag{16.27}$$

A process output transition from 0 to y_{sp} can be obtained by applying the following open-loop control action:

$$u_{ff}(t) = \begin{cases} \bar{u}_{ff} & \text{if } t < \tau, \\ y_{sp}/K & \text{if } t \geq \tau \end{cases} \tag{16.28}$$

where the value of $\bar{u}_{ff}$ is determined, after trivial calculations, in such a way that the process output y (which is necessarily zero until time $t = L$) is y_{sp} at time $t = \tau + L$. This results in the following

$$\bar{u}_{ff} = \frac{y_{sp}/K}{1 - e^{-\tau/T}}. \tag{16.29}$$

The corresponding process output is

$$y(t) = \begin{cases} 0 & \text{if } t < L, \\ \bar{u}_{ff} K (1 - e^{\frac{t-L}{T}}) & \text{if } L \leq t \leq L + \tau, \\ y_{sp} & \text{if } t > L + \tau. \end{cases} \tag{16.30}$$

It is important to note that if the value $\bar{u}_{ff}$ of the control variable u is selected a priori from (16.29), this value must be held constant for a time interval

$$\tau = -T \log\left(1 - \frac{y_{sp}}{K \bar{u}_{ff}}\right) \tag{16.31}$$

to obtain the desired output transition (16.30). Figure 16.7 shows the open-loop situation to be repeated by C_{ff} and the two events where the control action must change.

Consider now a proportional (event-based) feedback controller

$$C_{ff}(s) = K_p^{ff} \tag{16.32}$$

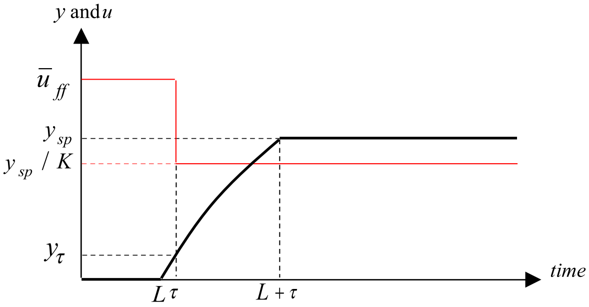


Fig. 16.7 Open-loop process response to be repeated in closed-loop by the two-event-based controller C_{ff}

where K_p^{ff} is selected to provide a control action as similar as possible to the previously described open-loop control action (16.28). For this purpose, consider that, at time $t = 0$, the control action is

$$u(0) = K_p^{ff} e(0) = K_p^{ff} y_{sp}. \tag{16.33}$$

Replacing (16.33) in (16.31), where $\bar{u}_{ff} = u(0)$, causes the initial control action to be kept constant for a time interval of duration

$$\tau = -T \log\left(1 - \frac{y_{sp}}{K K_p^{ff} y_{sp}}\right) = -T \log\left(1 - \frac{1}{K K_p^{ff}}\right). \tag{16.34}$$

Applying the control action (16.33) in (16.5) results in the following value of the process variable at time $t = \tau$

$$y_\tau = K K_p^{ff} y_{sp}\left(1 - e^{-\frac{\tau - L}{T}}\right). \tag{16.35}$$

For the proportional controller to provide the same control action as the open-loop controller described previously, the control action must switch to the required steady-state value (see (16.28)) at time $t = \tau$

$$u_\tau = \frac{y_{sp}}{K}. \tag{16.36}$$

Thus, it must be

$$K_p^{ff}(y_{sp} - y_\tau) = \frac{y_{sp}}{K} \tag{16.37}$$

which, by taking into account (16.35), can be rewritten as

$$K_p\left[1 - K K_p^{ff}\left(1 - e^{-\frac{\tau - L}{T}}\right)\right] = \frac{1}{K}. \tag{16.38}$$

From (16.38), it is expressed as

$$\tau = L - T \log\left(\frac{(K K_p^{ff})^2 - K K_p^{ff} + 1}{(K K_p^{ff})^2}\right). \tag{16.39}$$

Finally, the value of K_p^{ff} that makes the proportional control action equal to the open-loop one (16.28) can be determined by equating the values obtained in (16.34) and (16.39), namely, by solving the equation

$$-T\log\left(1-\frac{1}{KK_p^{ff}}\right)=L-T\log\left(\frac{(KK_p^{ff})^2-KK_p^{ff}+1}{(KK_p^{ff})^2}\right). \tag{16.40}$$

Posing $X:=KK_p^{ff}$ results in the following second-order equation

$$X^2\big(1-e^{L/T}\big)-X\big(1-e^{L/T}\big)+1=0. \tag{16.41}$$

Then, the value of K_p^{ff} can be computed easily from (16.41), and τ and y_τ can be obtained from (16.34) and (16.35), respectively. The algorithm for the set-point following task can thus be described as follows:

```
function u=2dof_pid_sp(ysp,y)

  % K,T,L are the parameters of the FOPDT process

  Kpff=max(roots([1-exp(L/T), -(1-exp(L/T)), 1]))/K;
  tau=-T*log(1-1/(K*Kpff));
  ytau=K*Kpff*ysp*exp(1-exp(-(tau-L)/T));

  % Calculate control variable
  if (abs(y)>abs(ytau)) % event detected
    u=Kpff*ysp;
  else
    u=Kpff*ytau;
  end
```

It is worth noting that, because the algorithm is based on the process model, a decrement in the performance can be expected in case of modeling uncertainties. However, a technique to handle the trade-off between aggressiveness and robustness can be implemented easily (see [31] for details).

Regarding the load disturbance rejection task, a deadband is first defined around the set-point value, depending on the control (steady-state) specifications

$$y_{ll}\le y_{sp}\le y_{ul}. \tag{16.42}$$

The proportional and integral parts of the controller are then enabled once the process is inside the deadband, and they start calculating at the very moment that the process leaves the band due to a disturbance. The event-based solution consists of applying a level crossing sampling strategy to each part to trigger the computation of the control action. In particular, the proportional action is calculated every time that the absolute difference between the current error and the error in the last crossing is greater than Δ_P

$$\left|e(t_{\text{act}}) - e(t_{\text{last}})\right| \geq \Delta_P. \tag{16.43}$$

The integral term is triggered every time a logical expression based on the level crossing of the integrated error IE is true

$$\left|\int_{t_{\text{last}}}^{t_{\text{act}}} e(t)\,dt\right| = \left|\text{IE}(t_{\text{act}}) - \text{IE}(t_{\text{last}})\right| \geq \Delta_I. \tag{16.44}$$

Thus, a new integral action will be produced if IE changes more than Δ_I with respect to the last time the previous logical expression became true.

The following algorithm for the load disturbance rejection task can thus be sketched:

```
% yul and yll are lower and upper limit of the deadband
% values of e, elast, IE, up and ui should be
  initialized to zero

function u=2dof_pid_load(y)

  if (y>yul)
    e=y-yul;
  elseif (y<yll)
    e=yll-y;
  else
    e=0;
    e_last=0;
  end

  IE=IE+hnom*e;

  % Detecting P-type events
  if (abs(e-e_last)>Delta_P)
    up=Kp*e;
    e_last=e;
  end

  % Detecting I-type events
  if (abs(IE)>Delta_I)
    ui=ui+Kp/Ti*IE;
    IE=0;
  end

  u=up+ui;
```

Note that hnom corresponds to the sampling period used to evaluate the logical conditions. In the previous listing, the code corresponding to the control input generator of P and I parts is so simple that it has been included in the event detector.

In the previous description, the set-point following and the disturbance rejection tasks were not coupled because the controller C_{fb} was not enabled until the process output entered the deadband for the first time. A simple time-triggered solution to couple both tasks would consist of setting a maximum allowable time to reach the deadband and to enable C_{fb} if a time-out occurs. Two effective event-based techniques that can be used to couple the two control tasks and that can provide guidelines for the tuning of the parameters are proposed in [31].

16.12 Conclusions

In this chapter, the state-of-the-art of event-based PID control has been analyzed and pseudo-codes to implement the different PID algorithms have been provided. The analysis began with the first PID published in the control-engineering literature and concluded with one of the latest contributions to the field, a 2DOF event-based PI controller. From this chapter, it is evident that even though the event-based paradigm is simple to adapt to a PID control law and provides many advantages that take into account resource utilization or the use of low-resolution sensors, there are open problems that require further attention from researchers.

The first problem is the absence of a standardized event-based PID control law. There are different ways to calculate the integral term, to produce the activation of the control input generator, and to define the type and composition of the event-based condition. Sometimes, these solutions are incompatible. Another open issue, which is closely related to the previous one, is the lack of specific formal methodologies to analyze the performance of these PID controllers. There are simulations and a few experimental evaluations that demonstrate the good performance of these controllers in local or networked configurations. There are researchers working to derive event-driven system theories; however, until now, these efforts have been focused on particular event-driven control structures. However, there are no specific design or formal analysis methods for a standardized event-based PID control law. To some extent, this situation prevents an event-based PID controller from being considered a useful alternative in some specific industrial applications.

One of the most relevant absences in the literature may be that of specific tuning rules. Some authors provide guidelines based on intensive simulations or obtain the controller parameters from tuning rules designed for continuous or discrete standard PID controllers. However, the guidelines are only provided for particular, non-generic architectures. Moreover, event-based PID controllers possess more parameters than their time-triggered counterparts (one or two thresholds, minimum and maximum inter-event times).

Finally, it is worth stressing that the design of event-based PID controllers for multiple-input-multiple-output (MIMO) or single-input-multiple-output (SIMO) processes must still be addressed. Leaving the formal analysis aside, the extension to MIMO or SIMO systems would not be very difficult. It would consist of associating an event-based condition and each sensor or a combination of several sensors to activate the control input generator. However, as before, some methodology to define the event-based conditions, the threshold values, and the controller activation is necessary.

Acknowledgements This work has been supported by the Spanish CICYT under grant DPI2007-61068 by the IV PRICIT (Plan Regional de Ciencia y Tecnología de la Comunidad de Madrid, 2005–2009) under grant S-0505/DPI/0391, and by funds from the Italian Ministry of Education, University, and Research (MIUR).

References

1. Årzén, K.E.: A simple event-based PID controller. In: Proc. of the 14th World Congress of IFAC, Beijing, China (1999)
2. Åström, K.J.: Event based control. In: Astolfi, A., Marconi, L. (eds.) Analysis and Design of Nonlinear Control Systems: In Honor of Alberto Isidori. Springer, Berlin (2008)
3. Åstrom, K.J., Bernhardsson, B.M.: Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In: Proc. of the 41st IEEE Conference on Decision and Control, Las Vegas, NV, USA (2002)
4. Åström, K.J., Hägglund, T.: Advanced PID Control. ISA Press, Research Triangle Park (2006)
5. Cervin, A., Åström, K.J.: On limit cycles in event-based control systems. In: Proc. of the 46th IEEE Conference on Decision and Control (CDC'07), New Orleans, LA, USA (2007)
6. Cervin, A., Henningsson, T.: Scheduling of event-triggered controllers on a shared network. In: Proc. of the 47th IEEE Conference on Decision and Control, Cancun, Mexico (2008)
7. Chacón, J., Sánchez, J., Dormido, S.: Experimental study of a pure event-based PI controller in a wireless control system. In: Proc. of the 9th Portuguese Conference on Automatic Control (CONTROLO'2010), Coimbra, Portugal (2010)
8. Dorf, R.C., Lember, M.C., Phillips, C.A.: Adaptive sampling frequency for sampled data control systems. IRE Trans. Autom. Control **7**(1), 38–47 (1962)
9. Durand, S., Marchand, N.: An event-based PID controller with low computational cost. In: Proc. of the 8th International Conference on Sampling Theory and Applications (SampTA'09), Special Session on Sampling and Industrial Applications, Marseille, France (2009)
10. Durand, S., Marchand, N.: Further results on event-based PID controller. In: Proc. 10th European Control Conference (ECC'09), Budapest, Hungary (2009)
11. Gupta, S.: Increasing the sampling efficiency for a control system. IEEE Trans. Autom. Control **8**(3), 263–264 (1963)
12. Heemels, W.P.M.H., Sandee, J.H., van den Bosch, P.P.J.: Analysis of event-driven controllers for linear systems. Int. J. Control **81**(4), 571–590 (2008)
13. Henningsson, T., Cervin, A.: Comparison of LTI and event-based control for a moving cart with quantized position measurements. In: Proc. of the 10th European Control Conference (ECC'09), Budapest, Hungary (2009)
14. Henningsson, T., Johannesson, E., Cervin, A.: Sporadic event-based control of first-order linear stochastic systems. Automatica **44**(11), 2890–2895 (2008)
15. Hsia, T.: Analytic design of adaptive sampling control law in sampled-data systems. IEEE Trans. Autom. Control **19**(1), 39–42 (1974)
16. Kofman, E., Braslavsky, J.: Level crossing sampling in feedback stabilization under data rate constraints. In: Proc. 45th IEEE International Conference on Decision and Control, San Diego, CA, USA (2006)

17. Lehmann, D., Lunze, J.: Extension and experimental evaluation of an event-based state-feedback approach. Control Eng. Pract. **19**(2), 101–112 (2011)
18. Lunze, J., Lehmann, D.: A state-feedback approach to event-based control. Automatica **46**(1), 211–215 (2010)
19. Marchand, N.: Stabilization of Lebesgue sampled systems with bounded controls: the chain of integrators case. In: Proc. of the 17th World Congress of IFAC, Seoul, Korea (2008)
20. Miskowicz, M.: Sampling of signals in energy domain. In: Proc. of the 10th IEEE Conference on Emerging Technologies and Factory Automation (EFTA'05), Catania, Italy (2005)
21. Miskowicz, M.: Send-on-delta: An event-based data reporting strategy. Sensors **6**, 49–63 (2006)
22. Miskowicz, M.: Efficiency of level-crossing sampling for bandlimited Gaussian random processes. In: Proc. of the IEEE International Workshop on Factory Communication Systems (WFCS'06), Torino, Italy (2006)
23. Miskowicz, M.: Asymptotic effectiveness of the event-based sampling according to the integral criterion. Sensors **7**(1), 16–37 (2007)
24. Mitchell, J.R., McDaniel, W.L. Jr: Adaptive sampling technique. IEEE Trans. Autom. Control **Ac-11**, 200–201 (1969)
25. Otanez, P., Moyne, J., Tilbury, D.: Using deadbands to reduce communication in networked control systems. In: Proc. of the American Control Conference (ACC'2002), Anchorage, USA (2002)
26. Pawlowski, A., Guzmán, J.L., Rodríguez, F., Berenguel, M., Sánchez, J., Dormido, S.: Event-based control and wireless sensor network for greenhouse diurnal temperature control: A simulated case study. In: Proc. of the 13th IEEE International Conference on Emerging Technologies and Factory Automation (EFTA-2008), Hamburg, Germany (2008)
27. Rabi, M., Baras, J.S.: Level-triggered control of a scalar linear system. In: Proc. of the 15th IEEE Mediterranean Conference on Control and Automation, Athens, Greece (2007)
28. Rabi, M., Johansson, K.H.: Event-triggered strategies for industrial control over wireless networks. In: Proc. of the 4th Annual International Conference on Wireless Internet, Maui, Hawaii, USA (2008)
29. Rabi, M., Johansson, K.H.: Scheduling packets for event-triggered control. In: Proc. of the 10th European Control Conference (ECC'09), Budapest, Hungary (2009)
30. Sánchez, J., Guarnes, M.A., Dormido, S., Visioli, A.: Comparative study of event-based control strategies: An experimental approach on a simple tank. In: Proc. of the 10th European Control Conference (ECC'09), Budapest, Hungary (2009)
31. Sánchez, J., Visioli, A., Dormido, S.: A two-degree-of-freedom PI controller based on events. J. Process Control (2011). doi:10.1016/j.jprocont.2010.12.001
32. Sandee, J.H., Heemels, W., van denBosch, P.P.J.: Event-driven control as an opportunity in the multidisciplinary development of embedded controllers. In: Proc. of the 24th IEEE American Control Conference, Portland, Oregon, USA (2005)
33. Smith, M. Jr.: An evaluation of adaptive sampling. IEEE Trans. Autom. Control **16**(3), 282–284 (1971)
34. Vasyutynskyy, V., Kabitzsch, K.: Implementation of PID controller with send-on-delta sampling. In: Proc. of the International Conference Control (ICC'2006), Glasgow, Scotland (2006)
35. Vasyutynskyy, V., Kabitzsch, K.: Simple PID control algorithm adapted to deadband sampling. In: Proc. of the 12th IEEE Conference on Emerging Technologies and Factory Automation, Patras, Greece (2007)
36. Vasyutynskyy, V., Kabitzsch, K.: First order observers in event-based PID controllers. In: Proc. of the 14th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'2009), Mallorca, Spain (2009)
37. Vasyutynskyy, V., Kabitzsch, K.: Event-based control: overview and generic model. In: Proc. of the IEEE International Workshop on Factory Communication Systems (WFCS'2010), Nancy, France (2010)

38. Vasyuntynskyy, V., Miskowicz, M.: Towards comparison of deadband sampling types. In: Proc. of the IEEE International Symposium on Industrial Electronics (ISIE 2007), Vigo, Spain (2007)
39. Vasyutynskyy, V., Luntovskyy, A., Kabitzsch, K.: Limit cycles in PI control loops with absolute deadband sampling. In: Proc. of the 18th IEEE International Conference on Microwave & Telecommunication Technology (CRIMICO'2008), Sevastopol, Ukraine (2008)
40. Visioli, A.: Practical PID Control. Springer, London (2006)

Chapter 17
Data-Driven PID Controller

Toru Yamamoto

17.1 Introduction

In recent years, many complicated control algorithms such as adaptive control theory and/or robust control theory have been proposed and implemented for real systems. Even though these complicated and subtle control algorithms exist, less sophisticated PID controllers continue to be widely employed in process industries. The reasons for the continued popularity of PID controllers are summarized as follows: (i) the control structure is quite simple; (ii) the physical meaning of the control parameters is clear; and (iii) the operators' know-how can be easily utilized in designing controllers.

Given these reasons, it is still attractive to design PID controllers, but they do have their drawbacks. Since most process systems have nonlinearities, it is difficult to obtain good control performances for such systems simply using the fixed PID parameters. The adaptive or self-tuning PID control schemes [1, 2], have been frequently employed for systems with weak nonlinearity. However, since these methods have been mainly researched and discussed for use with linear systems, it is impossible to obtain suitable control results for systems with strong nonlinearity. Where there is strong nonlinearity, PID parameter tuning methods using neural networks (NN) [3] and genetic algorithms (GA) [4] have been proposed, but there are obstacles to these methods. According to these methods, the learning cost is considerably large, and these PID parameters cannot be adequately adjusted due to the nonlinear properties. Therefore, to the best of our knowledge, there are currently few PID parameter tuning schemes of practical use for nonlinear systems.

In contrast, the development of computers has been extensive and progressive in the same time period. These advances in computers enable them to store, quickly retrieve, and read out an ever-increasing amount of data. With these advantages in

T. Yamamoto (✉)
Div. of Electrical, Systems and Mathematical Engineering, Faculty of Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi-Hiroshima, Hiroshima 739-8527, Japan
e-mail: yama@hiroshima-u.ac.jp

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_17, © Springer-Verlag London Limited 2012

mind, the following method has been proposed based on the computer data storage system: Whenever new data is obtained, the data is stored. Next, similar neighbors to the information requests, called ‘queries’, are selected from the stored data. Additionally, the local model or the local controller is constructed using these neighbors. This is known as a data-driven (DD) technique. The data-driven (DD) techniques called the Just-In-Time (JIT) method [5, 6], the Lazy Learning method [7, 8] and the Model-on-Demand (MoD) method [9], have received much attention in the last decade.

A few similar PID controllers have been already proposed based on the JIT method [10] and the MoD method [11]. Generally, the JIT method is used to supplement the feedback controller with a PID structure, however it is problematic. The tracking property is not reliable enough due to the nonlinearities occurring in the case where reference signals are changed. In this method, the controller does not create any integral action in the whole control system. On the other hand, the latter method has a PID control structure. PID parameters are tuned by operators’ skills, and they are stored in the database in advance. And also, a suitable set of PID parameters is generated using the stored data. Still, good control over the performance cannot be necessarily obtained in the case where nonlinearities are included in the controlled object and/or system parameters are changed. This is due to the fact that PID parameters are not adjusted in an on-line manner corresponding to properties of the controlled object.

In an effort to overcome the shortcomings of the JIT and MoD methods, this chapter describes a design scheme for PID controllers, based on the DD technique. This is called ‘DD-PID’ control scheme [12]. The main feature of the DD-PID control scheme is that the PID parameters are updated corresponding to the control error in an on-line manner and they are stored in the database. Therefore, the learning effect is gradually accumulated in the database. In order to avoid the excessive accumulation of the stored data, an algorithm is proposed which reduces the need for memory storage and computational costs. Finally, the effectiveness of the DD-PID control scheme is examined on some simulation examples, and experimentally evaluated on a pilot-scale heat process control system.

17.2 Data-Driven PID Controller

17.2.1 DD Technique

In order to create the DD technique, several mathematical considerations must first be undertaken. The first consideration is the following discrete-time nonlinear system, expresses as:

$$y(t) = f\big(\phi(t-1)\big), \tag{17.1}$$

where $y(t)$ denotes the system output and $f(\cdot)$ denotes the nonlinear function. In this case, $\phi(t-1)$ is called the ‘*information vector*’ and is defined by the following

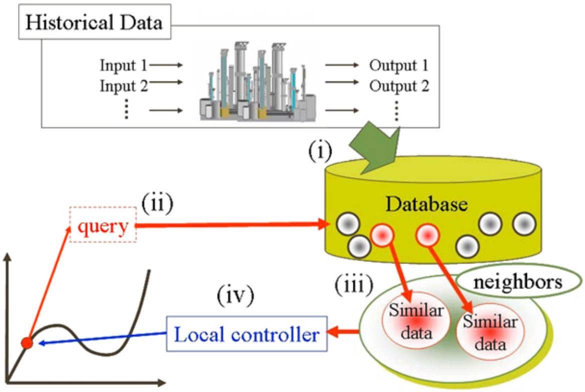


Fig. 17.1 Schematic figure of data-driven approach

equation:

$$\phi(t-1) := \left[y(t-1), \ldots, y(t-n_y), u(t-1), \ldots u(t-n_u)\right], \tag{17.2}$$

where $u(t)$ denotes the control input. Also, n_y and n_u denote the orders of the system output and the control input, respectively. Accordingly, (17.2) means that the controlled object is described using elements included in $\phi(t-1)$. In addition, it is assumed that, as regards the controlled object, the sign of the partial derivative of $y(t)$ with respect to $u(t-1)$ is known. The sign of the partial derivative is also the sign of the system Jacobian, so the system Jacobian is known.

The schematic figure of the DD technique is shown in Fig. 17.1. According to the DD technique, (i) the data is first stored in the form of the information vector ϕ expressed in (17.2). Moreover, (ii) the query $\phi(t-1)$ is required and (iii) after some similar neighbors to the query are selected from the database, (iv) the local controller corresponding to the query can be designed using these neighbors.

17.2.2 Controller Design

The following control law with a PID structure is considered:

$$\begin{aligned}\Delta u(t) &= \frac{k_c T_s}{T_I} e(t) - k_c \left(\Delta + \frac{T_D}{T_s}\Delta^2\right) y(t)\\ &= K_I e(t) - K_P \Delta y(t) - K_D \Delta^2 y(t),\end{aligned} \tag{17.3}$$

where $e(t)$ denotes the control error signal defined by

$$e(t) := r(t) - y(t). \tag{17.4}$$

The reference signal is denoted by $r(t)$. Also, k_c, T_I, and T_D respectively denote the proportional gain, the reset time, and the derivative time, and T_s denotes the sampling interval. Here, K_P, K_I, and K_D included in (17.3) are derived by the relations

$K_P = k_c$, $K_I = k_c T_s / T_I$, and $K_D = k_c T_D / T_s$. Δ denotes the differencing operator defined by $\Delta := 1 - z^{-1}$. However, it is quite difficult to obtain a good control performance due to nonlinearities if the PID parameters (K_P, K_I, K_D) in (17.3) are fixed. Therefore, a controller design scheme is explained, which can adjust PID parameters in an on-line manner corresponding to properties of the system. Thus, instead of (17.3), the following PID control law with time-variant PID parameters is employed:

$$\Delta u(t) = K_I(t) e(t) - K_P(t) \Delta y(t) - K_D(t) \Delta^2 y(t). \tag{17.5}$$

So with these additional pieces of information, (17.5) can be rewritten as the following relations:

$$u(t) = g\big(\phi'(t)\big), \tag{17.6}$$

$$\phi'(t) := \big[\mathbf{K}(t), r(t), y(t), y(t-1), y(t-2), u(t-1)\big], \tag{17.7}$$

$$\mathbf{K}(t) := \big[K_P(t), K_I(t), K_D(t)\big], \tag{17.8}$$

where $g(\phi'(t))$ denotes a linear function composed of terms which are given by (17.7). By substituting (17.6) and (17.7) into (17.1) and (17.2), the following equation can be derived:

$$y(t+1) = h\big(\tilde{\phi}(t)\big), \tag{17.9}$$

$$\tilde{\phi}(t) := \big[y(t), \ldots, y(t-n_y+1), \mathbf{K}(t), r(t), u(t-1), \ldots, u(t-n_u+1)\big], \tag{17.10}$$

where $n_y \geq 3$, $n_u \geq 2$, and $h(\tilde{\phi}(t))$ denotes a nonlinear function which has elements given by (17.10). Therefore, the PID parameters $\mathbf{K}(t)$ are related to the information vector $\bar{\phi}(t)$ which is expressed as

$$\mathbf{K}(t) = F\big(\bar{\phi}(t)\big), \tag{17.11}$$

$$\bar{\phi}(t) := \big[y(t+1), y(t), \ldots, y(t-n_y+1), r(t), u(t-1), \ldots, u(t-n_u+1)\big]. \tag{17.12}$$

Since the future output $y(t+1)$ included in (17.12) cannot be obtained at t, $y(t+1)$ is replaced by $r(t+1)$ because the purpose of the control is to realize $y(t+1) \rightarrow r(t+1)$. Therefore, $\bar{\phi}(t)$ included in (17.12) is newly rewritten as follows:

$$\bar{\phi}(t) := \big[r(t+1), r(t), y(t), \ldots, u(t-1), y(t-n_y+1), \ldots, u(t-n_u+1)\big]. \tag{17.13}$$

After making the above preparations, it is possible to design a new PID control scheme based on the DD technique. The procedure of the controller design is summarized as follows.

[STEP 1] Generate initial database The DD technique cannot work if the historical data is not saved at all, as there will be no information to work with. It is necessary, then, to first create a database based on historical data of the controlled object. Therefore, PID parameters are initially calculated using either the Ziegler & Nichols method [13] or the Chien, Hrones & Reswick (CHR) method [14]. Of course, PID parameters determined through the use of the operators' skill and/ or know-how can be also utilized as the initial database. That is, $\Phi(j)$ indicated in the following equation is generated as the initial database:

$$\Phi(j) := \big[\bar{\phi}(j), \mathbf{K}(j)\big], \quad j = 1, 2, \ldots, N(0) \tag{17.14}$$

where $\bar{\phi}(j)$ and $\mathbf{K}(j)$ are given by (17.13) and (17.8). Moreover, $N(0)$ denotes the number of information vectors stored in the initial database. If one set of fixed PID parameters is chosen as being typical, then all PID parameters included in the initial information vectors may be equal. Expressed numerically, $\mathbf{K}(1) = \mathbf{K}(2) = \cdots = \mathbf{K}(N(0))$ at the initial stage.

[STEP 2] Calculate distance and select neighbors It is necessary though to determine the distances between the query $\bar{\phi}(t)$ and the information vectors $\bar{\phi}(i)$ $(i \neq t)$. These are calculated using the following $\mathcal{L}_1$-norm with some weights:

$$d\big(\bar{\phi}(t), \bar{\phi}(j)\big) = \sum_{l=1}^{n_y+n_u+1} \left| \frac{\bar{\phi}_l(t) - \bar{\phi}_l(j)}{\max_m \bar{\phi}_l(m) - \min_m \bar{\phi}_l(m)} \right|, \\ \big(j = 1, 2, \ldots, N(t)\big) \tag{17.15}$$

where $N(t)$ denotes the number of information vectors stored in the database when the query $\bar{\phi}(t)$ is given. To further explain the variables: $\bar{\phi}_l(j)$ denotes the lth element of the jth information vector, and similarly, $\bar{\phi}_l(t)$ denotes the lth element of the query at t. Among the lth element of all information vectors ($\bar{\phi}(j)$, $j = 1, 2, \ldots, N(t)$) stored in the database, the maximum element is denoted by $\max \bar{\phi}_l(m)$. Similarly, $\min \bar{\phi}_l(m)$ denotes the minimum element. Here, k-pieces with the smallest distances between them are chosen from all information vectors.

[STEP 3] Compute PID parameters Next, using k-neighbors selected in [STEP 2], the suitable set of PID parameters is computed around the query using the following Linearly Weighted Average (LWA) [15]:

$$\mathbf{K}^{\text{old}}(t) = \sum_{i=1}^{k} w_i \mathbf{K}(i), \quad \sum_{i=1}^{k} w_i = 1, \tag{17.16}$$

where w_i denotes the weight corresponding to the ith information vector $\bar{\phi}(i)$ in the selected neighbors. This is calculated by

$$w_i = \sum_{l=1}^{n_u+n_y+1} \left(1 - \frac{[\bar{\phi}_l(t) - \bar{\phi}_l(i)]^2}{[\max_m \bar{\phi}_l(m) - \min_m \bar{\phi}_l(m)]^2} \right). \tag{17.17}$$

Using the PID parameters computed in (17.16), the control input is generated, and the output $y(t+1)$ is measured.

[STEP 4] PID parameters adjustment In the case where the suitable control performance cannot be obtained using the PID parameters computed in [STEP 3], these control parameters have to be updated and stored in the database. That is, it is necessary to adjust the PID parameters so that the control error is decreased. The following steepest descent method is utilized in order to modify the PID parameters:

$$\mathbf{K}^{\text{new}}(t) = \mathbf{K}^{\text{old}}(t) - \eta \frac{\partial J(t+1)}{\partial \mathbf{K}(t)}, \tag{17.18}$$

$$\eta := \text{diag}\{\eta_P, \eta_I, \eta_D\}, \tag{17.19}$$

where η denotes the learning rate, and the following $J(t+1)$ denotes the error criterion:

$$J(t+1) := \frac{1}{2}\varepsilon(t+1)^2, \tag{17.20}$$

$$\varepsilon(t) := y_r(t) - y(t). \tag{17.21}$$

The output of the reference model is denoted by $y_r(t)$, given by

$$y_r(t) = \frac{z^{-1}T(1)}{T(z^{-1})} r(t), \tag{17.22}$$

$$T\big(z^{-1}\big) := 1 + t_1 z^{-1} + t_2 z^{-2}. \tag{17.23}$$

Here, $T(z^{-1})$ is designed based on the following two features:

(i) rise-time, and
(ii) damping property.

Thus, let $T(z^{-1})$ be the denominator of the discrete-time version of the following desirable second-order continuous-time transfer function $G(s)$:

$$G(s) = \frac{1}{1 + \sigma s + \mu(\sigma s)^2}. \tag{17.24}$$

That is, $T(z^{-1})$ is defined as follows:

$$T\big(z^{-1}\big) = 1 + t_1 z^{-1} + t_2 z^{-2}, \tag{17.25}$$

$$t_1 = -2\exp\left(-\frac{\rho}{2\mu}\right)\cos\left(\frac{\sqrt{(}2\mu - 1)}{2\mu}\rho\right), \tag{17.26}$$

$$t_2 = \exp\left(-\frac{\rho}{\mu}\right), \tag{17.27}$$

$$\rho := T_s/\sigma, \tag{17.28}$$

$$\mu := 0.25(1-\delta) + 0.51\delta \tag{17.29}$$

where σ denotes the rise-time. The damping coefficient is denoted by μ. The damping coefficient itself is adjusted by changing δ. Where $\delta = 0$ and $\delta = 1$, the step response of $G(s)$ shows the Binominal model response and the Butterworth model response, respectively.

Moreover, each partial differential of (17.18) is developed as follows:

$$\left.\begin{aligned}
\frac{\partial J(t+1)}{\partial K_P(t)} &= \frac{\partial J(t+1)}{\partial \varepsilon(t+1)} \frac{\partial \varepsilon(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_P(t)} \\
&= \varepsilon(t+1)\big(y(t) - y(t-1)\big)\frac{\partial y(t+1)}{\partial u(t)}, \\
\frac{\partial J(t+1)}{\partial K_I(t)} &= \frac{\partial J(t+1)}{\partial \varepsilon(t+1)} \frac{\partial \varepsilon(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_I(t)} \\
&= -\varepsilon(t+1)e(t)\frac{\partial y(t+1)}{\partial u(t)}, \\
\frac{\partial J(t+1)}{\partial K_D(t)} &= \frac{\partial J}{\partial \varepsilon(t+1)} \frac{\partial \varepsilon(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_D(t)} \\
&= \varepsilon(t+1)\big(y(t) - 2y(t-1) + y(t-2)\big)\frac{\partial y(t+1)}{\partial u(t)}.
\end{aligned}\right\} \tag{17.30}$$

Note that *a priori* information with respect to the system Jacobian which is represented by $\partial y(t+1)/\partial u(t)$ is required in order to calculate (17.30). Here, using the relation $x = |x| \operatorname{sign}(x)$, the system Jacobian can be obtained by the following equation:

$$\frac{\partial y(t+1)}{\partial u(t)} = \left|\frac{\partial y(t+1)}{\partial u(t)}\right| \operatorname{sign}\left(\frac{\partial y(t+1)}{\partial u(t)}\right), \tag{17.31}$$

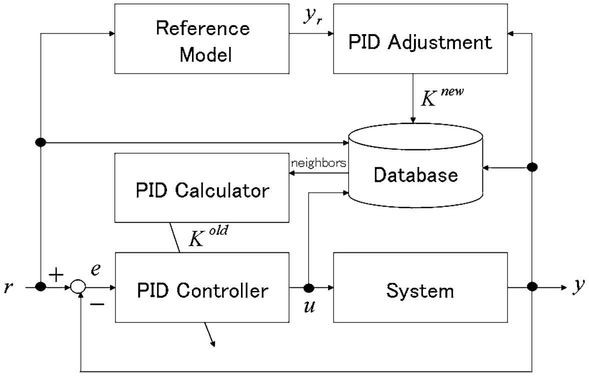
where $\operatorname{sign}(x) = 1(x > 0), -1(x < 0)$. From the assumption mentioned above, the sign of the system Jacobian is obtainable. And by including $|\partial y(t+1)/\partial u(t)|$ in η, the usage of the system Jacobian can be made easy [3]. Therefore, it is assumed that the sign of the system Jacobian is known.

[STEP 5] Remove redundant data In implementing the DD technique in real systems, the DD-PID control scheme has a constraint: the calculations between [STEP 2] and [STEP 4] must be finished within the sampling time. Here, storing the redundant data in the database results in the computer wasting time that could be used on the process. Therefore, an algorithm to avoid excessive stored data is put forward. The procedure is carried out in the following two steps.

[First condition] For information vectors in which k-neighbors are excepted beforehand, the information vectors which satisfy the following condition are extracted:

$$d\big(\bar{\phi}(t), \bar{\phi}(i)\big) \le \alpha_1, \quad i = 1, 2, \dots, N(t) - k. \tag{17.32}$$

Fig. 17.2 Block diagram of the DD-PID control system



Note that the distance is computed using only input/output data $\bar{\phi}$ included in the information vector. Meeting the first condition means that only the information vectors with short distance to the query $\bar{\phi}(t)$ are extracted from the database.

[Second condition] For information vectors extracted in the first condition, the information vectors which satisfy the following condition are extracted:

$$\sum_{l=1}^{3}\left\{\frac{\mathbf{K}_l(i)-\mathbf{K}_l^{\text{new}}(t)}{\mathbf{K}_l^{\text{new}}(t)}\right\}^2 \le \alpha_2, \tag{17.33}$$

where $\mathbf{K}_1$, $\mathbf{K}_2$, and $\mathbf{K}_3$ mean K_P, K_I, and K_D, respectively. Using the above procedure, it is possible to extract and delete from the database redundant data. This is achieved by extracting and deleting from the database information vectors with high similarity in the relationship between the newly generated PID gains and PID gains included in the extracted information vectors. If several information vectors exist which satisfy the second condition, then only the information vector with the smallest value in the second condition is removed.

Note that the query and the corresponding updated PID parameters are always stored in the database. In practice, α_1 and α_2 should be set between 0.1 and 1.0, but some trial and error may be necessary.

The above algorithm is illustrated by the block diagram in Fig. 17.2.

17.3 Simulation Examples

In order to evaluate the effectiveness of the DD-PID control scheme, some simulation examples for nonlinear systems are considered. These include a Hammerstein model and a system with a hysteresis. They are discussed below.

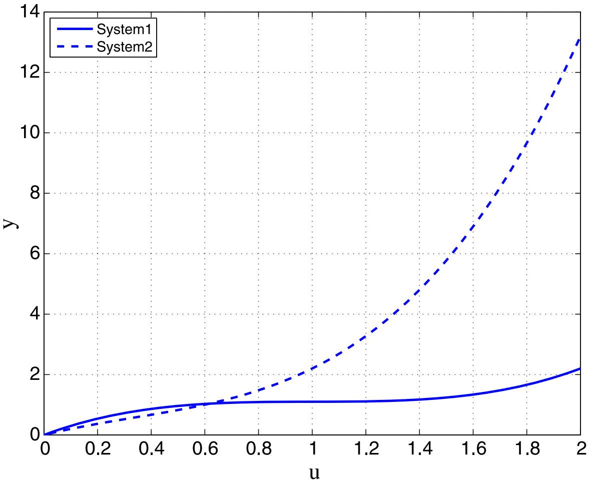


Fig. 17.3 Static input/output relation of System 1 and System 2

17.3.1 Hammerstein Model

To begin, the following Hammerstein models are discussed (see [16] for more information):

[System 1]

$$\left.\begin{aligned} y(t) &= 0.6y(t-1) - 0.1y(t-2) + 1.2x(t-1) - 0.1x(t-2) + \xi(t), \\ x(t) &= 1.5u(t) - 1.5u^2(t) + 0.5u^3(t), \end{aligned}\right\} \tag{17.34}$$

[System 2]

$$\left.\begin{aligned} y(t) &= 0.6y(t-1) - 0.1y(t-2) + 1.2x(t-1) - 0.1x(t-2) + \xi(t), \\ x(t) &= 1.0u(t) - 1.0u^2(t) + 1.0u^3(t), \end{aligned}\right\} \tag{17.35}$$

where $\xi(t)$ denotes the white Gaussian noise with zero mean and a variance of 0.01^2. The static properties of the System 1 and System 2 are shown in Fig. 17.3. Moreover, Fig. 17.4 shows the static gain corresponding to the control input signal. On examining Figs. 17.3 and 17.4, it is apparent that the static gains of System 2 are larger than those of System 1 at $u > 0.6$.

Here, the reference signal $r(t)$ is given as follows:

$$r(t) = \begin{cases} 0.5 & (0 \leq t < 50), \\ 1.0 & (50 \leq t < 100), \\ 2.0 & (100 \leq t < 150), \\ 1.5 & (150 \leq t \leq 200). \end{cases} \tag{17.36}$$

The information vector $\bar{\phi}$ is given by:

$$\bar{\phi}(t) := \big[r(t+1), r(t), y(t), y(t-1), y(t-2), u(t-1)\big]. \tag{17.37}$$

Fig. 17.4 Static gain properties of System 1 and System 2

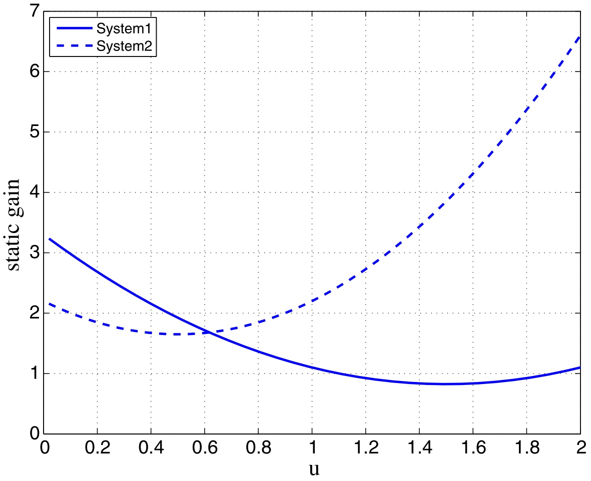


Table 17.1 User-specified parameters included in the DD-PID control scheme (Hammerstein model)

Orders of the information vector	$n_y = 3$
	$n_u = 2$
Number of neighbors	$k = 6$
Learning rates	$\eta_P = 0.8$
	$\eta_I = 0.8$
	$\eta_D = 0.2$
Coefficients to inhibit the data	$\alpha_1 = 0.5$
	$\alpha_2 = 0.1$
Initial number of data	$N(0) = 6$

The desired characteristic polynomial $T(z^{-1})$ included in the reference model was designed as follows:

$$T\left(z^{-1}\right) = 1 - 0.271z^{-1} + 0.0183z^{-2}, \tag{17.38}$$

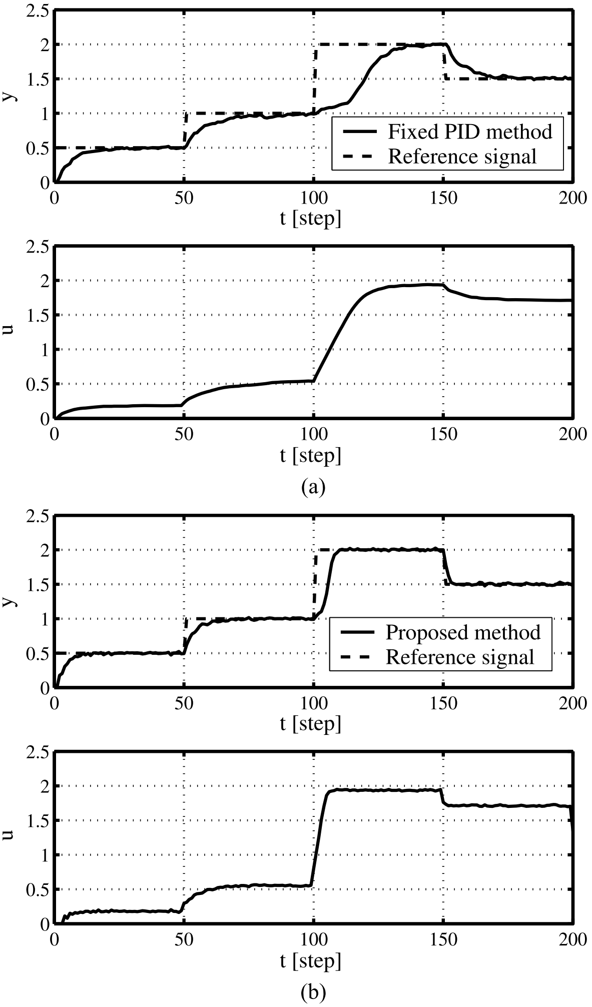
where $T(z^{-1})$ was designed by setting σ as 1.0 and δ as 0.0. Here, T_s is equal to 1.0. Table 17.1 shows the user-specified parameters as determined in the DD-PID control scheme.

For the purpose of comparing the DD-PID control scheme with conventional schemes, the fixed PID control scheme widely used in industrial processes is first employed. These PID parameters, tuned by the CHR method [14], are as follows:

$$K_P = 0.486, \qquad K_I = 0.227, \qquad K_D = 0.122. \tag{17.39}$$

The control results for System 1 (17.34) are shown in Fig. 17.5 using the fixed PID controller and the DD-PID control scheme. Trajectories of the PID parameters in the DD-PID control scheme are summarized in Fig. 17.6.

Fig. 17.5 Control results using the fixed PID control (**a**) and the DD-PID control scheme (**b**) for System 1



From Fig. 17.5, it is clear that the control results of the fixed PID controller are not good. This is owing to nonlinearities of the controlled object. The nonlinearities had the most influence in the third step where the control responses became slow. On the other hand, looking at Figs. 17.5 and 17.6, it is apparent that good control results can be obtained using the DD-PID control scheme because PID parameters are adequately adjusted.

The adjustment of PID parameters merits further discussion. Figure 17.7 shows the behavior of the updating scheme set out in [STEP 4], where the reference signal is changed from 0.5 through 1.0 around 50 [step]. According to Fig. 17.7, it is clear that a new set of PID parameters is gradually built up in the database, and PID parameters are suitably adjusted according to the change of the reference signal or the operating point.

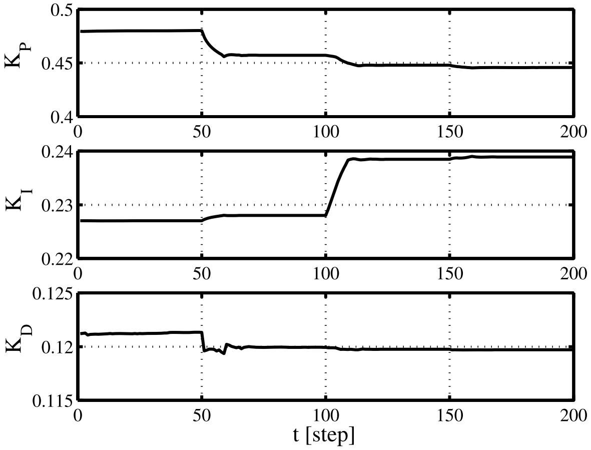


Fig. 17.6 Trajectories of PID parameters corresponding to Fig. 17.5

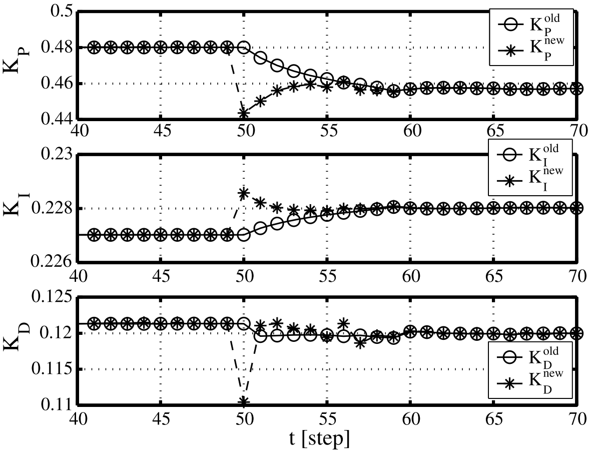


Fig. 17.7 Behavior of PID parameters adjustment around $t = 100$ [step]

Moreover, $*$ in Fig. 17.8 shows the point in which a new information vector is stored in the database. That is, Fig. 17.8 shows that the process of removing the redundant data in [STEP 5] works adequately, and the new data (i.e., the new information vector) is stored instead of the redundant data. Figure 17.8 also illustrates that new data is stored only in the transient state where the reference signals are changed. Note that there is very little replacement of the redundant data by the new data when the system is in a steady state (i.e. where the reference signal is constant). As a result of using the algorithm in [STEP 5] to remove needless data, the number of data items stored in the database was quite low, namely, 38, reduced from an original count of 206. In addition, using the DD-PID control scheme, the error ε_{ep} given by the following criterion is 0.0417,

$$\varepsilon_{ep}(\text{epoc}) := \frac{1}{N}\sum_{t=1}^{N}\left\{\frac{\varepsilon(t)}{r(t)}\right\}^2, \tag{17.40}$$

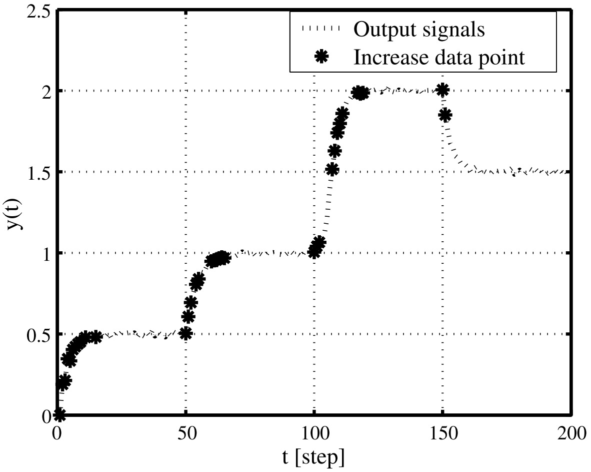


Fig. 17.8 Replacement behavior of the redundant data by the new data, where * shows the point in which a new information vector is stored in the database

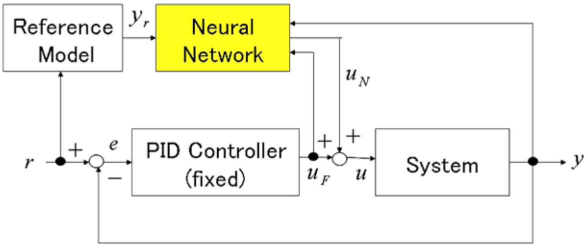


Fig. 17.9 Block diagram of the NN-PID control system considered in this chapter

where N denotes the number of steps per 1 [epoc], and N is set as 200 in this case. 1 [epoc] shows the iteration number for learning, that is, the simulation of 200 steps to be shown in Fig. 17.4 is equivalent to 1 [epoc]. Furthermore, because PID parameters can be adjusted in an on-line manner by the DD-PID control scheme, the number of iterations was set as 1.

Moreover, a certain type of neural net-based PID controller [3], called the NN-PID controller was applied to this system (17.34), where the NN was utilized in order to supplement the fixed PID controller as show in Fig. 17.9. The fixed PID parameters are given by (17.39). The trajectory of the learning error, ε_{ep}, (expressed in (17.40)) is shown in Fig. 17.10, and control results are shown in Fig. 17.11. It is clear from Fig. 17.10 that the number of learning iterations remains 86 [epoc] until the control results using the NN-PID controller can produce the same control performances as the DD-PID control scheme. In other words, until $\varepsilon_{ep} \leq 0.0417$ is satisfied. Since this condition has been satisfied, the effectiveness of the DD-PID control scheme is examined for nonlinear systems.

The next step is to consider the case where the system changes from (17.34) to (17.35) at $t = 70$. First, the control results using the fixed PID controller are shown

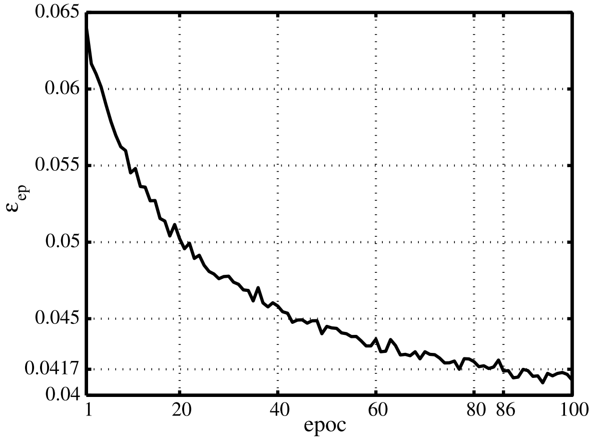


Fig. 17.10 Error behaviors using the NN-PID controller for Hammerstein model

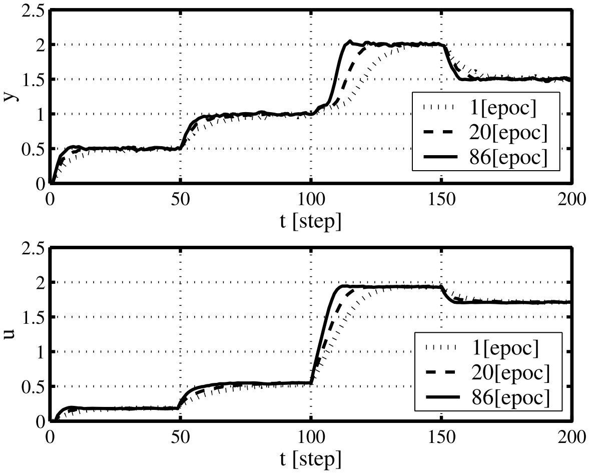


Fig. 17.11 Control result using the NN-PID controller for Hammerstein model

in Fig. 17.12, where the PID parameters used are the same as those shown in (17.39). Although the gain of the controlled object becomes high after $t = 70$, the PID parameters are not changed. Therefore, the control performance becomes oscillatory after $t = 100$. On the other hand, the DD-PID control scheme was also employed in this case. The control results and trajectories of PID parameters are shown in Figs. 17.13 and 17.14, respectively. Figures 17.13 and 17.14 demonstrate that, even if system parameters are changed, a fairly good control performance can be obtained because PID parameters are adjusted adequately using the DD-PID control scheme. This result shows the adaptability of the DD-PID control scheme remarkably well in the Hammerstein model. Next, the DD-PID control scheme was applied to another type of nonlinear system: a system with a hysteresis.

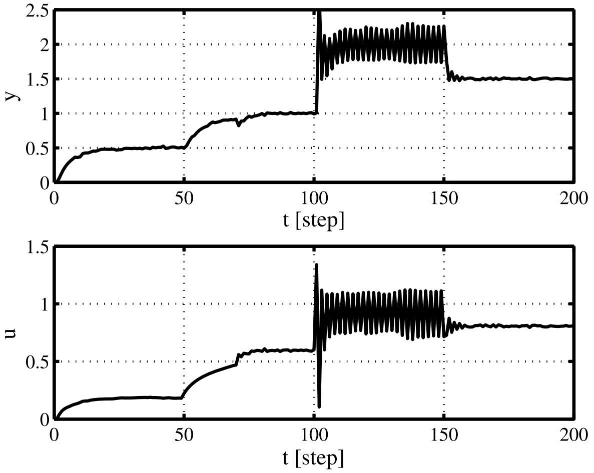


Fig. 17.12 Control result using the fixed PID controller in the case where the system is changed from System 1 to System 2

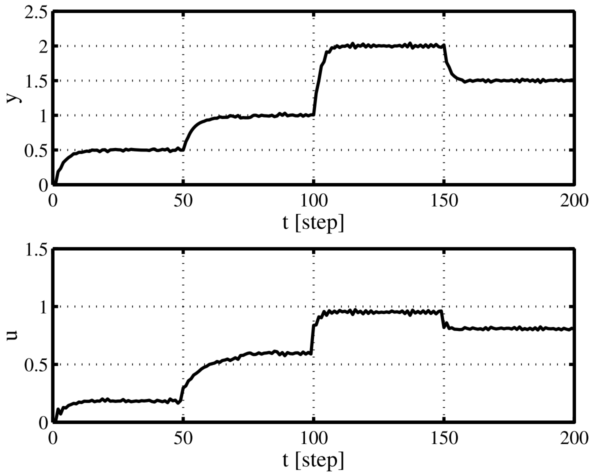


Fig. 17.13 Control result using the DD-PID control scheme in the case where the system parameters are changed

17.3.2 System with a Hysteresis

Next, the following model with another nonlinear type of system was tested. In this system, like that set out by Narendra and Parthasarathy [17], the following equation applies:

$$y(t+1) = \frac{y(t)y(t-1)[y(t)+2.5]}{1+y^2(t)+y^2(t-1)} + u(t) + \xi(t) \tag{17.41}$$

where $\xi(t)$ denotes the white Gaussian noise with zero mean and a variance of 0.01^2. The static property of this model is shown in Fig. 17.15 and discussed below. From Fig. 17.15, it is clear that this model has strong nonlinearities. In particular, there is a kind of hysteresis occurring between $y = 0$ and $y = 2.0$. Here, the reference signal

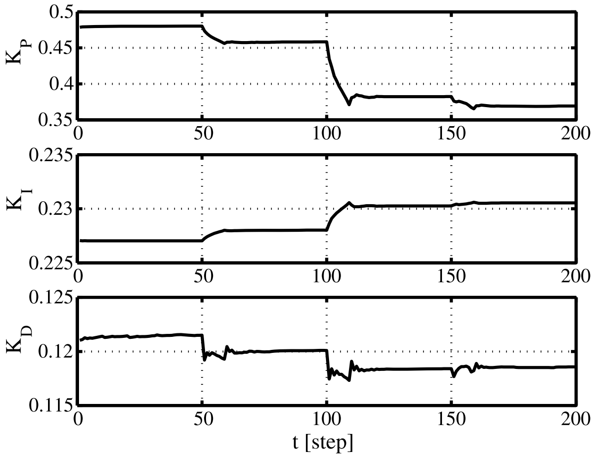


Fig. 17.14 Trajectories of PID parameters corresponding to Fig. 17.13

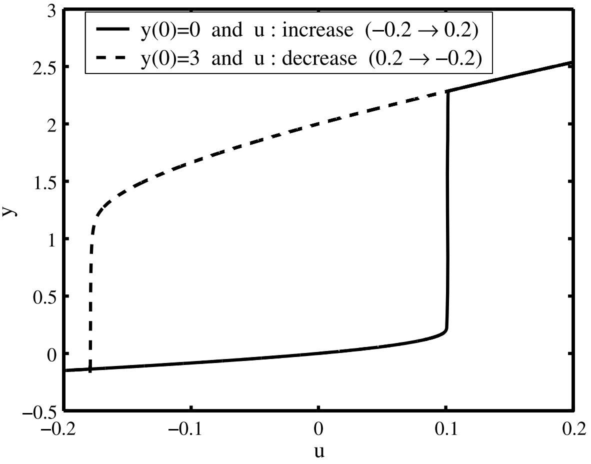


Fig. 17.15 Static property of the system (17.41)

$r(t)$ was given by:

$$r(t) = \begin{cases} 1.5 & (0 \le t < 100), \\ 0.8 & (100 \le t < 200), \\ 2.5 & (200 \le t < 300), \\ -1.0 & (300 \le t \le 400). \end{cases} \tag{17.42}$$

Equation (17.37) was utilized as the information vector $\bar{\phi}(t)$, and the desired characteristic polynomial $T(z^{-1})$ included in the reference model was designed as

$$T\left(z^{-1}\right) = 1 - 0.271z^{-1} + 0.0183z^{-2}, \tag{17.43}$$

where σ and δ are set as 1.0 and 0.0, respectively, and T_s equals 1.0. In addition, the user-specified parameters included in the DD-PID control scheme were determined and are summarized in Table 17.2.

Table 17.2 User-specified parameters included in the DD-PID control schema

Orders of the information vector	$n_y = 3$
	$n_u = 2$
Number of neighbors	$k = 6$
Learning rates	$\eta_P = 0.6$
	$\eta_I = 0.03$
	$\eta_D = 0.01$
Coefficients to inhibit the data	$\alpha_1 = 0.1$
	$\alpha_2 = 0.3$
Initial number of data	$N(0) = 6$

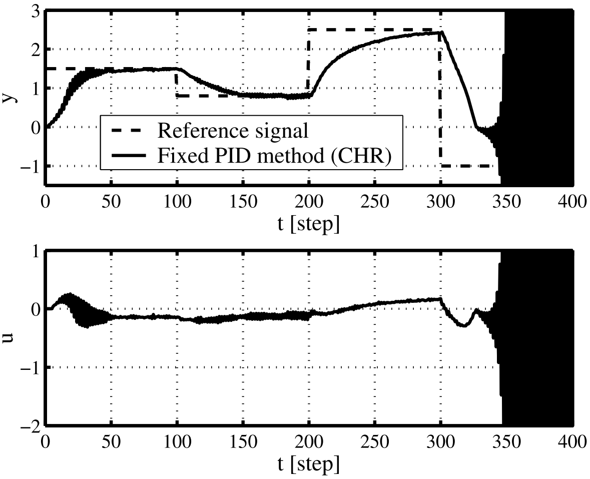


Fig. 17.16 Control result using the fixed PID controller for Ex. 2

For the purpose of comparison, the fixed PID control scheme was first employed, wherein the PID parameters were tuned by the CHR method [14]. Then, PID parameters were calculated as

$$K_P = 0.654, \qquad K_I = 0.028, \qquad K_D = 0.327. \tag{17.44}$$

The control results are shown in Fig. 17.16. As seen in Fig. 17.16, due to nonlinearities of the controlled object, the control results using the fixed PID controller include excessive oscillation around $y = 0.0$. The consequence is that the system falls into an unstable state.

Next, the self-tuning PID control scheme was employed, and the control results are shown in Fig. 17.17. The self-tuning PID control scheme examined here has been previously considered by Yamamoto and Shah in 1994; see [2] for details. Thus, the system parameters were recursively estimated and the PID parameters were computed using the estimates based on the relationship between the generalized minimum variance control and PID control. Due to the strong nonlinearity, the control performance was inferior in the second portion of the reference signal, that is, where $r(t) = 0.8$.

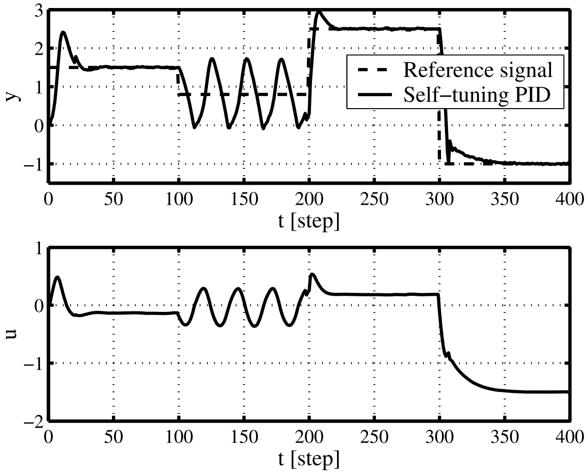


Fig. 17.17 Control result using the self-tuning PID controller for Ex. 2

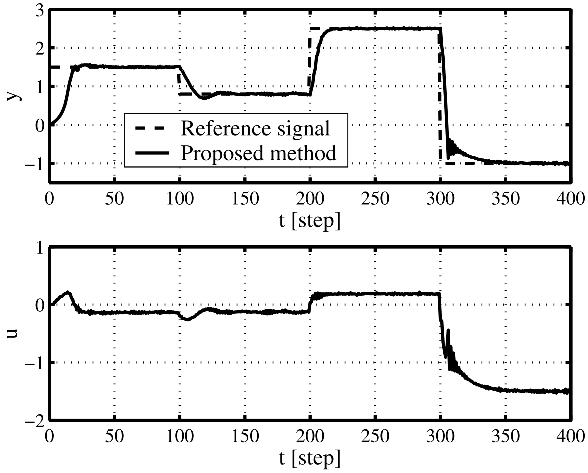


Fig. 17.18 Control result using the DD-PID control scheme for Ex. 2

Finally, the control results using the DD-PID control scheme are shown in Fig. 17.18, and then trajectories of PID parameters are shown in Fig. 17.19. Figures 17.18 and 17.19 show that the superiority of the DD-PID control scheme is quite clear. The PID adjustment in an on-line manner works exceptionally well, and removes the excessive oscillation.

17.4 Application to a Heat Process

The DD-PID control scheme was experimentally evaluated on a pilot-scale temperature control system. Figure 17.20 shows a photograph of this equipment, and the corresponding schematic figure of this system is illustrated in Fig. 17.21.

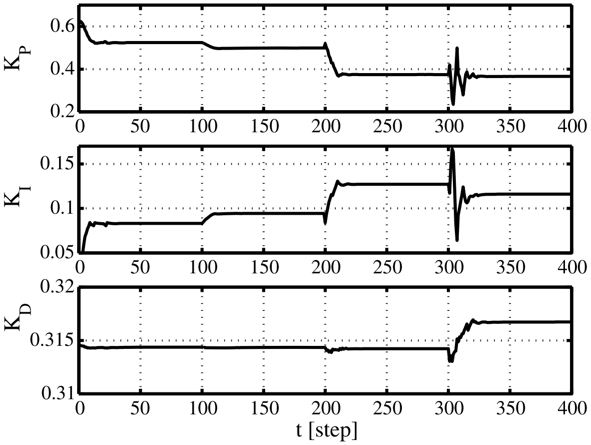


Fig. 17.19 Trajectories of PID parameters corresponding to Fig. 17.18

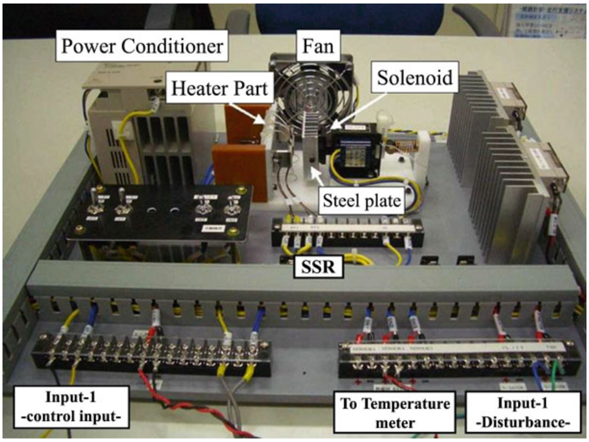


Fig. 17.20 Photograph of the experimental temperature control system

Two heaters were secured on a steel plate. These heaters worked synchronously, corresponding to the input signal from the computer. One thermo-couple was also prepared on the steel plate, and the measured temperature of the steel plate was sent to the computer as the system output signal. The control objective was to regulate the temperature of the steel to the desired reference signal by manipulating the power of the heater. The static input/output relation of the temperature control system are shown in Fig. 17.22. This figure shows that the static gain at $u > 60$ becomes small.

The fixed PID controller and the DD-PID control scheme were employed for this system. The reference temperature signal was changed alternately from 60 to 100 degrees. The fixed PID parameters were computed using the CHR method [14] as follows:

$$K_P = 1.20, \qquad K_I = 0.0120, \qquad K_D = 0.60. \tag{17.45}$$

Figure 17.23 shows the control results. The dotted line and the solid line show the results using the fixed PID and the DD-PID control scheme, respectively. The above

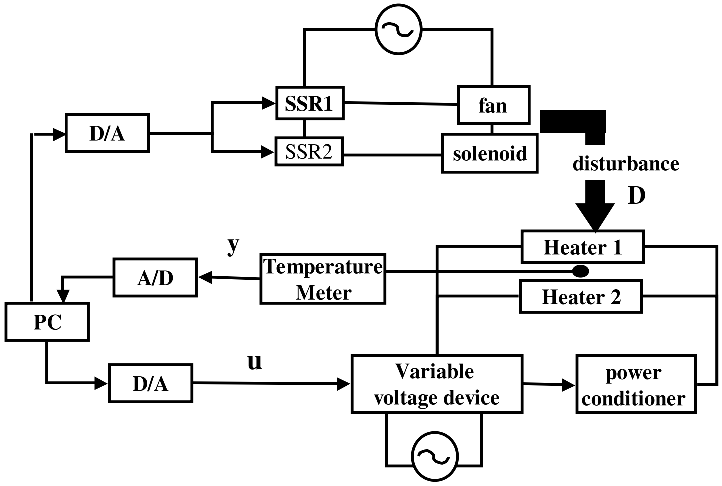
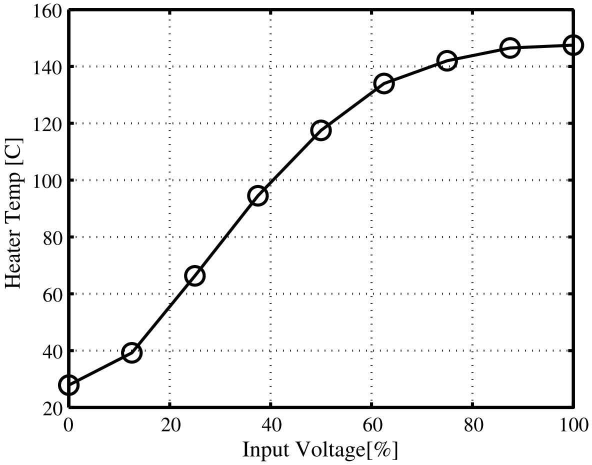


Fig. 17.21 Schematic figure of the experimental temperature control system

Fig. 17.22 Static input/output relation of the temperature control system



PID parameters were determined using the historical data gathered when the system output temperature was around 80 degrees. Because the PID parameters were not suitably calculated, the control results are not so good. In particular, due to the small integral gain, the tracking property to the reference signal is considerably inferior.

On the other hand, according to the DD-PID control scheme, PID parameters were appropriately adjusted in an on-line manner according to the reference signal as shown in Fig. 17.24. Here, the user-specified parameters included in the DD-PID control scheme were determined as shown in Table 17.3, and PID parameters in the initial database were set the same as in (17.45).

Next, the robustness and the adaptability to cyclic disturbances were investigated. Key to this investigation is the solenoid coil (see Figs. 17.20 or 17.21 which is part

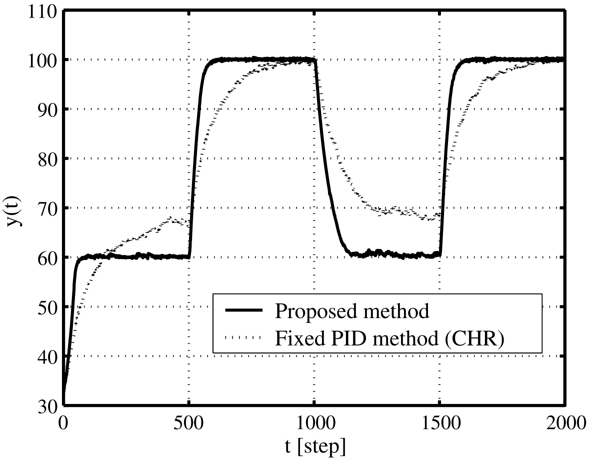


Fig. 17.23 Control result using the fixed PID controller (*dotted line*) and the DD-PID controller (*solid line*)

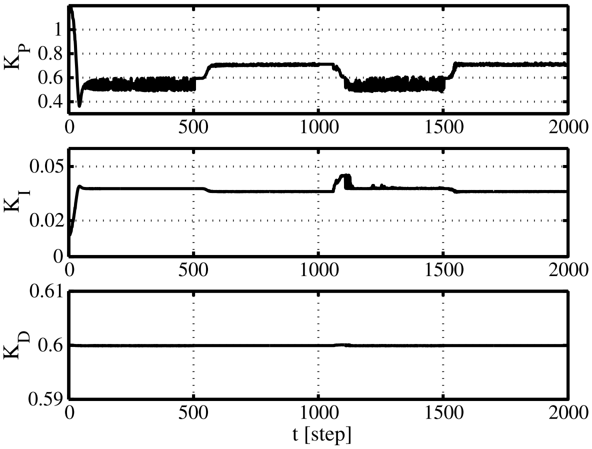


Fig. 17.24 Trajectories of PID parameters corresponding to Fig. 17.23

of the apparatus in this experiment). Because this solenoid works only periodically, it creates a cyclic disturbance in the following way. The solenoid is attached to the second steel plate which comes into contact with the main steel plate. It returns to its original position when the temperature is kept constant. This results in the main steel plate acting as the periodic disturbance. In other words, when the solenoid and the steel plate come in contact, the temperature of the main steel plate falls. The temperature rises again if the steel plate returns.

This cycle of temperature rising and dropping was created as an imitation of a common industrial process wherein the object is first put on the main steel board, and the object is processed while keeping the temperature constant. It moves to the next process when the processing ends, and the next new object is put on the main steel board. This procedure is repeated many times. As a result, if the temperature recovers quickly, a lot of objects can be processed. Therefore, the tracking properties of the PID controller are strongly demanded in industries.

Table 17.3 User-specified parameters included in the DD-PID control scheme (Temperature controller)

Sampling time	$T_s = 1$
Orders of the information vector	$n_y = 3$
	$n_u = 2$
Number of neighbors	$k = 10$
Learning rates	$\eta_P = K_P/100$
	$\eta_I = K_I/100$
	$\eta_D = K_D/10^4$
Coefficients to inhibit the data	$\alpha_1 = 0.5$
	$\alpha_2 = 0.1$
Initial number of data	$N(0) = 10$

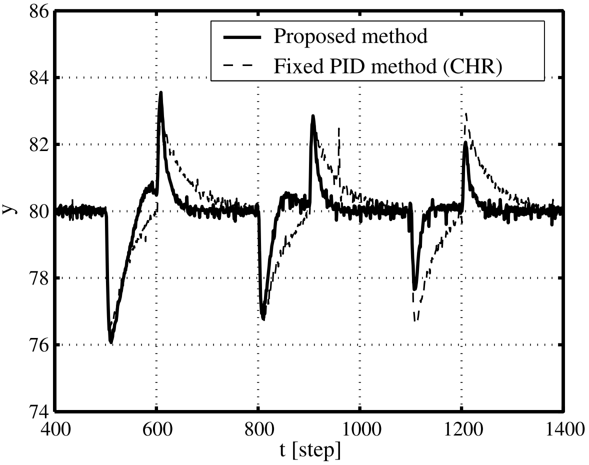


Fig. 17.25 Control result using the fixed PID controller (*dotted line*) and the DD-PID control scheme (*solid line*) in the case where the disturbance is periodically added

For the purpose of comparison, the fixed PID controller and the DD-PID controller were employed for the case where the periodic disturbance is introduced. The control results are summarized in Fig. 17.25: the dotted line shows the result given by the fixed PID controller and the solid line shows the DD-PID control scheme. Moreover, Fig. 17.26 shows the trajectories of PID parameters created by the DD-PID control scheme. Here, the fixed PID parameters were computed using the CHR method [14] as follows:

$$K_P = 0.12, \qquad K_I = 0.0024, \qquad K_D = 0.06. \tag{17.46}$$

And also, the user-specified parameters included in the DD-PID control scheme utilized were the same as those given in Table 17.3. It is clear that the tracking property to the reference signal is gradually improved according to the DD-PID control scheme. The behavior is shown more clearly in Figs. 17.27 and 17.28, where these figures show the control results around $t = 500$ and $t = 1100$, respectively. For the disturbance, another set of PID parameters is immediately extracted from the database corresponding to the current state. It is clear that the DD-PID control

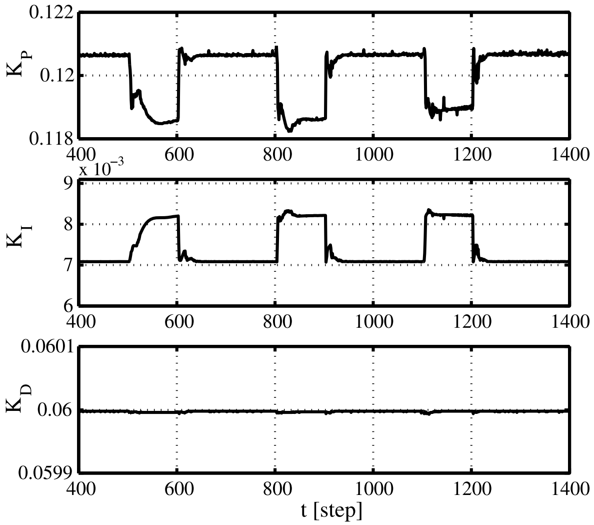


Fig. 17.26 Trajectories of PID parameters corresponding to Fig. 17.25

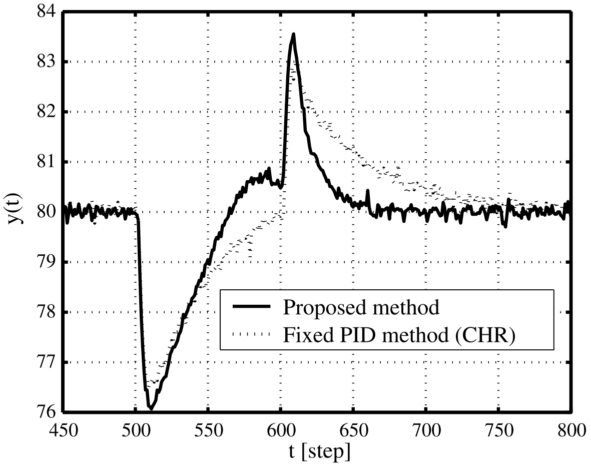


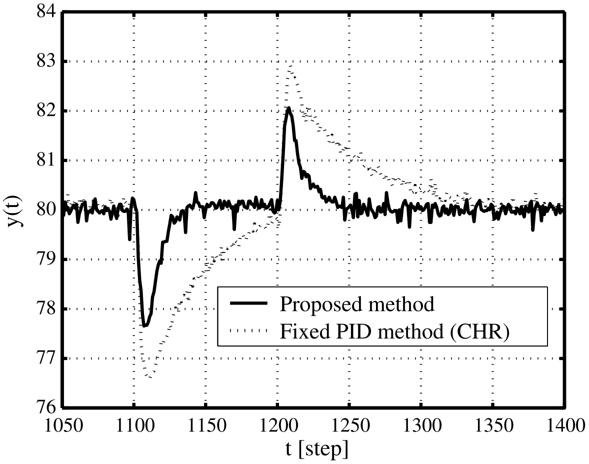
Fig. 17.27 Enlarged figure around $t = 500$ in Fig. 17.25

scheme effectively uses the previously accumulated knowledge. This is one of the typical features of the DD-PID control scheme.

17.5 Conclusions and Future Perspectives

In this chapter, a design scheme of PID controllers based on the DD approach has been explained. To date, many PID controller design schemes using NNs and GAs have been proposed for nonlinear systems. In employing these schemes for real systems, however, the considerably large learning cost becomes a serious problem. This problem can be avoided by using the DD-PID control scheme because such com-

Fig. 17.28 Enlarged figure around $t = 1100$ in Fig. 17.25



putational burdens can be effectively reduced by using the algorithm for removing the redundant data. In addition, the effectiveness of the DD-PID control scheme has been verified by some numerical simulation examples. Given the success of the DD technique for PID controllers on the heat process, there is good reason to continue to explore its success on other systems. Applications of the DD-PID control scheme for existing systems and the extension to multivariate cases are in our future works.

References

1. Åström, K.J., Hägglund, T.: Automatic Tuning of PID Controllers. Instrument Society of America, Research Triangle Park (1988)
2. Yamamoto, T., Shah, S.L.: Design and experimental evaluation of a multivariable self-tuning PID controller. IEE Proc., Control Theory Appl. **151**(5), 645–652 (2004)
3. Omatu, S., Marzuki, K., Rubiyah, Y.: Neuro-Control and Its Applications. Springer, London (1995)
4. Porter, B., Jones, A.H.: Genetic tuning of digital PID controllers. Electron. Lett. **28**, 843–844 (1992)
5. Stenman, A., Gustafsson, F., Ljung, L.: Just in time models for dynamical systems. In: 35th IEEE Conference on Decision and Control, pp. 1115–1120 (1996)
6. Zheng, Q., Kimura, H.: A new just-in-time modeling method and its applications to rolling set-up modeling. Trans. Soc. Instrum. Control Eng. **37**(7), 640–646 (2001) (in Japanese)
7. Zhang, J., Yim, Y., Yang, J.: Intelligent selection of instances for prediction functions in lazy learning algorithms. Artif. Intell. Rev. **11**, 175–191 (1997)
8. Bontempi, G., Birattari, M., Bersini, H.: Lazy learning for local modeling and control design. Int. J. Control **72**(7–8), 643–658 (1999)
9. Stenman, A.: Model on demand: Algorithms, analysis and applications. PhD thesis, Department of Electrical Engineering Linköping University (1990)
10. Zheng, Q., Kimura, H.: Just-in-time PID control. In: Proc. of the 44th Japan Joint Automatic Control Conference, Tokyo, pp. 336–339 (2001)
11. Ohta, J., Yamamoto, S.: Database-driven tuning of PID controllers. Trans. Soc. Instrum. Control Eng. **40**(6), 664–669 (2004) (in Japanese)

12. Yamamoto, T., Takao, K., Yamada, T.: Design of a data-driven PID controller. IEEE Trans. Control Syst. Technol. **17**(1), 29–39 (2009)
13. Ziegler, J.G., Nichols, N.B.: Optimum settings for automatic controllers. Trans. Am. Soc. Mech. Eng. **64**(8), 759–768 (1942)
14. Chien, K.L., Hrones, J.A., Reswick, J.B.: On the automatic control of generalized passive systems. Trans. Am. Soc. Mech. Eng. **74**, 175–185 (1972)
15. Atkeson, C.G., Moore, A.W., Schaal, S.: Locally weighted learning for control. Artif. Intell. Rev. **11**, 75–114 (1997)
16. Zi-Qiang, L.: On identification of the controlled plants described by the Hammerstein system. IEEE Trans. Autom. Control **AC-39**, 569–573 (1994)
17. Narendra, K.S., Parthasarathy, K.: Identification and control of dynamical systems using neural networks. IEEE Trans. Neural Netw. **1**(1), 4–27 (1990)

Chapter 18
Predictive Control Approaches for PID Control Design and Its Extension to Multirate System

Takao Sato

18.1 Introduction

Proportional-Integral-Derivative (PID) control [5, 19, 25, 40, 53, 58] is a very useful and easy control method because its structure is simple and the meaning of the design parameters, referred to as PID parameters, is clear. PID control performance is adjusted by tuning the PID parameters. However, it is rather difficult to find optimal parameters. Hence, numerous methods have been proposed for designing PID parameters [3, 23, 27, 31, 34, 35]. These methods comprise two main classes. In the first, the parameters are designed to satisfy given control specifications, such as phase and amplitude margins. The second is based on comparison and approximation of controllers designed by other methods by a PID controller. In this chapter, PID parameters are designed on the basis of generalized predictive control (GPC) [6, 9]. The control performance of GPC is achieved by a PID controller with a simple control structure, so the obtained controller can be intuitively understood and existing PID controllers will be available in the future.

GPC has been widely studied theoretically and used in industrial applications. It has been formulated in discrete time, but a real controlled plant is a continuous-time system. Hence, a controller is implemented with sampling and holding action, and a digital velocity form of PID controller is designed on the basis of GPC in this chapter. When designing a GPC-based PID controller, there are two problems to be resolved. First, the coefficient polynomials of a GPC law are generally higher than those of a PID controller, hence a GPC law cannot be straightforwardly replaced by a PID controller. Second, the future reference trajectory of GPC needs future information, but PID control essentially consists of the present and the past data.

GPC is often employed for the compensation of process deadtime. To compensate, Tan et al. [45] proposed a predictive PI controller capable of deadtime compen-

T. Sato (✉)
Division of Mechanical System, Department of Mechanical Engineering, School of Engineering, University of Hyogo, 2167 Shosha, Himeji, Hyogo, Japan
e-mail: tsato@eng.u-hyogo.ac.jp

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2_18, © Springer-Verlag London Limited 2012

sation and compared it with a Smith compensator [41, 42]. Further, GPC-based PID control systems were designed [24, 44, 46]. In these methods [44–46], to achieve GPC performance by PID control, PID gains are time varying in the case when processes have the deadtime. Further, a GPC-based PID control system with a Smith predictor was proposed [47]. In this method, PID gains are time invariant, but a control law consists of future predictive state variable.

In Sect. 18.2, the high-order polynomial of a GPC law is approximated by a time-varying proportional gain. To implement the GPC strategy in industry, the GPC-based PID control is applied to a weigh feeder and results are reported in Sect. 18.3. Further, in Sect. 18.4, the GPC-based PID control system is extended for a multirate system in which the sampling interval of the plant output is longer than the update interval of the control input.[1]

18.2 Generalized Predictive Control Based PID Control

18.2.1 Introduction

GPC performance is decided by selecting design parameters, namely, prediction horizon, control horizon, and weighting factor. The computation load mainly depends on the range of the predictive horizon. The infinite future prediction has the same control effect as long range prediction of GPC. Therefore, by using a GPC with a terminal matching condition (abbreviated as γGPC) [20], it is possible to maintain robust stability similar to that in long range predictive control realized in the standard GPC, and to also reduce computation load simultaneously.

When applying γGPC to the design of the PID controller, there is a problem that the γGPC law has a denominator with high order and it also needs several past values of control inputs. However, the PID compensator has at most a first-order denominator and can include only the present and the one-past value as the input. Miller et al. [23] proposed replacing the past control inputs by filtered signals from reference inputs and outputs to resolve this problem. Since their PID controller requires inclusion of filters, the control structure becomes complex. In addition, when plant parameters change, filter redesign is also required. The PID controller is designed on the basis of GPC law using a steady-state gain instead of the denominator polynomial of the control law corresponding to identified parameters at the sampling step [4, 31] to simplify the controller. A method using a time-varying proportional gain is proposed in [3, 35] to improve the approximation of the high-order denominator of the γGPC law. In the approximation, to have a stable proportional gain, the denominator of the original approximated control law must be stable. That is, the approximated control law must have a stable closed-loop and a stable control law simultaneously. In other words, the approximated control law must be strongly stable [10, 16, 52, 57]. This idea was first proposed in the approximation of generalized

[1] This chapter summarizes published papers [28–30].

minimum variance control law [34], and it has been extended into a two-degree-of-freedom GMVC based method [37]. In this study, the approximated control law is replaced by γGPC and a strongly stable γGPC is used. First, this study derives a strongly stable γGPC law, then the PID controller based on the extended γGPC is derived.

If a controlled plant is given by a first-order plus deadtime model, then the extended γGPC law has a second-order numerator and the law is determined exactly by the PID controller. Hence, the obtained PID controller will provide the same control performance as the γGPC law. Because the dynamic characteristics of most chemical plants are well represented as first-order plus deadtime, the use of the derived PID controller achieves good control performance. Therefore, in this study, the plant to be controlled is assumed to have a first-order denominator and deadtime.

In this chapter, z^{-1} denotes the one-step backward shift operator, and $z^{-1}y(k) = y(k-1)$. $A(z^{-1})$ is a polynomial or rational function.

18.2.2 Controlled Plant and PID Controller

Consider a discrete-time ARIMAX model as a controlled plant:

$$\begin{aligned} A(z^{-1})y(k) &= B(z^{-1})u(k-1) + \frac{\xi(k)}{\Delta}, \\ A(z^{-1}) &= 1 + a_1 z^{-1}, \\ B(z^{-1}) &= b_0 + b_1 z^{-1} + \cdots + b_m z^{-m}, \\ \Delta &= 1 - z^{-1} \end{aligned} \tag{18.1}$$

where $u(k)$ is the control input, $y(k)$ is the plant output, and $\xi(k)$ is noise. If a controlled plant has non-zero deadtime, the leading elements of the polynomial $B(z^{-1})$ are 0. When the deadtime is unknown or time-varying, a control system is designed using estimated minimum deadtime [55]. Generally, a plant is a high-order system with deadtime. However, since it is difficult to identify high-order elements accurately, they are approximated as deadtime. Therefore, the plant is expressed by a first-order system with deadtime. Since the deadtime cannot be identified accurately, the order of the polynomial $B(z^{-1})$ is taken higher [56].

Assumption 18.1

- *The degree m of $B(z^{-1})$ is known.*
- *The coefficients $a_1, b_0, b_1, \ldots, b_m$ of $A(z^{-1})$ and $B(z^{-1})$ are unknown, but the nominal values are known.*
- *The polynomials $A(z^{-1})$ and $B(z^{-1})$ are coprime.*
- *$\xi(k)$ is Gaussian white noise.*
- *The deadtime is unknown but is not greater than m.*

A control system is designed using a discrete-time PID controller [53]. The structure of the discrete-time PID control law considered in this study is given as:

$$\Delta u(k) = k_c(k)\bigl(L(1)w(k) - L\bigl(z^{-1}\bigr)y(k)\bigr), \tag{18.2}$$

$$L\bigl(z^{-1}\bigr) = \Delta + \frac{T_s}{T_I} + \frac{T_D}{T_s}\Delta^2 \tag{18.3}$$

where $w(k)$ is the reference input given as piecewise constant components to be followed by the output $y(k)$. The parameters $k_c(k)$, T_I, and T_D are the proportional gain at time t, the reset time, and the derivative time, respectively. Furthermore, T_s denotes the sampling interval; $k_c(k)$ is time-varying, and T_I and T_D are assumed to be constants. The design problem in this study is, using Assumption 18.1, to design the PID gains, $k_c(k)$, T_I, and T_D.

In Sects. 18.2.3 and 18.2.4, the design problem is discussed using the nominal values of coefficients $a_1, b_0, b_1, \ldots, b_m$.

18.2.3 Strongly Stable GPC

To obtain a GPC-based PID controller having a time-varying proportional gain, γGPC [20] is extended into a strongly stable controller using Youla parametrization [16, 52]. In this section, first, a γGPC law is derived in Sect. 18.2.3.1, and next, it is extended into a strongly stable controller in Sect. 18.2.3.2.

18.2.3.1 GPC with Terminal Matching Condition [20]

γGPC gives a control law that minimizes a performance index of the standard GPC by considering steady-state error. γGPC with a short-range prediction has a similar effect on control performance to the effect of GPC with a long-range prediction because a steady-state predictive output works in an equivalent manner to the long-range prediction of GPC.

Using a finite-time predictive output and a steady-state predictive output, the performance index of γGPC is given as

$$J = \sum_{j=N_1}^{N_2} \bigl\{\hat{y}(k+j) - w(k)\bigr\}^2 + \sum_{j=1}^{N_u} \lambda_j \bigl\{\Delta u(k+j-1)\bigr\}^2 + \gamma\bigl\{\hat{y}_s(k) - w_s\bigr\}^2 \tag{18.4}$$

where N_1 is minimum predictive horizon, N_2 is maximum predictive horizon, N_u is control horizon, and λ_j and γ are positive real weighting factors for control inputs and steady-state error, respectively. In this study, to design the PID controller based on the γGPC law, $w(k)$ is used in place of $w(k+j)$ in (18.4). $y_s(k)$ and w_s are the steady-state predictive output, as described herein, and the reference value in the steady state, respectively. The γGPC will be the standard GPC if $\gamma = 0$. The

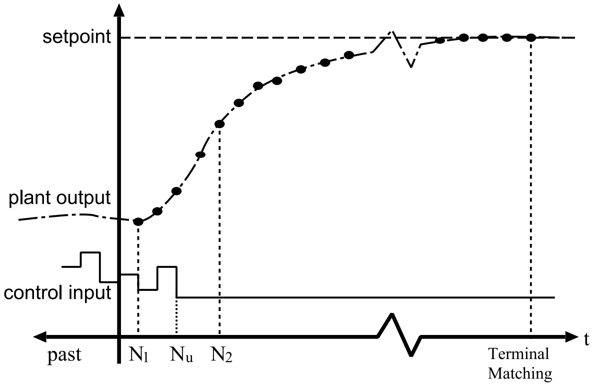


Fig. 18.1 Short prediction with terminal matching condition

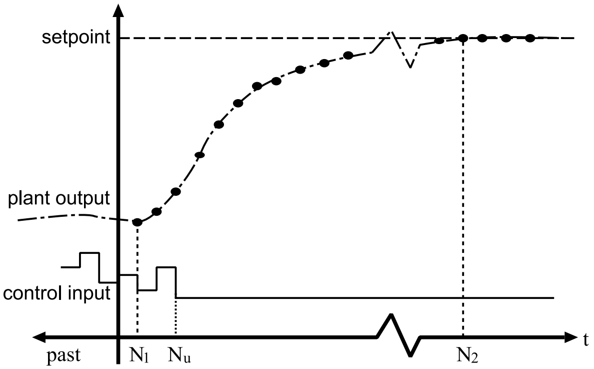


Fig. 18.2 Long prediction without terminal matching condition

concept of a terminal matching condition is shown in Fig. 18.1. Hence, γGPC with a short prediction is similar to GPC with a long prediction (Fig. 18.2).

Diophantine equation (18.5) is solved and $E_j(z^{-1})$ and $F_j(z^{-1})$ are obtained to get a predictive output in future steps [9].

$$1 = \Delta A\left(z^{-1}\right)E_j\left(z^{-1}\right) + z^{-j}F_j\left(z^{-1}\right), \tag{18.5}$$

$$E_j\left(z^{-1}\right) = 1 + e_{j,1}z^{-1} + \cdots + e_{j,j-1}z^{-(j-1)}, \tag{18.6}$$

$$F_j\left(z^{-1}\right) = f_{j,0} + f_{j,1}z^{-1}. \tag{18.7}$$

Furthermore, (18.8) is solved, and $S_j(z^{-1})$ and $R_j(z^{-1})$ are given as [9]:

$$E_j\left(z^{-1}\right)B\left(z^{-1}\right) = R_j\left(z^{-1}\right) + z^{-j}S_j\left(z^{-1}\right), \tag{18.8}$$

$$R_j\left(z^{-1}\right) = r_0 + r_1z^{-1} + \cdots + r_{j-1}z^{-(j-1)}, \tag{18.9}$$

$$S_j\left(z^{-1}\right) = s_{j,0} + s_{j,1}z^{-1} + \cdots + s_{j,m-1}z^{-(m-1)}. \tag{18.10}$$

Using the derived polynomials, the finite time optimal jth step ahead of predictive output is calculated as:

$$\hat{y}(k+j) = R_j\big(z^{-1}\big)\Delta u(k+j-1) + f_j(k), \tag{18.11}$$

$$f_j(k) = F_j\big(z^{-1}\big)y(k) + S_j\big(z^{-1}\big)\Delta u(k-1). \tag{18.12}$$

Based on (18.11) and (18.12), the vector form of the finite time predictive output is given as

$$\hat{\boldsymbol{y}}(k) = R\Delta\boldsymbol{u}(k) + \boldsymbol{f}(k) \tag{18.13}$$

where

$$\begin{aligned}
\hat{\boldsymbol{y}}(k) &= \big[\hat{y}(k+N_1) \quad \hat{y}(k+N_1+1) \quad \dots \quad \hat{y}(k+N_2)\big]^{\mathrm{T}}, \\
\boldsymbol{f}(k) &= \big[f_{N_1}(k) \quad f_{N_1+1}(k) \quad \dots \quad f_{N_2}(k)\big]^{\mathrm{T}}, \\
\Delta\boldsymbol{u}(k) &= \big[\Delta u(k) \quad \Delta u(k+1) \quad \dots \quad \Delta u(k+N_u-1)\big]^{\mathrm{T}}, \\
R &= \begin{bmatrix}
r_{N_1-1} & \cdots & r_0 & 0 & \cdots & 0 \\
r_{N_1} & \ddots & & \ddots & \ddots & \vdots \\
\vdots & & & & \ddots & 0 \\
r_{N_u-1} & \cdots & & & & r_0 \\
\vdots & & & & & \vdots \\
r_{N_2-1} & r_{N_2-2} & \cdots & & \cdots & r_{N_2-N_u}
\end{bmatrix}.
\end{aligned}$$

A lemma due to Kwok and Shah [20] is introduced to obtain steady-state predictive output.

Lemma 18.1 ([20]) *Given $A(z^{-1})$ with its roots all inside the unit circle, for sufficiently large j, the polynomials $F_j(z^{-1})$ in* (18.5) *and $S_j(z^{-1})$ in* (18.8) *converge to finite constant polynomials $F_s(z^{-1})$ and $S_s(z^{-1})$, respectively. Further, the combination of the two identities* (18.5) *and* (18.8) *becomes*

$$z^j\left(\frac{B(z^{-1})}{A(z^{-1})\Delta} - R_j\big(z^{-1}\big)\right) = S_s\big(z^{-1}\big) + \frac{B(z^{-1})F_s(z^{-1})}{A(z^{-1})\Delta} \quad \textit{as } j \to \infty. \tag{18.14}$$

From the above lemma, $F_s(z^{-1})$ and $S_s(z^{-1})$ are given as:

$$F_s\big(z^{-1}\big) = e_s A\big(z^{-1}\big), \quad e_s = 1/A(1),$$

$$S_s\big(z^{-1}\big) = s_{s,0} + s_{s,1}z^{-1} + \cdots + s_{s,m-1}z^{-(m-1)}, \quad s_{s,i} = e_s \sum_{j=i+1}^{m} b_j.$$

Using $F_s(z^{-1})$ and $S_s(z^{-1})$, the steady-state predictive output is given as:

$$\begin{aligned}\hat{y}_s(k) &= \boldsymbol{g}_s^{\mathrm{T}} \Delta \boldsymbol{u}(k) + f_s(k),\\ \boldsymbol{g}_s &= [\underbrace{g_s \quad \dots \quad g_s}_{N_u}]^{\mathrm{T}},\\ g_s &= B(1)/A(1),\\ f_s(k) &= S_s\left(z^{-1}\right)\Delta u(k-1) + F_s\left(z^{-1}\right)y(k).\end{aligned} \tag{18.15}$$

The vector form of (18.4) is described as:

$$\begin{aligned}J &= \left\{\hat{\boldsymbol{y}}(k) - \boldsymbol{w}(k)\right\}^{\mathrm{T}}\left\{\hat{\boldsymbol{y}}(k) - \boldsymbol{w}(k)\right\} + \Delta\boldsymbol{u}(k)^{\mathrm{T}} \Lambda \Delta \boldsymbol{u}(k) + \gamma\left\{\hat{y}_s(k) - w_s\right\}^2,\\ \Lambda &= \operatorname{diag}\{\lambda_1, \dots, \lambda_{N_u}\}\end{aligned} \tag{18.16}$$

where $\boldsymbol{w}(k)$ is defined as

$$\boldsymbol{w}(k) = \big[\underbrace{w(k) \quad \dots \quad w(k)}_{N_2 - N_1 + 1}\big]^{\mathrm{T}}.$$

Taking $\partial J/\partial \Delta \boldsymbol{u}$, $\Delta \boldsymbol{u}(k)$ to minimize (18.16) is given as

$$\Delta \boldsymbol{u}(k) = \left(R^{\mathrm{T}}R + \Lambda + \gamma \boldsymbol{g}_s \boldsymbol{g}_s^{\mathrm{T}}\right)^{-1}\left\{R^{\mathrm{T}}\left(\boldsymbol{w}(k) - \boldsymbol{f}(k)\right) + \gamma\left(w_s - f_s(k)\right)\right\}. \tag{18.17}$$

In the GPC design, receding horizon method is employed. Hence, the first element of the control input series that minimizes the performance index (18.4) is employed as the control input. Hence, a γGPC law is derived as

$$G_p\left(z^{-1}\right)\Delta u(k) = P\left(z^{-1}\right)w(k) - F_p\left(z^{-1}\right)y(k) \tag{18.18}$$

where

$$\begin{aligned}[p_{N_1} \quad \dots \quad p_{N_2}] &= [1 \quad 0 \quad \dots \quad 0]\left(R^{\mathrm{T}}R + \Lambda + \gamma \boldsymbol{g}_s \boldsymbol{g}_s^{\mathrm{T}}\right)^{-1} R^{\mathrm{T}},\\ p_s &= [1 \quad 0 \quad \dots \quad 0]\left(R^{T}R + \Lambda + \gamma \boldsymbol{g}_s \boldsymbol{g}_s^{T}\right)^{-1}\boldsymbol{g}_s,\\ P\left(z^{-1}\right) &= p_{N_1} + \dots + p_{N_2} + p_s,\\ F_p\left(z^{-1}\right) &= p_{N_1}F_{N_1}\left(z^{-1}\right) + \dots + p_{N_2}F_{N_2}\left(z^{-1}\right) + p_s F_s\left(z^{-1}\right),\\ S_p\left(z^{-1}\right) &= p_{N_1}S_{N_1}\left(z^{-1}\right) + \dots + p_{N_2}S_{N_2}\left(z^{-1}\right) + p_s S_s\left(z^{-1}\right),\\ G_p\left(z^{-1}\right) &= 1 + z^{-1}S_p\left(z^{-1}\right).\end{aligned}$$

Substituting the control law (18.18) into the plant (18.1), the closed-loop system is calculated as

$$\begin{aligned} y(k) &= \frac{z^{-1}B(z^{-1})P(z^{-1})}{T(z^{-1})}w(k) + \frac{G_p(z^{-1})}{T(z^{-1})}\xi(k), \\ T(z^{-1}) &= \Delta A(z^{-1})G_p(z^{-1}) + z^{-1}B(z^{-1})F_p(z^{-1}). \end{aligned} \tag{18.19}$$

18.2.3.2 Strongly Stable GPC with Terminal Matching Condition

The derived γGPC law (18.18) is extended using Youla parametrization [52] to obtain a strongly stable control system.

For the coprime factorization approach, the family of stable rational functions is considered [16, 57]:

$$RH_\infty = \left\{ G(z^{-1}) = \frac{G_n(z^{-1})}{G_d(z^{-1})}, G_d(z^{-1}) : \text{ stable polynomial} \right\}.$$

The transfer function is given in the form of a ratio of a rational function in RH_∞:

$$G(z^{-1}) = N(z^{-1})/D(z^{-1}) \tag{18.20}$$

where $N(z^{-1})$ and $D(z^{-1})$ are rational functions in RH_∞ and coprime to each other. Then, all of the stabilizing two-degree-of-freedom compensators are given in Youla parametrization [16, 52, 57]

$$u(k) = C_1(z^{-1})w(k) - C_2(z^{-1})y(k), \tag{18.21}$$

$$C_1(z^{-1}) = \left(Y(z^{-1}) - U(z^{-1})N(z^{-1})\right)^{-1}K(z^{-1}), \tag{18.22}$$

$$C_2(z^{-1}) = \left(Y(z^{-1}) - U(z^{-1})N(z^{-1})\right)^{-1}\left(X(z^{-1}) + U(z^{-1})D(z^{-1})\right) \tag{18.23}$$

where $U(z^{-1})$ and $K(z^{-1})$ are rational functions in RH_∞ and design parameters. $X(z^{-1})$ and $Y(z^{-1})$ are also in RH_∞ and are the solutions of the Bezout equation:

$$X(z^{-1})N(z^{-1}) + Y(z^{-1})D(z^{-1}) = 1. \tag{18.24}$$

It is assumed that the γGPC law (18.18) for the given plant (18.1) is designed such that the closed-loop characteristic polynomial $T(z^{-1})$ is stable.

Comparing the plant (18.1) to the transfer function (18.20), $N(z^{-1})$ and $D(z^{-1})$ in RH_∞ can be chosen as

$$N(z^{-1}) = \frac{z^{-1}B(z^{-1})}{T(z^{-1})}, \tag{18.25}$$

$$D(z^{-1}) = \frac{A(z^{-1})}{T(z^{-1})}. \tag{18.26}$$

Substituting (18.25) and (18.26) into Youla parametrization (18.21)–(18.23) and comparing the form (18.21)–(18.23) having $U(z^{-1}) = 0$ to the control law (18.18), the equations are obtained as:

$$X(z^{-1}) = F_p(z^{-1}), \tag{18.27}$$

$$Y(z^{-1}) = \Delta G_p(z^{-1}), \tag{18.28}$$

$$K(z^{-1}) = P(z^{-1}). \tag{18.29}$$

Then, $X(z^{-1})$, $Y(z^{-1})$, and $K(z^{-1})$ are in RH_∞. Based on Diophantine equation (18.5), it is confirmed that the Bezout equation (18.24) is satisfied by $N(z^{-1})$ in (18.25), $D(z^{-1})$ in (18.26), $X(z^{-1})$ in (18.27), and $Y(z^{-1})$ in (18.28).

$U(z^{-1})$ is used as a newly introduced design parameter for the control law (18.18) instead of choosing $U(z^{-1})$ as $U(z^{-1}) = 0$ to extend the γGPC law (18.18). Using new design polynomials $U_d(z^{-1})$ and $U_n(z^{-1})$, $U(z^{-1})$ is rewritten to simplify the description of the control law:

$$U(z^{-1}) = \Delta \frac{U_n(z^{-1})}{U_d(z^{-1})} T(z^{-1}). \tag{18.30}$$

Substituting (18.25)−(18.30) into (18.21)−(18.23), a newly extended γGPC law is given as

$$G_e(z^{-1})\Delta u(k) = P_e(z^{-1})w(k) - F_e(z^{-1})y(k), \tag{18.31}$$

$$F_e(z^{-1}) = U_d(z^{-1})F_p(z^{-1}) + U_n(z^{-1})\Delta A(z^{-1}), \tag{18.32}$$

$$G_e(z^{-1}) = U_d(z^{-1})G_p(z^{-1}) - U_n(z^{-1})z^{-1}B(z^{-1}), \tag{18.33}$$

$$P_e(z^{-1}) = U_d(z^{-1})P(z^{-1}). \tag{18.34}$$

If parameters $U_d(z^{-1})$ and $U_n(z^{-1})$ are chosen as $U_d(z^{-1}) = 1$, $U_n(z^{-1}) = 0$, the extended control law (18.31) coincides with the original γGPC law (18.18).

Substitution of the extended control law (18.31) into the plant (18.1) gives the closed-loop system

$$y(k) = \frac{z^{-1}B(z^{-1})P(z^{-1})}{T(z^{-1})} w(k) + \frac{G_e(z^{-1})}{T(z^{-1})U_d(z^{-1})} \xi(k). \tag{18.35}$$

Comparing (18.19) with (18.35), the characteristic polynomials of the closed-loop system from a reference input to the plant output are the same. Hence, the characteristic polynomials are independent of the selection of $U_d(z^{-1})$ and $U_n(z^{-1})$. Furthermore, the denominator $G_e(z^{-1})$ of the extended control law will be designed by selecting $U_d(z^{-1})$ and $U_n(z^{-1})$ without changing $T(z^{-1})$, and a strongly stable controller can be obtained.

The denominator of a GPC law must be stable to obtain stable a time-varying proportional gain in a PID controller, to be designed in the next section. The stability of a controller discussed in this study is not $\Delta G_e(z^{-1})$ but $G_e(z^{-1})$. Hence, the new design parameters $U_d(z^{-1})$ and $U_n(z^{-1})$ are designed such that the polynomial $G_e(z^{-1})$ is stabilized.

18.2.4 Self-tuning PID Controller

In this section, a self-tuning PID controller is designed on the basis of strongly stable γGPC. Comparison of the PID controller (18.2) with the strongly stable γGPC law (18.31) gives:

$$L(z^{-1}) = \frac{1}{\tilde{k}_c} F_e(z^{-1}), \tag{18.36}$$

$$k_c(k) = \frac{\tilde{k}_c}{G_e(z^{-1})}. \tag{18.37}$$

If the PID controller is designed on the basis of the original γGPC (18.18), the time-varying proportional gain in the PID controller diverges if denominator $G_p(z^{-1})$ in (18.18) is unstable. Hence, the PID controller having the time-varying proportional gain $k_c(k)$ is designed on the basis of the strongly stable γGPC law (18.31) to obtain stable the time-varying proportional gain.

Equation (18.36) shows that polynomial $F_e(z^{-1})$ should be of second-order because $L(z^{-1})$ is of order 2. However, $F_e(z^{-1})$ becomes a high order polynomial due to the new design polynomial $U_d(z^{-1})$ and $U_n(z^{-1})$ to stabilize $G(z^{-1})$. Hence, the approximation of the polynomial $F_e(z^{-1})$ to second-order is obtained in this way [34]: first, the roots of $F_e(z^{-1}) = 0$ are calculated and the largest or the 'dominant' roots λ_1 and λ_2 are selected such that they correspond to the first and the second largest absolute values $|\lambda_1|$ and $|\lambda_2|$. Second, the approximated polynomial $\bar{F}_e(z^{-1})$ is given as:

$$\begin{aligned} \bar{F}_e(z^{-1}) &= F_e(1)\frac{(\lambda_1 - z^{-1})(\lambda_2 - z^{-1})}{(\lambda_1 - 1)(\lambda_2 - 1)} \\ &= \bar{f}_{e,0} + \bar{f}_{e,1}z^{-1} + \bar{f}_{e,2}z^{-2}. \end{aligned} \tag{18.38}$$

In this study, because the polynomial $A(z^{-1})$ is assumed to be of the first order, if one can find $U_d(z^{-1})$ of the first order and $U_n(z^{-1})$ of the zeroth order, which makes $G_e(z^{-1})$ (18.33) stable, then $F_e(z^{-1})$ will be of the second order. If the order of the polynomial $F_e(z^{-1})$ is higher than 2, the approximation of the polynomial $F_e(z^{-1})$ to second-order is obtained. Then, the PID controller is designed using the approximated $F_e(z^{-1})$.

From (18.3), (18.36) and using notations for $\bar{F}_e(z^{-1})$, the parameters of the PID controller are given as

$$\tilde{k}_c = -(\bar{f}_{e,1} + 2\bar{f}_{e,2}),$$

$$T_I = -\frac{\bar{f}_{e,1} + 2\bar{f}_{e,2}}{\bar{f}_{e,0} + \bar{f}_{e,1} + \bar{f}_{e,2}} T_s, \tag{18.39}$$

$$T_D = -\frac{\bar{f}_{e,2}}{\bar{f}_{e,1} + 2\bar{f}_{e,2}} T_s,$$

$\bar{F}_e(z^{-1}) = F_e(z^{-1})$ if the order of $F_e(z^{-1})$ is 2.

Based on (18.37), the proportional gain $k_c(k)$ is obtained as

$$k_c(k) = \frac{1}{g_{e,0}}\bigl(\tilde{k}_c - g_{e,1}k_c(k-1) - \cdots - g_{e,n_{ge}}k_c(k - n_{g_e})\bigr) \tag{18.40}$$

where the polynomial $G_e(z^{-1})$ is given as

$$G_e\bigl(z^{-1}\bigr) = g_{e,0} + g_{e,1}z^{-1} + \cdots + g_{e,n_{ge}}z^{-n_{ge}},$$
$$n_{g_e} = m + \max\{n_{u_n} + 1, n_{u_d}\}$$

where n_{u_d} and n_{u_n} are the orders of $U_d(z^{-1})$ and $U_n(z^{-1})$, respectively.

If the coefficients $a_1, b_0, b_1, \ldots, b_m$ of the plant are known, then the tuning parameters of the PID controller can be calculated through (18.39) and (18.40). In this study, since the case with the unknown coefficients is considered, the estimated values $\hat{a}_1(k), \hat{b}_0(k), \hat{b}_1(k), \ldots, \hat{b}_m(k)$ of the unknown parameters $a_1, b_0, b_1, \ldots, b_m$ are obtained using the recursive least squares identification law [12, 43]:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \frac{\Gamma(k-1)\psi(k-1)}{1 + \psi^{\mathrm{T}}(k-1)\Gamma(k-1)\psi(k-1)}, \varepsilon(k)$$
$$\Gamma(k) = \Gamma(k-1) + \frac{\lambda\Gamma(k-1)\psi(k-1)\psi^{\mathrm{T}}(k-1)\Gamma(k-1)}{1 + \lambda\psi^{\mathrm{T}}(k-1)\Gamma(k-1)\psi(k-1)},$$
$$\varepsilon(k) = \Delta y(k) - \hat{\theta}^{\mathrm{T}}(k-1)\psi(k-1),$$
$$\hat{\theta}(k) = \begin{bmatrix}\hat{a}_1(k) & \hat{b}_0(k) & \hat{b}_1(k) & \ldots & \hat{b}_m(k)\end{bmatrix}^{\mathrm{T}},$$
$$\psi(k-1) = \begin{bmatrix}-\Delta y(k-1) & \Delta u(k-1) & \Delta u(k-2) & \ldots & \Delta u(k-m-1)\end{bmatrix}^{\mathrm{T}},$$
$$\Gamma(0) = \alpha I \quad (0 < \alpha < \infty)$$

where λ is the forgetting factor $(0 < \lambda < 2)$, and $\Gamma(k)$ is the estimated covariance matrix.

The following assumption is given as:

Assumption 18.2 *There exists a matrix R that satisfies the following equation*:

$$R = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} \psi(k-1)\psi^{\mathrm{T}}(k-1). \tag{18.41}$$

Then, the estimated parameter vector $\hat{\theta}(k)$ converges to its true value θ. Hence,

$$p \cdot \lim_{t \to \infty} \hat{\theta}(k) = \theta \tag{18.42}$$

where $p \cdot \lim$ means convergence in probability [12, 26].

The design scheme of the derived self-tuning PID controller is summarized as follows:

[Self-tuning PID controller]

Step 1 Select minimum and maximum prediction horizon N_1 and N_2, control horizon N_u, weighting factor of control input λ_j, and terminal matching condition γ. Design the parameters of $U_d(z^{-1})$ and $U_n(z^{-1})$, which make $G_e(z^{-1})$ stable, using nominal values of a plant.
Step 2 Obtain the estimates $\hat{a}_1(k)$, $\hat{b}_0(k)$, …, $\hat{b}_m(k)$ of unknown plant parameters using the recursive least squares identification law.
Step 3 Using $\hat{a}_1(k)$, $\hat{b}_0(k), \dots, \hat{b}_m(k)$ obtained in Step 2, and solving Diophantine equation (18.5), get $E_j(z^{-1})$, $F_j(z^{-1})$.
Step 4 Calculate $F_e(z^{-1})$ in (18.32) and $G_e(z^{-1})$ in (18.33), and obtain approximated $\bar{F}_e(z^{-1})$ in (18.38).
Step 5 Calculate the PID parameters using (18.39) and (18.40).
Step 6 Obtain control input $u(k)$ from the PID controller (18.2).
Step 7 Repeat steps from 2 to 6 at each sampling time.

18.2.5 Numerical Example

In this section, a controller is designed for

$$G(s) = \frac{1}{2s+1} e^{-s}. \tag{18.43}$$

Using a sampling time $T_s = 1$ [s], the continuous-time plant (18.43) is transformed into a discrete-time system, and the ARIMAX model is given as

$$\left(1 + a_1 z^{-1}\right) y(k) = 0.40 u(k-2) + \frac{\xi(k)}{\Delta}. \tag{18.44}$$

To confirm the effectiveness of the proposed method on dynamic fluctuation, the time-constant is changed after 120 [s]. Hence, the plant parameter a_1 is given as

$$a_1 = -0.61 \quad (0 \le t < 120);$$
$$a_1 = -0.50 \quad (120 \le t \le 200).$$

The design parameters of γGPC are set as: $N_1 = 1$, $N_2 = 3$, $N_u = 1$, and $\lambda_1 = 0.1$. For the controller with the terminal matching condition, $\gamma = 10$. Because

$P(1)$ is equal to $F_p(1)$, the PID controller (18.2) can be designed on the basis of the γGPC law. However, this γGPC law has an unstable pole that is -1.008. Hence, the new design parameters $U_d(z^{-1})$ and $U_n(z^{-1})$ are set to $1.0 - 0.55z^{-1}$, and -1.4, respectively, to stabilize the control law. These parameters are chosen such that the polynomial $F_e(z^{-1})$ is second-order, and $G_e(z^{-1})$ is stable. The poles of the extended control law, that is, the zeros of $G_e(z^{-1})$, are -0.46 and 0.20×10^{-5}, and all of them are stable. In this simulation, $F_e(z^{-1})$ and $G_e(z^{-1})$ are unknown because the plant parameters are unknown. Hence, these polynomials are calculated using identified plant parameters $\hat{a}_1(k)$ and $\hat{b}_0(k)$. Then, the PID parameters are obtained using $\hat{F}_e[t : z^{-1}]$ and $\hat{G}_e[t : z^{-1}]$. After 120 steps, a_1 is changed to -0.5. In this case, the denominator of the γGPC law is unstable, but it can be stabilized using $U_d(z^{-1})$ and $U_n(z^{-1})$.

Simulations are conducted using the following conditions: simulation length is 200 steps, the variance of random disturbance $\xi(k)$ is 0.02^2, the reference input $w(k)$ is a rectangular wave with amplitude 1.0 and a period of 50 steps, and the recursive least squares identification law having reset with the forgetting factor value of 0.99 is used. The initial value of the estimated covariance matrix is $10^{-1}I$, and those of the identified coefficients are the nominal values which are the true values of (18.44) multiplied by 0.8.

Figure 18.3 shows the simulated result obtained using γGPC. The response has slight overshoot and stays within the range of the white noise disturbance. Using the proposed PID controller having a time-varying proportional gain, simulation result is shown in Figs. 18.4 and 18.5. To achieve $G_e(z^{-1})$ by the proportional gain, it fluctuates at the start of simulation (Fig. 18.5). It follows from the simulation result that the proposed PID controller approximates the γGPC well.

For comparison with the proposed method, the plant is controlled using a conventional GPC-based PI controller, in which the proportional gain is not time-varying. The conventional PI controller is designed using the steady-state gain of the denominator of the standard GPC law [4]. The design parameters of the GPC are the same as above, and the conventional PI controller is designed on the basis of the GPC. The simulation result illustrated in Figs. 18.6 and 18.7 shows that the plant output follows the reference input stably but oscillates. This is because the proportional gain is fixed, and the denominator of the GPC law is replaced with its steady-state gain (Fig. 18.7). The standard GPC with a short prediction ($N_2 = 3$) gives insufficient performance. Hence, the conventional GPC-based PI controller is redesigned with a long prediction ($N_2 = 20$). Figures 18.8 and 18.9 show that oscillation can be attenuated compared with the case of the short prediction. However, the maximum prediction is not long enough, and longer prediction is needed to improve transient response.

In the design of the GPC, if the prediction horizon is large, the computation load increases. Since the proposed controller is designed on the basis of γGPC, it has comparable performance to a long prediction using a short prediction with a smaller computational load. In the design of GPC-based PID controllers, the controller of the GPC cannot be achieved because its order is higher than the order of a PID compensator. However, the proposed PID controller approximates γGPC well using the time-varying proportional gain.

Fig. 18.3 Plant output obtained using γGPC

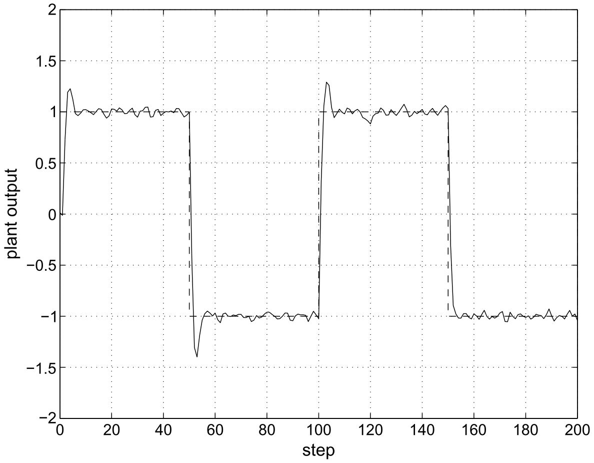
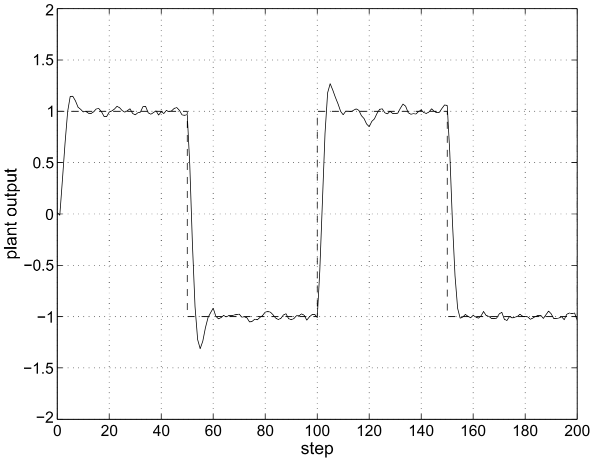


Fig. 18.4 Plant output obtained using strongly stable γGPC-based PID controller with time-varying proportional gain



18.2.6 Conclusion

This study has discussed a design scheme of a self-tuning PID controller. In this design method, PID parameters of a PID controller are decided to approximate γGPC. The derived PID controller has a time-varying proportional gain to obtain better approximation. Furthermore, to have a stable time-varying proportional gain, a strongly stable γGPC which gives a stable closed-loop system and a stable control law simultaneously is derived. Simulated examples have demonstrated its effectiveness. In this study, simulation was conducted using the assumption that the denominator of a GPC law can be stabilized using the new design parameters. However, the stability of a controller is not ensured in the case of an unknown plant parameter. Furthermore, the closed-loop system using this designed PID controller is not the same as that of γGPC. Therefore, further study is required as the stability of both

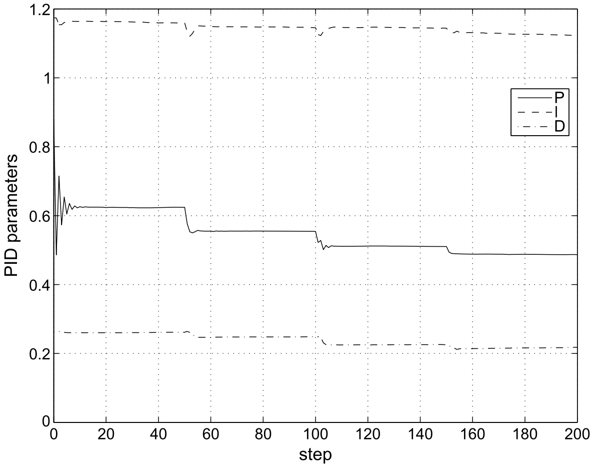


Fig. 18.5 PID parameters of strongly stable γGPC-based PID controller with time-varying proportional gain

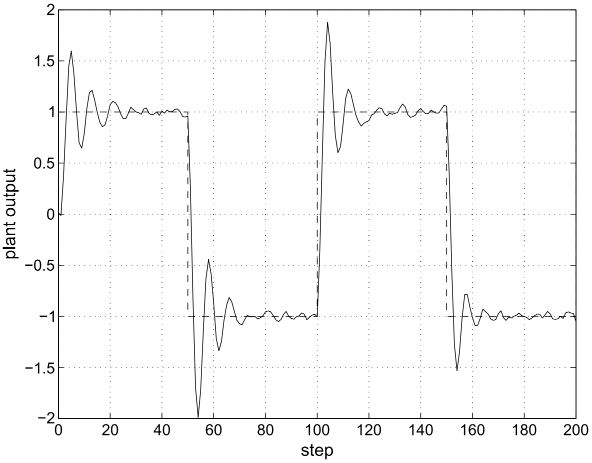


Fig. 18.6 Plant output obtained using standard GPC-based PI controller with fixed proportional gain ($N_2 = 3$)

a controller and a closed-loop system. Model predictive control can deal with constrained problems [22]. Hence, a future work is to extend this design method such that the quadratic program is solved.

18.3 Application to Weigh Feeder

18.3.1 Introduction

Weigh feeders [15] have been developed to dispense material at a specified rate, and they have been employed in various areas [13, 14, 51], e.g., process industry, cement manufacturing, food industry, and so on. Most of weigh feeders operating in industry are mainly designed using proportional and integral compensation.

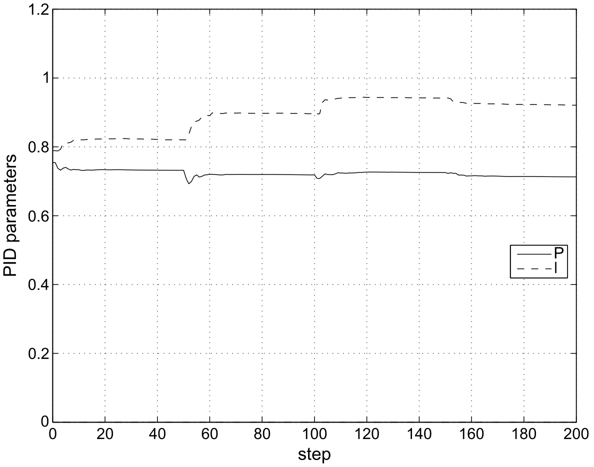


Fig. 18.7 PI parameters of standard GPC-based PI controller with fixed proportional gain ($N_2 = 3$)

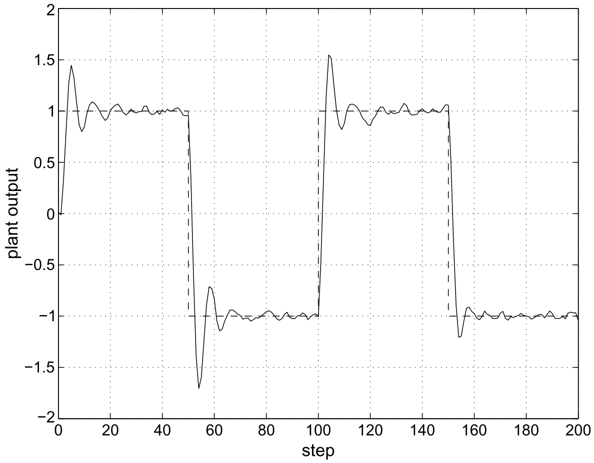


Fig. 18.8 Plant output obtained using standard GPC-based PI controller with fixed proportional gain ($N_2 = 20$)

Since the control performance is decided by the selection of these adjustable controller parameters, the controller parameters must be designed to satisfy required specification. In addition, the dynamic characteristics of a weigh feeder depend on the material to be discharged. Hence, the author has proposed an adaptive control method for a weigh feeder that minimizes the variance of discharged material [33] using generalized minimum variance control (GMVC) [8]. Because the adaptive controller includes parameter identification, good performance can be obtained even if the dynamic characteristics of a weigh feeder are unknown. However, it is difficult to adopt this control method in industry because GMVC is not easy to understand intuitively compared with the PID control. Therefore, in this study, to obtain both a performance upgrade and simple control structure, a PID controller is designed on the basis of GPC.

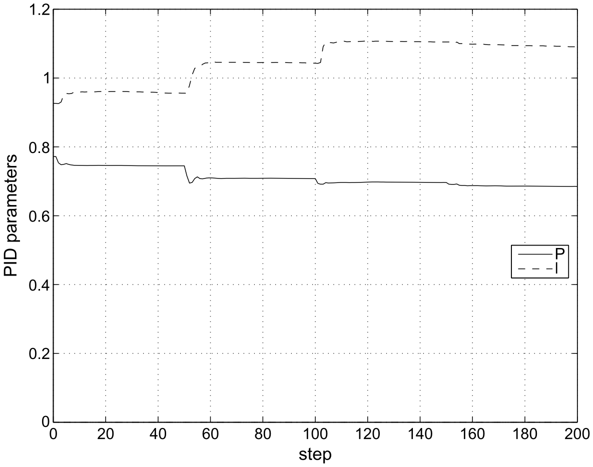


Fig. 18.9 PI parameters of standard GPC-based PI controller with fixed proportional gain ($N_2 = 20$)

In the design of the controller for a weigh feeder, a reference input to be followed by a measured signal is a ramp-type signal [50] because the measured signal is discharged mass. However, conventional GPC-based PID controllers have been designed to have the plant output follow a step-type reference input. Therefore, to apply a GPC-based PID controller to a weigh feeder, the future reference trajectory employed in the design of GPC is represented using the present plant output and a set-point, and a GPC-based PID controller for a ramp-type signal is newly obtained. Consequently, the future reference trajectory can be expressed without future predictive data, and a high order coefficient polynomial is rewritten as a low order polynomial which can be replaced with a PID controller, hence an upgraded PID controller is obtained for controlling a weigh feeder. Finally, to confirm the effectiveness of the design method, experimental results are shown.

18.3.2 Controlled Plant and PID Controller

18.3.2.1 Weigh Feeder

In a weigh feeder that is the controlled object in this study (Fig. 18.10), to have the discharged mass follow to a given reference input, the rotation velocity of a motor, which actuates a discharge mechanism, is controlled by adjusting the control input that is input voltage supplied to an inverter. The discharged mass that is the output signal of a weigh feeder is obtained by Loss-in-Weight method, and it is measured using a loadcell. The dynamic characteristics of this weigh feeder can be approximated as a first order system, an integrator, and a second-order system, and its block diagram is illustrated in Fig. 18.11. In Fig. 18.11, the dynamic characteristics of a motor and a loadcell are approximated as a first-order system and a second-order system, respectively. As for the dynamic characteristics of this weigh feeder, a first-order system and an integrator are dominant [33]. Hence, a controller is designed

Fig. 18.10 Weigh feeder

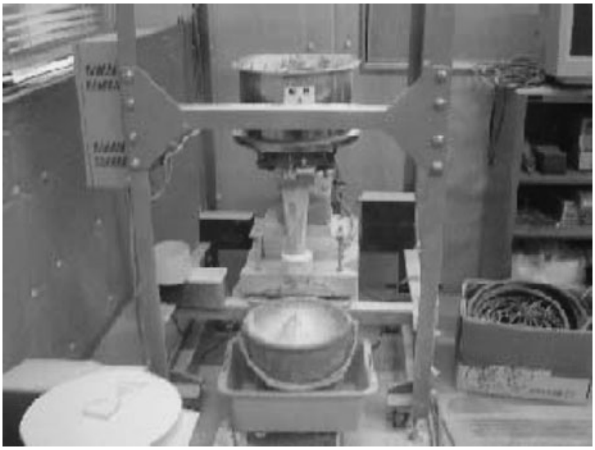
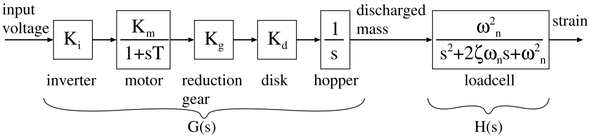


Fig. 18.11 Block diagram of a weigh feeder



using this plant model given as

$$G(s) = \frac{1}{s}\frac{K_m}{1+Ts} \tag{18.45}$$

where K_m is gain and T is assumed to be the time-constant of a motor in this study. Because (18.45) is employed as the dynamic characteristics of this weigh feeder, control performance is influenced by the choice of the plant parameter. Therefore, the proposed controller is designed using a self-tuning controller which recursively estimates unknown parameters.

18.3.2.2 Discrete-Time Model and Controller

A controller for a weigh feeder is designed by the following discrete-time model:

$$\begin{aligned} A(z^{-1})y(k) &= B(z^{-1})u(k-1) + \xi(k), \\ A(z^{-1}) &= (1 - z^{-1})(1 + \alpha_1 z^{-1}) \\ &= 1 + a_1 z^{-1} + a_2 z^{-2}, \\ B(z^{-1}) &= b_0 \end{aligned} \tag{18.46}$$

where z^{-1} is the backward shift operator. Polynomial $B(z^{-1})$ is generally first-order but it is assumed to be zeroth order in this study; $y(k)$ is the plant output or the measured output signal, $u(k)$ is the control input or the input voltage to be supplied to an inverter, and $\xi(k)$ is the noise that disturbs the output signal. Due to the specification of an inverter built in the weigh feeder, the control input is limited as

$$0 \leq u(k) \leq 9.82 \tag{18.47}$$

In this study, a weigh feeder is controlled by the following PID controller:

$$\Delta u(k) = C_1\left(z^{-1}\right)r(k) - C_2\left(z^{-1}\right)y(k), \tag{18.48}$$

$$C_1\left(z^{-1}\right) = k_{c1}\left(\Delta + \frac{T_s}{T_{I1}}\right), \tag{18.49}$$

$$C_2\left(z^{-1}\right) = k_{c2}\left(\Delta + \frac{T_s}{T_{I2}} + \frac{T_{D2}}{T_s}\Delta^2\right),$$

$$\Delta = 1 - z^{-1} \tag{18.50}$$

where k_{c1} and k_{c2} are the proportional gains, T_{I1} and T_{I2} are the integral times, and T_{D2} is the derivative time. $r(k)$ is a ramp-type reference input, and T_s denotes the sampling time.

The purpose of this study is to obtain a self-tuning PID controller which updates its own tuning parameters automatically on the basis of a weigh feeder model which has time-varying parameters.

18.3.3 Design of Controller

A GPC law is derived first, and then, a GPC-Based PID controller is obtained to design the PID parameters based on GPC.

18.3.3.1 GPC

GPC is designed on the basis of minimization of the following performance function:

$$J(k) = E\left[\sum_{j=N_1}^{N_2}\left\{y(k+j) - w(k+j)\right\}^2 + \sum_{j=1}^{N_u}\lambda_j \Delta u(k+j-1)^2\right] \tag{18.51}$$

where N_1, N_2, N_u, and λ_j are minimum predictive horizon, maximum predictive horizon, control horizon, and the weighting factor for the variation in the control input, respectively. In the conventional methods, $w(k+j)$ is replaced with $w(k)$ in (18.51), and the PID controller is based on GPC [3, 23]. Hence, the conventional GPC-based PID controller is deteriorated compared to the original GPC. However, the GPC-based PID control method with the future reference trajectory has been proposed for obtaining control performance as good as GPC [31, 35]. In the conventional design methods [31, 35], the reference input to be followed by the plant output is restricted as the step signal. In this study, the reference input is a ramp-type signal, hence the future reference trajectory is defined as follows:

$$w(k) = y(k), \tag{18.52}$$

$$w(k+j) = (1-\alpha)r(k+j) + \alpha w(k+j-1), \tag{18.53}$$

$$r(k) = kT_s r_g \tag{18.54}$$

$$(0 \le \alpha < 1)$$

where r_g denotes the gradient of the ramp-type reference signal.

Diophantine equations (18.55) and (18.56) are solved to obtain a future predictive output:

$$\begin{aligned} 1 &= \Delta A(z^{-1})E_j(z^{-1}) + z^{-j}F_j(z^{-1}), \\ E_j(z^{-1}) &= 1 + e_1 z^{-1} + \cdots + e_{j-1}z^{-(j-1)}, \\ F_j(z^{-1}) &= f_{j0} + f_{j1}z^{-1} + f_{j2}z^{-2}, \end{aligned} \tag{18.55}$$

$$\begin{aligned} E_j(z^{-1})B(z^{-1}) &= R_j(z^{-1}) + z^{-j}S_j(z^{-1}), \\ R_j(z^{-1}) &= r_0 + r_1 z^{-1} + \cdots + r_{j-1}z^{-(j-1)}, \\ S_j(z^{-1}) &= s_{j0} + s_{j1}z^{-1} + \cdots + s_{jm-1}z^{-(m-1)} \end{aligned} \tag{18.56}$$

where $F_j(z^{-1})$ in Sect. 18.2 is of order 1, but the above-mentioned $F_j(z^{-1})$ is of order 2 because it depends on the order of polynomial $A(z^{-1})$. The j steps ahead predictive output is given as

$$\begin{aligned} y(k+j) &= \hat{y}(k+j) + \varepsilon(k+j), \\ \hat{y}(k+j) &= R_j(z^{-1})\Delta u(k+j-1) + f_j(k), \\ f_j(k) &= F_j(z^{-1})y(k) + S_j(z^{-1})\Delta u(k-1), \\ \varepsilon(k+j) &= E_j(z^{-1})\xi(k+j). \end{aligned} \tag{18.57}$$

The vector form of the optimal predictive outputs is expressed as

$$\hat{\boldsymbol{y}}(k) = R\Delta\boldsymbol{u}(k) + \boldsymbol{f}(k) \tag{18.58}$$

where

$$\begin{aligned} \hat{\boldsymbol{y}}(k) &= \begin{bmatrix} \hat{y}(k+N_1) & \hat{y}(k+N_1+1) & \dots & \hat{y}(k+N_2) \end{bmatrix}^{\mathrm{T}}, \\ \boldsymbol{f}(k) &= \begin{bmatrix} f_{N_1}(k) & f_{N_1+1}(k) & \dots & f_{N_2}(k) \end{bmatrix}^{\mathrm{T}}, \\ \Delta\boldsymbol{u}(k) &= \begin{bmatrix} \Delta u(k) & \Delta u(k+1) & \dots & \Delta u(k+N_u-1) \end{bmatrix}^{\mathrm{T}}, \end{aligned}$$

$$R=\begin{bmatrix} r_{N_1-1} & \cdots & r_0 & & 0 \\ \vdots & & & \ddots & \\ r_{N_u-1} & & & & r_0 \\ \vdots & & & & \vdots \\ r_{N_2-1} & \cdots & \cdots & \cdots & r_{N_2-N_u} \end{bmatrix}.$$

Using the above stated equations, the vector form of the performance function (18.51) is obtained:

$$\begin{aligned} J(k) = E\big[&\big(R\Delta\boldsymbol{u}(k)+\boldsymbol{f}(k)-\boldsymbol{w}(k)\big)^{\mathrm{T}}\big(R\Delta\boldsymbol{u}(k)+\boldsymbol{f}(k)-\boldsymbol{w}(k)\big) \\ &+\Delta\boldsymbol{u}(k)^{\mathrm{T}}\Lambda\Delta\boldsymbol{u}(k)\big]+E\big[\boldsymbol{\varepsilon}(k)^{\mathrm{T}}\boldsymbol{\varepsilon}(k)\big] \end{aligned} \tag{18.59}$$

where

$$\begin{aligned} \boldsymbol{w}(k) &= \big[w(k+N_1) \quad w(k+N_1+1) \quad \ldots \quad w(k+N_2)\big]^{\mathrm{T}}, \\ \boldsymbol{\varepsilon}(k) &= \big[\varepsilon(k+N_1) \quad \varepsilon(k+N_1+1) \quad \ldots \quad \varepsilon(k+N_2)\big]^{\mathrm{T}}, \\ \Lambda &= \mathrm{diag}\{\lambda_1,\ldots,\lambda_{N_u}\}. \end{aligned}$$

Since (18.59) is quadratic in $\Delta\boldsymbol{u}(k)$, the optimal solution with respect to $\Delta\boldsymbol{u}(k)$ is obtained as

$$\Delta\boldsymbol{u}(k)=\big(R^{\mathrm{T}}R+\Lambda\big)^{-1}R^{\mathrm{T}}\big(\boldsymbol{w}(k)-\boldsymbol{f}(k)\big). \tag{18.60}$$

The use of Receding Horizon gives the following control law:

$$G\big(z^{-1}\big)\Delta u(k)=P\big(z^{-1}\big)w(k+N_2)-F\big(z^{-1}\big)y(k) \tag{18.61}$$

where

$$\begin{aligned} P\big(z^{-1}\big) &= p_{N_2}+p_{N_2-1}z^{-1}+p_{N_2-2}z^{-2}+\cdots+p_{N_1}z^{-(N_2-N_1)}, \\ G\big(z^{-1}\big) &= 1+z^{-1}S\big(z^{-1}\big), \\ S\big(z^{-1}\big) &= p_{N_1}S_{N_1}\big(z^{-1}\big)+p_{N_2}S_{N_2}\big(z^{-1}\big)+\cdots+p_{N_2}S_{N_2}\big(z^{-1}\big), \\ F\big(z^{-1}\big) &= p_{N_1}F_{N_1}\big(z^{-1}\big)+p_{N_2}F_{N_2}\big(z^{-1}\big)+\cdots+p_{N_2}F_{N_2}\big(z^{-1}\big), \\ [p_{N_1} \quad \ldots \quad p_{N_2}] &= [1 \quad 0 \quad \ldots \quad 0]\big(R^{\mathrm{T}}R+\Lambda\big)^{-1}R^{\mathrm{T}}. \end{aligned}$$

18.3.3.2 Calculation of PID Parameters

The gradient of a ramp-type reference signal is assumed to be constant (r_g: constant) to design the PID parameters based on the GPC law. In this case, instead of (18.52)–

(18.54), the future reference input is expressed as

$$w(k+j)=\left(\left(1-\alpha^j\right)k+j-\sum_{j=1}^{j}\alpha^i\right)T_s r_g+\alpha^j y(k). \tag{18.62}$$

Using (18.62), the first term on the right-hand side of (18.61) is rewritten as

$$\begin{aligned}
P\left(z^{-1}\right)w(k+N_2)&=p_{r1}r(k)+p_{r2}\Delta r(k)+p_y y(k),\\
p_{r1}&=\sum_{j=N_1}^{N_2}p_j\left(1-\alpha^j\right),\\
p_{r2}&=\sum_{j=N_1}^{N_2}p_j\left(j-\sum_{i=1}^{j}\alpha^i\right),\\
p_y&=\sum_{j=N_1}^{N_2}p_j\alpha^j.
\end{aligned} \tag{18.63}$$

Hence, the GPC law (18.61) is rearranged as

$$G\left(z^{-1}\right)\Delta u(k)=p_{r1}r(k)+p_{r2}\Delta r(k)-\left(F\left(z^{-1}\right)-p_y\right)y(k). \tag{18.64}$$

Comparison of the PID controller (18.48) with the GPC law (18.64) shows that the following equations are required to obtain a GPC-Based PID controller with the ramp-type reference input:

$$C_1\left(z^{-1}\right)=\frac{1}{G(z^{-1})}(p_{r1}+\Delta p_{r2}), \tag{18.65}$$

$$C_2\left(z^{-1}\right)=\frac{1}{G(z^{-1})}\left(F\left(z^{-1}\right)-p_y\right). \tag{18.66}$$

It follows from the condition equations that

$$k_{c1}=\frac{1}{\nu}p_{r2}, \tag{18.67}$$

$$T_{I1}=\frac{p_{r2}}{p_{r1}}T_s, \tag{18.68}$$

$$k_{c2}=-\frac{1}{\nu}(f_1+2f_2), \tag{18.69}$$

$$T_{I2}=-\frac{f_1+2f_2}{f_0-p_y+f_1+f_2}T_s, \tag{18.70}$$

$$T_{D2}=-\frac{f_2}{f_1+2f_2}T_s \tag{18.71}$$

where

$$F(z^{-1}) = f_0 + f_1 z^{-1} + f_2 z^{-2}, \tag{18.72}$$

$$\nu = G(1). \tag{18.73}$$

In this study, the design method of a GPC-based PID controller is proposed using approximation (18.73) in case the deadtime of a controlled plant is longer than 1. However, the obtained PID controller for controlling the weigh feeder is not deteriorated by this approximation and it has the same performance as GPC. This is because the deadtime of the weigh feeder is 1.

If the reference input $r(k)$ is a step-wise function, the second term on the right hand side of (18.64) is eliminated, and the proposed GPC-Based PID controller is equivalent to the conventional GPC-Based PID controller [31, 35].

18.3.3.3 Role of Design Parameter α

The role of the design parameter α is shown. It follows from (18.67)–(18.71) that $C_1(z^{-1})$ and the integral time of $C_2(z^{-1})$ are changed but k_{c2} and $T_D 2$ are not changed by adjusting the design parameter α.

Next, based on $f_{j0} + f_{j1} + f_{j2} = 1$ and $f_{j1} + 2f_{j2} = -j$, right-hand side terms of (18.69) and (18.70) are rewritten as

$$f_1 + 2f_2 = \sum_{j=N_1}^{N_2} p_j(-j), \tag{18.74}$$

$$f_0 + f_1 + f_2 = \sum_{N_1}^{j=N_2} p_j. \tag{18.75}$$

In addition, if $\alpha = 0$,

$$p_{r1} = \sum_{j=N_1}^{N_2} p_j, \qquad p_{r2} = \sum_{j=N_1}^{N_2} p_j j, \qquad p_y = 0. \tag{18.76}$$

Then, $k_{c1} = k_{c2}$, $T_{I1} = T_{I2}$, and further the proposed controller has simple structure:

$$\Delta u(k) = C_1(z^{-1})(r(k) - y(k)) - k_{c1}\frac{T_{D2}}{T_s}\Delta^2 y(k) \tag{18.77}$$

where $A(z^{-1}) = (1 - z^{-1})(1 + \alpha_1 z^{-1})$.

18.3.3.4 Self-tuning Controller

The obtained controller was designed as if the plant parameters were known. Actually, the controller is updated using estimated parameters recursively since the

Table 18.1 PID parameters ($b_0 = 3.0 \times 10^{-6}$)

	P	I	D
$C_1(z^{-1})$	6.0	1.5	–
$C_2(z^{-1})$	6.0	1.5	0.70

Table 18.2 PID parameters ($b_0 = 5.0 \times 10^{-5}$)

	P	I	D
$C_1(z^{-1})$	96	1.5	–
$C_2(z^{-1})$	96	1.5	0.7

plant parameters are actually unknown or time-varying. The estimated parameters are calculated using the following least squares identification law:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \frac{\Gamma(k-1)\psi(k-1)}{1+\psi^{\mathrm{T}}(k-1)\Gamma(k-1)\psi(k-1)}\varepsilon(k),$$

$$\Gamma(k) = \Gamma(k-1) - \frac{\lambda_\Gamma \Gamma(k-1)\psi(k-1)\psi^{\mathrm{T}}(k-1)\Gamma(k-1)}{1+\lambda_\Gamma \psi^{\mathrm{T}}(k-1)\Gamma(k-1)\psi(k-1)},$$

$$\varepsilon(k) = y(k) - \hat{\theta}^{\mathrm{T}}(k-1)\psi(k-1),$$

$$\hat{\theta}(k) = \begin{bmatrix}\hat{\alpha}_1(k) & \hat{b}_0(k)\end{bmatrix}^{\mathrm{T}},$$

$$\psi(k-1) = \begin{bmatrix}-y(k-1) & \dfrac{u(k-1)}{\Delta}\end{bmatrix}^{\mathrm{T}}$$

where λ_Γ is the forgetting factor ($0 < \lambda_\Gamma < 2$), and the initial value of an estimated covariance matrix is $\Gamma[0] = \alpha_\Gamma I$ ($0 < \alpha_\Gamma < \infty$).

18.3.4 *Experiment*

18.3.4.1 Non-adaptive Control

Two case study experiments have been done to confirm the effect of plant parameter selection, where b_0 is set as 3.0×10^{-6} and 5.0×10^{-5}. The sampling time is 0.1 [s], and the gradient of the reference input is 0.01. The design parameters of GPC are set as $N_1 = 1$, $N_2 = 20$, $N_u = 1$, $\lambda_j = 0.01$ ($j = 1, \ldots, N_u$), and $\alpha = 0$. Using the conditions, a GPC-based PID controller has been designed, and the designed controller has been applied for controlling a weigh feeder. Obtained PID parameters are shown in Table 18.1 and 18.2.

The experimental results using the obtained PID parameters are shown in Figs. 18.12, 18.13, 18.14 and 18.15. In the output results given in Figs. 18.12 and 18.14, the red line is a reference input and the blue line is discharged mass.

Fig. 18.12 Discharged mass obtained by PID controller with fixed parameters ($b_0 = 3.0 \times 10^{-6}$)

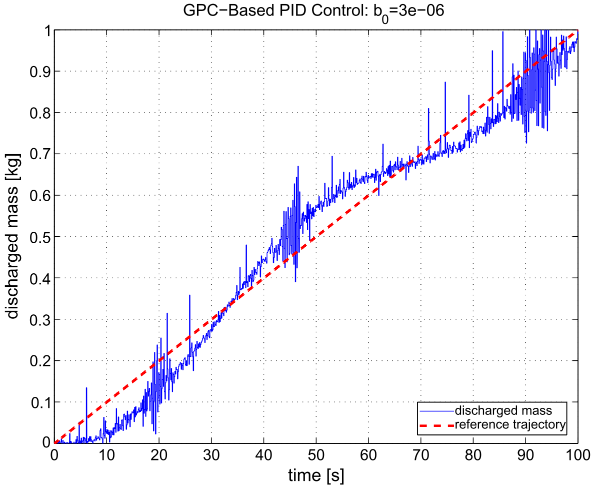


Fig. 18.13 Input voltage obtained by PID controller with fixed parameters ($b_0 = 3.0 \times 10^{-6}$)

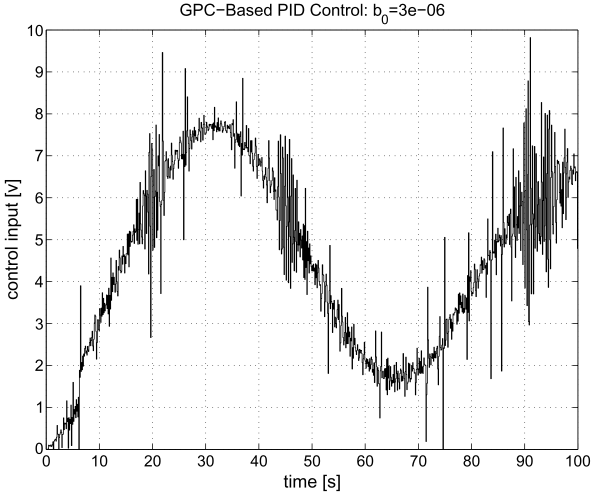


Figure 18.12 shows that transient response is slow and it takes too much time to convergence to the reference input because of small controller gain. However, it follows from Fig. 18.14 that the initial response is fast but the output signal vibrates due to too large controller gain. Furthermore, the control input is saturated because its amplitude is too large (Fig. 18.15).

Therefore, it follows from the experimental results that acceptable control performance is not obtained if parameter identification is wrong.

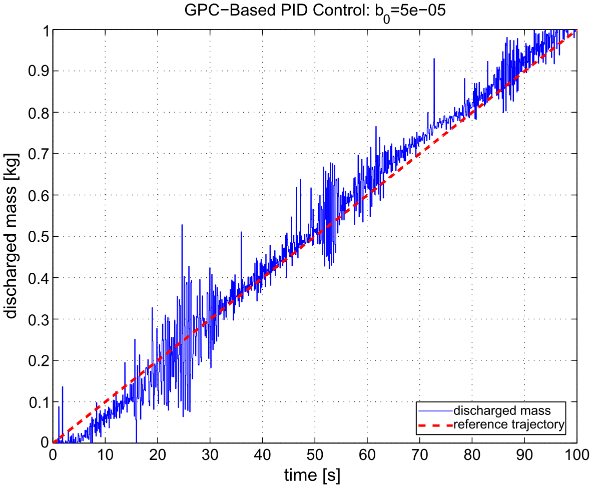


Fig. 18.14 Discharged mass obtained by PID controller with fixed parameters ($b_0 = 5.0 \times 10^{-5}$)

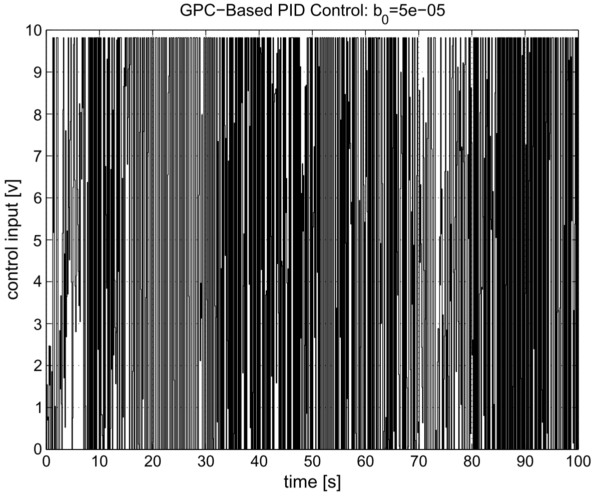


Fig. 18.15 Input voltage obtained by PID controller with fixed parameters ($b_0 = 5.0 \times 10^{-5}$)

18.3.4.2 Adaptive Control

The proposed self-tuning controller has been applied to the weigh feeder to compensate the performance deterioration caused by wrong parameter identification. The initial values of the identified parameters and the covariance matrix are set as $\hat{\alpha}_1[0] = -0.7$, $\hat{b}_0[0] = 10^{-5}$, and $\Gamma[0] = 10^2 I$, and the forgetting factor is 0.99. The other design parameters are the same as in the non-adaptive case.

Experimental results are shown in Fig. 18.16, 18.17, 18.18, 18.19 and 18.20. The output response is faster than in Figs. 18.12 and 18.14, and the variation of the output signal is suppressed more than the non-adaptive case. Hence, it can be seen that the effectiveness of the proposed method is confirmed.

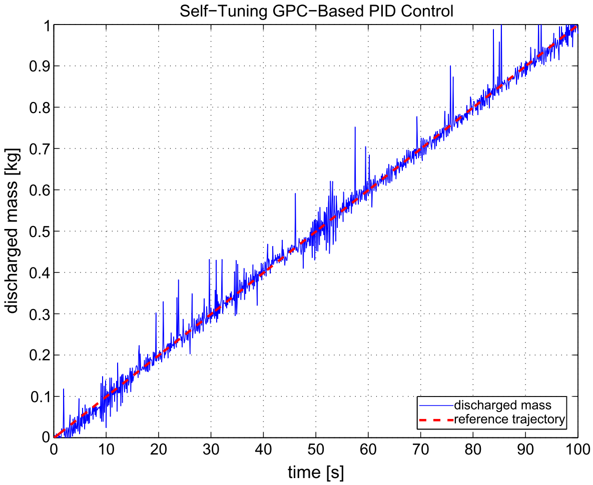


Fig. 18.16 Discharged mass obtained by self-tuning controller

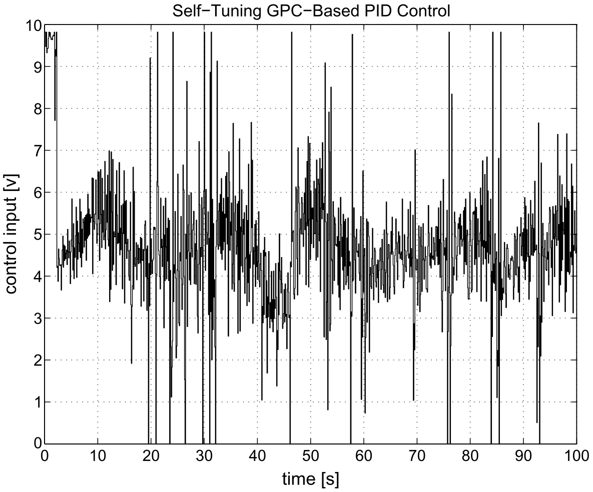


Fig. 18.17 Input voltage obtained by self-tuning controller

18.3.5 Conclusion

This study has discussed a design method for controlling a weigh feeder. Because the weigh feeder employed in industry has been controlled by a PID controller, PID parameters have been designed on the basis of GPC to achieve the concept of GPC by PID control. Experimental results have demonstrated the effectiveness of the this design method.

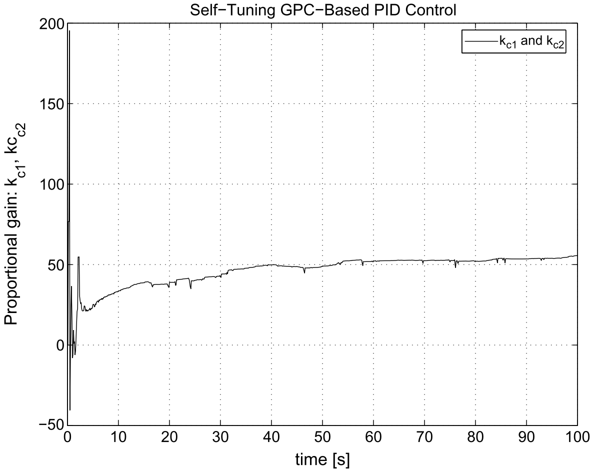


Fig. 18.18 Obtained proportional gain k_{c1}

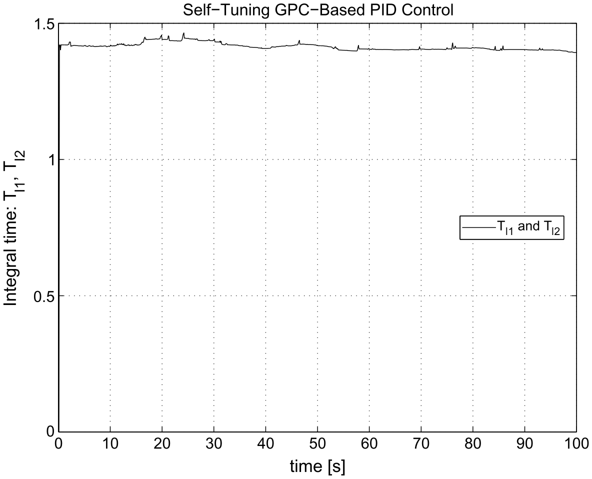


Fig. 18.19 Obtained integral time T_{I1}

18.4 Extension to Multirate System

18.4.1 Introduction

Most GPC-based PID controllers have been designed in a single-rate system such that the update interval of the control input is equivalent to the sampling interval of the plant output [3, 23, 31]. Actually, a controlled system is often a multirate system, where the update interval differs from the sampling interval due to various constraints, e.g., the performance of a sensor, the structure of a controlled plant, and so on [1, 2, 11, 18, 32, 39, 54].

In this section, a PID controller is designed for a multirate system, where the sampling interval is longer than the update interval. In particular, this section deals

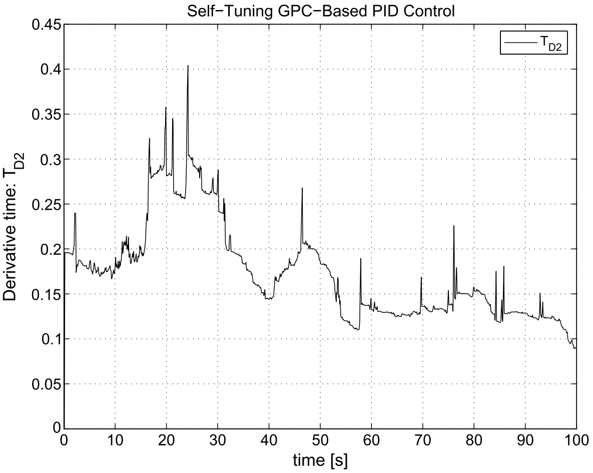


Fig. 18.20 Obtained derivative time T_{D2}

with a multirate system where the sampling interval is an integer multiple of the update interval. If the update interval is set as equivalent to the sampling interval, a controlled system is a single-rate system, and the design of the obtained system is easier than that of a multirate system. However, the control performance of the multirate system can be improved because the control input in the multirate system can be finely adjusted, although the design of the multirate system will be more complicated than that of the single-rate system.

In a multirate system, ripples might emerge between sampled outputs even if the plant output converges to a reference input at sampling instants [17, 21, 48, 49]. As is well known, the ripples can be suppressed if an integrator is employed in design of a control system. Ripple-free multirate control has been studied for a long time, but there has been no report of a ripple-free design method with PID control. Hence, the multirate PID controller is based on the multirate GPC to suppress ripples by PID control. Consequently, ripple-free multirate control can be obtained by PID control.

18.4.2 Controlled Plant and PID Controller

Consider a continuous-time system described as the following transfer function:

$$G(s) = \frac{\beta_1 s + \beta_0}{s^2 + \alpha_1 s + \alpha_0} e^{-Ls} \tag{18.78}$$

where L denotes the deadtime in continuous-time. Generally, real plants are high-order systems with deadtime, but since chemical processes can be approximated by first-order or second-order plus deadtime systems, a controlled plant is assumed to be a second-order plus deadtime system [36]. Furthermore, PID control is sufficient for processes where the dominant dynamics are of the second order [5].

A continuous-time plant is controlled using a digital controller with zeroth-order hold. Using a sampling interval T_s, the continuous-time transfer function is transformed into the following discrete-time model:

$$\begin{aligned} A\bigl(z_1^{-1}\bigr)y(k) &= B\bigl(z_1^{-1}\bigr)u(k-1), \\ A\bigl(z_1^{-1}\bigr) &= 1 + a_1 z_1^{-1} + a_2 z_1^{-2}, \\ B\bigl(z_1^{-1}\bigr) &= b_0 + b_1 z_1^{-1} + \cdots + b_m z_1^{-m} \end{aligned} \tag{18.79}$$

where $y(k)$ and $u(k)$ are the plant output and the control input, respectively; z_1^{-1} denotes the one step backward shift operator. If the deadtime in discrete time is a positive integer, the leading elements of the polynomial $B(z_1^{-1})$ are 0 [9].

Assumption 18.3 *The sampling interval of the plant output differs from the update interval of the control input, and the following constraint is assumed*:

- *Control input $u(k)$ is updated at intervals of T_s.*
- *Plant output $y(k)$ is sampled at intervals of lT_s,*

where l is an integer.

If the update interval is set to lT_s, a control system is obtained as a slow-rate single-rate system, and the simplicity of its design is superior to that of a multirate system. However, the potential of a multirate system should be employed to improve control performance. In this study, the method for designing a PID controller in a multirate system is proposed to effectively utilize the potential of a multirate system and to maintain the simplicity of a PID control system.

A multirate PID controller is given as:

$$\begin{aligned} \Delta_1 \boldsymbol{u}(k) &= \begin{bmatrix} \frac{lT_s}{T_{I1}} \\ \vdots \\ \frac{lT_s}{T_{Il}} \end{bmatrix} w(k) - \begin{bmatrix} k_{c1}(\Delta_l + \frac{lT_s}{T_{I1}} + \frac{T_{D1}}{lT_s}\Delta_l^2) \\ \vdots \\ k_{cl}(\Delta_l + \frac{lT_s}{T_{Il}} + \frac{T_{Dl}}{lT_s}\Delta_l^2) \end{bmatrix} y(k) \\ &= \boldsymbol{C}(1)w(k) - \boldsymbol{C}\bigl(z_l^{-1}\bigr)y(k), \end{aligned} \tag{18.80}$$

$$\Delta_1 \boldsymbol{u}(k) = \bigl[\Delta_1 u(k) \quad \Delta_1 u(k+1) \quad \ldots \quad \Delta_1 u(k+l-1)\bigr]^{\mathrm{T}}, \tag{18.81}$$

$$\begin{aligned} &\boldsymbol{C}\bigl(z_l^{-1}\bigr) = \bigl[C_1\bigl(z_l^{-1}\bigr) \quad \ldots \quad C_l\bigl(z_l^{-1}\bigr)\bigr]^{\mathrm{T}}, \\ &\quad C_i\bigl(z_l^{-1}\bigr) = k_{ci}\left(\Delta_l + \frac{lT_s}{T_{Ii}} + \frac{T_{Di}}{lT_s}\Delta_l^2\right) \quad (i = 1, \ldots, l), \\ &\quad z_j^{-1} = z_1^{-j}, \\ &\quad \Delta_j = 1 - z_j^{-1} \end{aligned} \tag{18.82}$$

where $w(k)$ is the reference input, k_{ci}, T_{Ii}, and T_{Di} are the proportional gain, the integral time, and the derivative time, respectively; z_j^{-1} denotes the j steps backward shift operator. In this study, the PID parameters of the multirate PID controller (18.80) are designed to prevent ripples between sampled outputs.

18.4.3 Generalized Predictive Control in Multirate System

In this study, a control law of GPC is derived to design the PID parameters in a multirate system.

First, a discrete-time state space model of a controlled plant is obtained [32, 38]. Equation (18.79) is represented as the following state space model:

$$\boldsymbol{x}(k+1) = A\boldsymbol{x}(k) + \boldsymbol{b}u(k), \tag{18.83}$$

$$y(k) = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}(k) \tag{18.84}$$

where $\boldsymbol{x}(k)$ is a state vector, A is a matrix, and $\boldsymbol{b}$ and $\boldsymbol{c}$ are vectors. Using an integrator, the state-space model (18.83) and (18.84) is extended as:

$$\bar{\boldsymbol{x}}(k+1) = \bar{A}\bar{\boldsymbol{x}}(k) + \bar{\boldsymbol{b}}\Delta_1 u(k), \tag{18.85}$$

$$y(k) = \bar{\boldsymbol{c}}^{\mathrm{T}}\bar{\boldsymbol{x}}(k) \tag{18.86}$$

where

$$\bar{\boldsymbol{x}}(k) = \begin{bmatrix} \boldsymbol{x}(k) \\ x_I(k) \end{bmatrix}, \qquad \bar{A} = \begin{bmatrix} A & \boldsymbol{b} \\ 0 & 1 \end{bmatrix},$$

$$\bar{\boldsymbol{b}} = \begin{bmatrix} \boldsymbol{b} \\ 1 \end{bmatrix},$$

$$\bar{\boldsymbol{c}}^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{c}^{\mathrm{T}} & 0 \end{bmatrix},$$

$$x_I(k+1) = x_I(k) + \Delta_1 u(k),$$

$$u(k) = x_I(k) + \Delta_1 u(k).$$

Using a technique such as lifting [7, 48], the extended model is transformed into the following l-inputs single-output single-rate system:

$$\bar{\boldsymbol{x}}(k+l) = A_l\bar{\boldsymbol{x}}(k) + B_l\Delta_1\boldsymbol{u}(k), \tag{18.87}$$

$$y(k) = \bar{\boldsymbol{c}}^{\mathrm{T}}\bar{\boldsymbol{x}}(k), \tag{18.88}$$

$$A_l = \bar{A}^l.$$

$$B_l = \begin{bmatrix} \bar{A}^{l-1}\bar{\boldsymbol{b}} & \bar{A}^{l-2}\bar{\boldsymbol{b}} & \ldots & \bar{\boldsymbol{b}} \end{bmatrix}.$$

The use of the l steps backward shift operator z_l^{-1} expresses the lifted system as the following input–output representation:

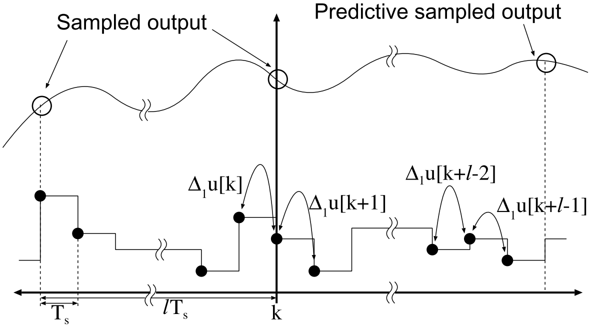


Fig. 18.21 GPC for evaluating the variation of the control input in a multirate system

$$\begin{aligned} A\left(z_l^{-1}\right)\Delta_l y(k) &= \boldsymbol{B}\left(z_l^{-1}\right)\Delta_1 \boldsymbol{u}(k-l), \\ A\left(z_l^{-1}\right) &= 1 + a_1 z_l^{-1} + a_2 z_l^{-2}, \\ \boldsymbol{B}\left(z_l^{-1}\right) &= \left[B_1\left(z_l^{-1}\right) \quad \dots \quad B_l\left(z_l^{-1}\right)\right], \\ B_j\left(z_l^{-1}\right) &= b_{j0} + b_{j1} z_l^{-1} + \dots + b_{jm} z_l^{-m}. \end{aligned} \tag{18.89}$$

A GPC law in a multirate system is derived by minimizing the following performance index [32, 38]:

$$J(k) = E\left[\sum_{j=N_1}^{N_2} \left\{y(k+jl) - w(k+jl)\right\}^2 + \sum_{j=0}^{N_u-1} \left\|\Delta_1 \boldsymbol{u}(k+jl)\right\|_{\Lambda}^2\right] \tag{18.90}$$

where N_1, N_2, N_u, and λ_j $(j = 1, \dots, l)$ are the minimum predictive horizon, maximum predictive horizon, control horizon, and a weighting factor for l-control inputs. Since $\Delta_1 \boldsymbol{u}(k)$ is evaluated on the right-hand side of (18.90), the variation of the control input can be attenuated (Fig. 18.21). A future reference input $w(k + jl)$ is given as:

$$\begin{aligned} &w(k) = y(k), \\ &w(k+jl) = (1-\alpha)r + \alpha w\big(k + (j-1)l\big) \\ &\quad (0 \le \alpha < 1) \end{aligned} \tag{18.91}$$

where r is a set-point and is given as a step-wise function.

Diophantine equation (18.92) is solved and polynomials $E_j(z_l^{-1})$ and $F_j(z_l^{-1})$ are obtained to derive a GPC law:

$$\begin{aligned} 1 &= \Delta_l A\left(z_l^{-1}\right)E_j\left(z_l^{-1}\right) + z_l^{-j} F_j\left(z_l^{-1}\right), \\ E_j\left(z_l^{-1}\right) &= 1 + e_{j,1} z_l^{-1} + \dots + e_{j,j-1} z_l^{-(j-1)}, \\ F_j\left(z_l^{-1}\right) &= \bar{f}_{j,0} + \bar{f}_{j,1} z_l^{-1} + \bar{f}_{j,2} z_l^{-2}. \end{aligned} \tag{18.92}$$

Next, using $E_j(z_l^{-1})$, polynomials $G_j(z_1^{-1})$ and $H_j(z_1^{-1})$ are derived from the following equations:

$$\begin{aligned}
&M_j\left(z_1^{-1}\right) = G_j\left(z_1^{-1}\right) + z^{-(jl-1)} H_j\left(z_1^{-1}\right),\\
&G_j\left(z_1^{-1}\right) = g_0 + g_1 z_1^{-1} + \cdots + g_{jl-1} z_1^{-(jl-1)}
\end{aligned}$$

where $M_j(z_1^{-1})$ are defined as

$$\begin{aligned}
&M_j\left(z_1^{-1}\right) = \sum_{i=1}^{l} E_j\left(z_l^{-1}\right) z_1^{-(l-i)} B_i\left(z_l^{-1}\right) L\left(z_1^{-1}\right),\\
&L\left(z_1^{-1}\right) = 1 + z_1^{-1} + \cdots + z_1^{-(l-1)}.
\end{aligned}$$

Then, a GPC law is obtained as

$$\boldsymbol{H}\left(z_1^{-1}\right)\Delta_1 \boldsymbol{u}(k) = \boldsymbol{P}\left(z_l^{-1}\right) w(k+N_2 l) - \boldsymbol{F}\left(z_l^{-1}\right) y(k) \tag{18.93}$$

where l-dimensional polynomial vectors $\boldsymbol{P}(z_l^{-1})$ and $\boldsymbol{F}(z_l^{-1})$ and l-dimensional polynomial matrix $\boldsymbol{H}(z_1^{-1})$ are given as

$$\left[\boldsymbol{p}_{N_1} \quad \boldsymbol{p}_{N_1+1} \quad \cdots \quad \boldsymbol{p}_{N_2}\right] = [I_{l\times l} \quad \boldsymbol{0}_{l\times l(N_u-1)}]\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} + \bar{\Lambda}\right)^{-1}\boldsymbol{G}^{\mathrm{T}},$$

$$\boldsymbol{P}\left(z_l^{-1}\right) = \boldsymbol{p}_{N_2} + \boldsymbol{p}_{N_2-1} z_l^{-1} + \cdots + \boldsymbol{p}_{N_1} z_l^{-(N_2-N_1)}, \tag{18.94}$$

$$\boldsymbol{F}\left(z_l^{-1}\right) = \boldsymbol{p}_{N_1} F_{N_1}\left(z_l^{-1}\right) + \cdots + \boldsymbol{p}_{N_2} F_{N_2}\left(z_l^{-1}\right), \tag{18.95}$$

$$\boldsymbol{H}\left(z_1^{-1}\right) = I_{l\times l} + z_1^{-1}\,\mathrm{diag}\left\{\boldsymbol{p}_{N_1} H_{N_1}\left(z_1^{-1}\right) + \cdots + \boldsymbol{p}_{N_2} H_{N_2}\left(z_1^{-1}\right)\right\}, \tag{18.96}$$

$$\text{Element } (i,j) \text{ in } \boldsymbol{G} = \begin{cases} \boldsymbol{g}_{(N_1-1+i)l-j} & (N_1-1+i-j \geq 0),\\ \boldsymbol{0}_{1\times l} & (N_1-1+i-j < 0), \end{cases}$$

$$\bar{\Lambda} = \text{block diag}\{\Lambda, \ldots, \Lambda\}.$$

18.4.4 Design of PID Parameters

The PID parameters of the PID controller (18.80) are designed on the basis of the derived GPC law (18.93). Because a reference input is given as future information in the design of GPC, a PID compensator composed of present and past information cannot deal with the reference input. Hence, this study proposes a method for designing the PID parameters using a future reference input in a multirate system.

The reference input $w(k+jl)$ given in (18.91) is rearranged as:

$$w(k+jl) = \alpha^j y(k) + \left(1-\alpha^j\right) r. \tag{18.97}$$

It follows from (18.97) that the reference input of a multirate GPC is composed of the present sampled output $y(k)$ and the set-point r. Based on (18.94) and (18.97), the following equations are calculated:

$$\boldsymbol{P}\left(z_l^{-1}\right)w(k+N_2l) = \boldsymbol{p}_r r + \boldsymbol{p}_y y(k), \tag{18.98}$$

$$\boldsymbol{p}_r = \sum_{j=N_1}^{N_2}\left(1-\alpha^j\right)\boldsymbol{p}_j, \tag{18.99}$$

$$\boldsymbol{p}_y = \sum_{j=N_1}^{N_2}\alpha^j \boldsymbol{p}_j. \tag{18.100}$$

Then, the multirate GPC law (18.93) is rewritten as

$$\begin{aligned}
&\boldsymbol{H}\left(z_1^{-1}\right)\Delta_1\boldsymbol{u}(k) = \boldsymbol{p}_r r - \boldsymbol{F}_p\left(z_l^{-1}\right)y(k),\\
&\boldsymbol{F}_p\left(z_l^{-1}\right) = \boldsymbol{F}\left(z_l^{-1}\right) - \boldsymbol{p}_y,\\
&\boldsymbol{F}\left(z_l^{-1}\right) = \begin{bmatrix} f_{1,0}+f_{1,1}z_l^{-1}+f_{1,2}z_l^{-2} \\ \vdots \\ f_{l,0}+f_{l,1}z_l^{-1}+f_{l,2}z_l^{-2}\end{bmatrix}, \quad \boldsymbol{p}_y = \begin{bmatrix} p_{y,1} \\ \vdots \\ p_{y,l}\end{bmatrix}
\end{aligned} \tag{18.101}$$

where the following equation is satisfied:

$$\boldsymbol{p}_r = \boldsymbol{F}_p(1).$$

A new diagonal matrix X is defined as $\boldsymbol{H}(1)$, which is the steady-state gain matrix of the polynomial matrix $\boldsymbol{H}(z_1^{-1})$:

$$X \overset{\triangle}{=} \boldsymbol{H}(1). \tag{18.102}$$

Next, the GPC law (18.101), $\boldsymbol{H}(z_1^{-1})$ which is replaced by X, is compared with the PID controller (18.80). Consequently, the PID compensator $\boldsymbol{C}(z_l^{-1})$ is designed to satisfy the following equation:

$$\boldsymbol{C}\left(z_l^{-1}\right) = X^{-1}\boldsymbol{F}_p\left(z_l^{-1}\right). \tag{18.103}$$

It follows from (18.81) and (18.103) that the PID parameters are calculated as

$$k_{ci} = -\frac{1}{x_i}(f_{i,1}+2f_{i,2}), \tag{18.104}$$

$$T_{Ii} = -\frac{f_{i,1}+2f_{i,2}}{f_{i,0}-p_{y,i}+f_{i,1}+f_{i,2}}lT_s, \tag{18.105}$$

$$T_{Di} = -\frac{f_{i,2}}{f_{i,1}+2f_{i,2}}lT_s, \quad i=1,\dots,l \tag{18.106}$$

where x_i is scalar and

$$X = \mathrm{diag}\{x_1, \dots, x_l\},$$
$$\boldsymbol{F}_p\big(z_l^{-1}\big) = \begin{bmatrix} f_{1,0} - p_{y,1} + f_{1,1}z_l^{-1} + f_{1,2}z_l^{-2} \\ \vdots \\ f_{l,0} - p_{y,l} + f_{l,1}z_l^{-1} + f_{l,2}z_l^{-2} \end{bmatrix}.$$

A multirate GPC-based PID controller is obtained from PID parameters designed using (18.104) through (18.106), and the future reference trajectory of GPC can be applied to the design of a PID controller. Furthermore, it follows from (18.104)–(18.106) that the design parameter α influences only the integral time.

18.4.5 Numerical Examples

Numerical examples are given to show the effectiveness of the proposed method. First, the relation between the design parameter α and the integral time is shown, and then the influence of α on the response of the plant output is presented. It is then shown that the proposed method is effective in suppressing ripples between sampled outputs.

Consider a plant described as the following transfer function:

$$G(s) = \frac{-2s+1}{(10s+1)(25s+1)}e^{-4s}. \tag{18.107}$$

A multirate PID controller is designed to have the plant output follow a reference input given as a step-wise function. The use of $T_s = 2$ transforms (18.107) into the following discrete-time model:

$$\big(1 - 1.74z_1^{-1} + 0.76z_1^{-2}\big)y(k) = \big(-0.0066z_1^{-2} + 0.021z_1^{-3}\big)u(k-1). \tag{18.108}$$

Assuming that the control input $u(k)$ is updated every step, but the plant output $y(k)$ is sampled every 2 steps, the system is a multirate system, and $l = 2$. Equation (18.108) is extended using an integrator, and the following two-input single-output single-rate discrete-time model is obtained using lifting to design a multirate PID controller:

$$\begin{aligned}&\big(1 - 1.5z_2^{-1} + 0.57z_2^{-2}\big)\Delta_2 y(k)\\ &\quad = \big[0.0024z_2^{-1} + 0.046z_2^{-2} - 0.0066z_2^{-1} + 0.040z_2^{-2} + 0.016z_2^{-3}\big]\Delta_1 \boldsymbol{u}(k-2).\end{aligned} \tag{18.109}$$

Table 18.3 Calculated PD parameters

	Proportional gain	Derivative time
$C_1(z_2^{-1})$	0.77	4.3
$C_2(z_2^{-1})$	0.56	4.4

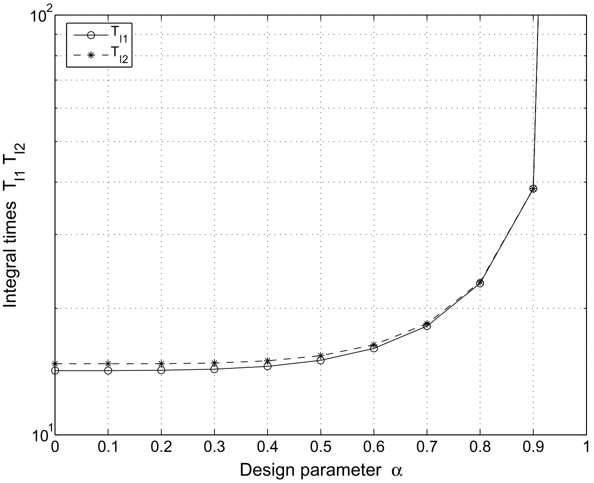


Fig. 18.22 Relationship between design parameter α and integral time

18.4.5.1 Design of Multirate PID Controller Using Future Reference Trajectory

The effectiveness of the design parameter α is shown in this example. GPC is considered first to design PID parameters. The design parameters of GPC are set as: $N_1 = 1$, $N_2 = 5$, $N_u = 1$, $\Lambda = \text{diag}\{1.5,\ 1.5\}$, and α is set from 0 to 1 in 0.1 increments. It follows from (18.91) that α must be set less than 1 in the design of GPC, but simulation has been conducted with $\alpha = 1$ to confirm the influence of α on the integral time. The proportional gain and the derivative time are designed independently of α and are calculated as shown in Table 18.3, and the integral time with respect to α is shown in Fig. 18.22. Output results obtained by the proposed multirate PID controller are shown in Fig. 18.23. This figure shows the output responses at sampling instants, and intersample behavior of the plant output is confirmed in the next example to verify the effectiveness of the design parameter α and the integral time. It follows from Fig. 18.22 that the larger the value of α, the larger the integral time. Furthermore, Fig. 18.23 shows that the larger the integral time, the slower the response of the plant output. If α is small, the change of the integral time is small. Hence, output responses, where α is $0 \sim 0.3$, are about the same. Furthermore, if $\alpha = 1$, the integral time is infinite, and Fig. 18.23 shows that the plant output could not follow the reference input.

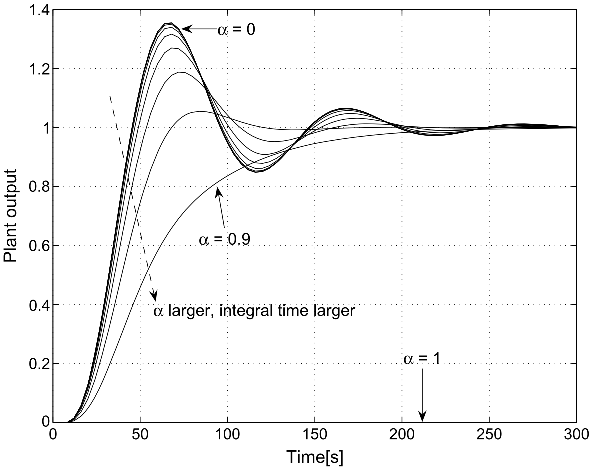


Fig. 18.23 Output results obtained using the proposed PID controller

18.4.5.2 Suppression of Ripples

The next example shows that ripples between sampled outputs can be suppressed using the proposed multirate PID controller. Simulation results are compared with two controllers to show the effectiveness of the proposed method. One is a conventional single-rate GPC-based PID controller [31], and the other is a multirate PID controller based on GPC where the variation of the control input is not evaluated. The conventional single-rate GPC-based PID controller is a special case of the proposed method where the update interval is equal to the sampling interval (i.e., 4 [s]) and $l = 1$. The latter multirate PID controller is designed using the following model, which is the lifted model of (18.108) without an integrator:

$$\begin{aligned}&\left(1 - 1.5z_2^{-1} + 0.57z_2^{-2}\right)y(k)\\&\quad = \left[0.0090z_2^{-1} + 0.016z_2^{-2} \quad -0.0066z_2^{-1} + 0.040z_2^{-2}\right]\boldsymbol{u}(k-2).\end{aligned} \tag{18.110}$$

A GPC law for the two-input single-output model (18.110) is derived by minimizing the performance index given as:

$$J_c(k) = E\left[\sum_{j=N_1}^{N_2}\left\{y(k+jl) - w(k+jl)\right\}^2 + \sum_{j=0}^{N_u-1}\left\|\Delta_l\boldsymbol{u}(k+jl)\right\|_{\Lambda}^2\right]. \tag{18.111}$$

Then, the following multirate PID controller is based on a GPC law minimizing (18.111):

$$\begin{aligned}\Delta_l\boldsymbol{u}(k) &= \boldsymbol{C}(1)w(k) - \boldsymbol{C}\left(z_l^{-1}\right)y(k),\\ \Delta_l\boldsymbol{u}(k) &= \left[\Delta_l u(k) \quad \ldots \quad \Delta_l u(k+l-1)\right].\end{aligned} \tag{18.112}$$

Because $\Delta_1\boldsymbol{u}(k)$ is not evaluated at $J_c(k)$, the variation of control input is not attenuated by the criterion (18.111) (see Fig. 18.24). The multirate PID controller (18.112)

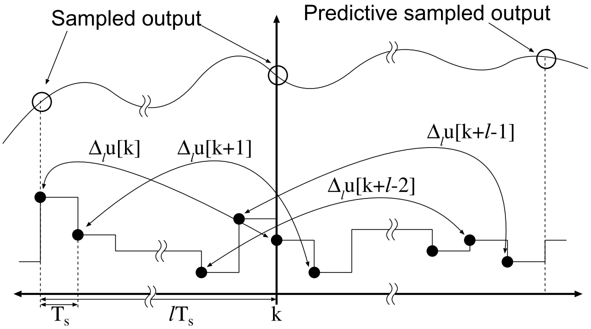

Sampled output
Predictive sampled output
$\Delta_l u[k]$
$\Delta_l u[k+1]$
$\Delta_l u[k+l\text{-}1]$
$\Delta_l u[k+l\text{-}2]$
T_s
lT_s
k

Fig. 18.24 GPC not suppressing the variation of the control input in the multirate system

Table 18.4 Calculated PID parameters of the multirate PID controller (18.112) based on multirate GPC not evaluating the variation of the control input (18.111); $l = 2$, $T_s = 2$

	Proportional gain	Integral time	Derivative time
$C_1(z_2^{-1})$	0.54	14	4.3
$C_2(z_2^{-1})$	0.35	15	4.4

Table 18.5 Calculated PID parameters of the PID controller (18.80) based on multirate GPC suppressing the variation of the control input (18.90); $l = 2$, $T_s = 2$

	Proportional gain	Integral time	Derivative time
$C_1(z_2^{-1})$	0.77	26	4.3
$C_2(z_2^{-1})$	0.56	26	4.4

Table 18.6 Calculated PID parameters of the single-rate PID controller (18.80) based on single-rate GPC (18.90); $l = 1$, $T_s = 4$

	Proportional gain	Integral time	Derivative time
$C_1(z_1^{-1})$	0.79	26	4.3

is designed on the basis of a control law minimizing the criterion. However, the proposed multirate PID controller is based on a GPC law where the variation of the control input shown in Fig. 18.21 is evaluated.

GPC is designed first to calculate PID parameters on the basis of GPC. The design parameters of GPC were set the same as in the previous example and α was fixed as 0.83. Using these design parameters, three GPC-based PID controllers were obtained. The calculated PID parameters are shown in Tables 18.4, 18.5 and 18.6.

Simulations have been conducted using the obtained three GPC-based PID controllers. In Fig. 18.25, the solid line with "○" shows the output result obtained using the multirate GPC-based PID controller (18.112) based on minimization of (18.111) (Fig. 18.24), the dashed-dotted line with "◇" shows the output result obtained using the conventional single-rate GPC-based PID controller, and the dashed line with

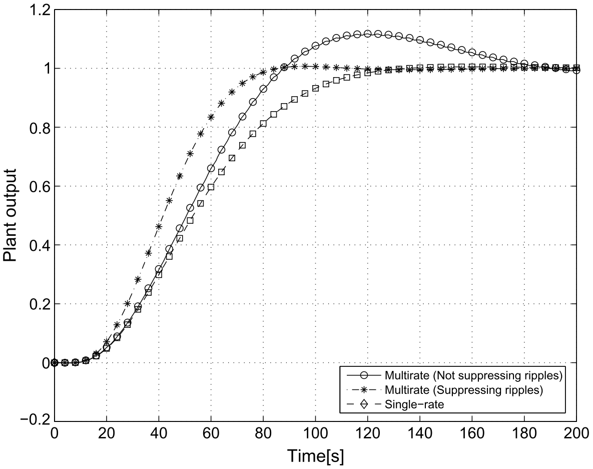


Fig. 18.25 Output results obtained using two multirate and one single-rate GPC-based PID controllers

"$*$" shows the output result obtained using the proposed multirate GPC-based PID controller. The output result obtained using the proposed multirate GPC-based PID controller converges to the reference input faster than those of the multirate GPC-based PID controller (18.112) and the single-rate GPC-based PID controller. It can be seen that every simulated output follows the reference input at the sampling instants. However, the solid line illustrated in Fig. 18.26 is the input result obtained using the multirate PID controller (18.112) and it oscillates. However, the input results obtained using the proposed multirate and the single-rate GPC-based PID controllers converge to a constant value (Fig. 18.26). Furthermore, Fig. 18.27 shows that the plant output obtained by the multirate GPC-based PID controller (18.112) oscillates between sampled outputs because the control input oscillates between sampled outputs in steady state. However, since the proposed multirate PID controller is designed on the basis of GPC where the variation of the control input is evaluated, the control input converges without oscillation. Consequently, ripples can be suppressed using the proposed multirate PID controller. The plant output obtained by the single-rate PID controller also converges to the reference input without ripples. However, the plant output obtained by the proposed multirate PID controller converges faster than that of the single-rate controller, and the rise time obtained using the proposed multirate PID controller is shorter than that of the single-rate PID controller. As mentioned above, the effectiveness of the proposed method is confirmed.

18.4.6 Conclusion

This study has discussed a method for designing a PID controller in a multirate system, where the sampling interval of the plant output is an integer multiple of the update interval of the control input. A multirate GPC law is approximated by the multirate PID controller to tune the PID parameters of a PID controller in the

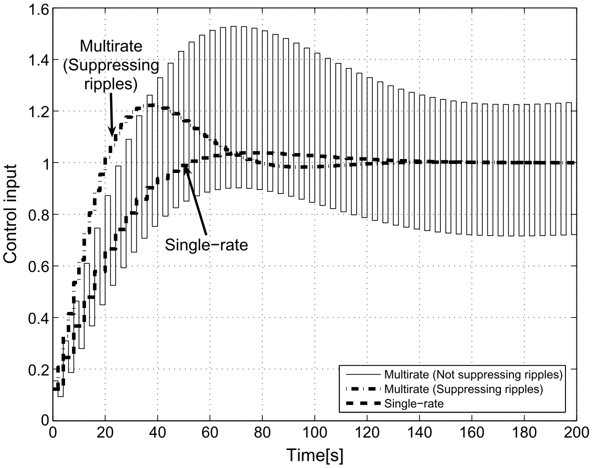


Fig. 18.26 Input results obtained using two multirate and the one single-rate GPC-based PID controllers

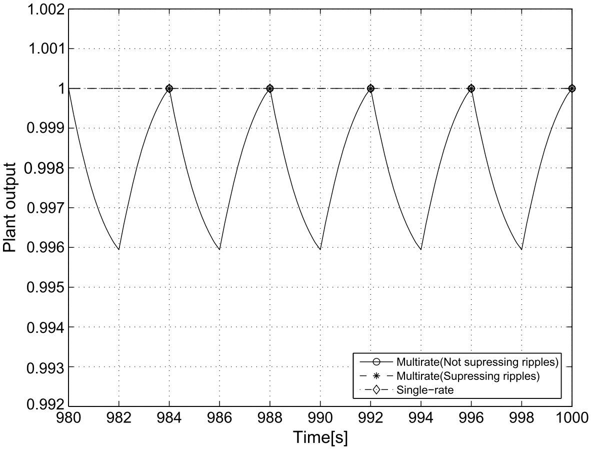


Fig. 18.27 Output results on expanded scale of Fig. 18.25

multirate system. Because GPC is designed to suppress ripples between sampled outputs, the multirate PID controller, which can suppress the ripples, is obtained. Further, this design method has given PID parameters using the design parameter of GPC, α, which adjusts an output response, and it has been shown that the design parameter redesigns the integral time independently of the proportional gain and the derivative time. If $l = 1$, the proposed multirate PID controller is equivalent to a conventional single-rate PID controller. Therefore, the proposed method includes conventional single-rate design methods. In the last section, numerical examples have demonstrated the effectiveness of the proposed method.

A multirate GPC-based PID controller has been designed in a single-input single-output case in this study. Hence, a future work is to extend the proposed design into a multivariate case. It is necessary and useful to design a multirate PID controller on the basis of H_∞ criterion and to compare it with the proposed method, and further, this design method may be applied to another multirate system in which the sampling interval of the plant output is shorter than the update interval of the control input.

18.5 Conclusions and Further Perspectives

In this chapter, to attain GPC performance by PID control, PID control laws are based on GPC laws.[2] In the case that a GPC law is approximated by a PID control law, high-order polynomials cannot be easily replaced with a PID compensator.

In Sect. 18.2, a high-order polynomial is approximated by a time-varying proportional gain, and a GPC law is extended to a strongly stable system to obtain stable a time-varying proportional gain. However, the extended control law is not always stabilized if a plant model is unknown. In this case, the proposed method should be modified because the time-varying proportional gain diverges. Next, a GPC-based PID control system was applied to the weigh feeder. In Sect. 18.3, to make the plant output follow the ramp-type reference input, the ramp-type reference trajectory is rearranged, and it can be rewritten using the present plant output. Finally, to extend the PID control system into a multirate system, a multirate PID control law was designed on the basis of a multirate GPC law in Sect. 18.4. In this section, the sampling interval of the plant output is an integer multiple of the update interval of the control input, but this design method can be applied to other cases, e.g., the update interval is longer than the sampling interval, multivariable system, and so on.

Acknowledgements The author would like to express his sincere gratitude to Professor Akira Inoue and Dr. Shiro Masuda for their helpful suggestions and valuable discussions.

The author is grateful to Professor Toru Yamamoto and Professor Sirish L. Shah for their detailed comments and suggestions.

The author would like to thank Professor Koichi Kameoka and Yamato Scale Co. Ltd. for providing the experimental setup.

References

1. Albertos, P.: Block multirate input-output model for sampled-data control systems. IEEE Trans. Autom. Control **35**(9), 1085–1088 (1990)
2. Araki, M., Yamamoto, K.: Multivariable multirate sampled-data systems: state-space description, transfer characteristics, and Nyquist criterion. IEEE Trans. Autom. Control **AC-31**(2), 145–154 (1986)
3. Asano, M., Yamamoto, T.: A design of self-tuning predictive PID controllers. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E84-A**(7), 1779–1783 (2001)
4. Asano, M., Yamamoto, T., Oki, T., Kaneda, M.: A design of neural-net based predictive PID controllers. In: Proc. of IEEE Conference on Systems, Man and Cybernetics, pp. 1113–1118 (1999)
5. Åström, K.J., Hägglund, T.: PID Controllers: Theory, Design and Tuning, 2nd edn. Instrument Society of America, Research Triangle Park (1995)

[2] These studies have been published in [28–30].

6. Camacho, E.F., Bordons, C.: Model Predictive Control, 2nd edn. Springer, Berlin (2000)
7. Chen, T., Francis, B.: Optimal Sampled-data Control Systems. Springer, Berlin (1995)
8. Clarke, D.W.: Self-tuning control of nonminimum-phase systems. Automatica **20**(5), 501–517 (1984)
9. Clarke, D.W., Mohtadi, C., Tuffs, P.S.: Generalized predictive control—part I and II. Automatica **23**(2), 137–160 (1987)
10. Deng, M., Inoue, A., Ishibashi, N., Yanou, A.: Application of an anti-windup multivariable continuous-time generalised predictive control to a temperature control of an aluminium plate. Int. J. Model. Identif. Control **2**(2), 130–137 (2007)
11. Ding, F.: Generalized yule-walker and two-stage identification algorithms for dual-rate systems. J. Control Theory Appl. **4**, 338–342 (2006)
12. Goodwin, G.C., Sin, K.S.: Adaptive Filtering Prediction and Control. Prentice-Hall, New York (1984)
13. Haefner, H.: Advanced weigh feeder system for optimization of the kiln system. World Cem. **27**(6), 19–27 (1996)
14. Heinrici, H.: Continuous weighing and feeding systems—an integral component in process engineering. Aufbereit.-Tech. **41**(8), 376–383 (2000) (in German)
15. Hopkins, M.: Loss in weight feeder systems. Meas. Control **39**(8), 237–240 (2006)
16. Inoue, A., Yanou, A., Hirashima, Y.: A design of a strongly stable generalized predictive control using coprime factorization approach. In: Proc. of the American Control Conference, San Diego, pp. 652–656 (1999)
17. Ishitobi, M., Kawanaka, M., Nishi, H.: Ripple-suppressed multirate adaptive control. In: Proc. of 15th IFAC World Congress, pp. 327–332 (2002)
18. Izadi, I., Zhao, Q., Chen, T.: An h_∞ approach to fast rate fault detection for multirate sampled-data systems. J. Process Control **16**, 651–658 (2006)
19. Johnson, M.A., Moradi, M.H. (eds.): PID Control—New Identification and Design Methods. Springer, London (2005)
20. Kwok, K., Shah, S.: Long-range predictive control with steady state error weighting. Int. J. Adapt. Control Signal Process. **11**, 201–215 (1997)
21. Liang, S., Ishitobi, M.: Properties of zeros of discretised system using multirate input and hold. IEE Proc., Control Theory Appl. **151**(2), 180–184 (2004)
22. Maciejowski, J.M.: Predictive Control with Constraints. Prentice Hall, New York (2002)
23. Miller, R.M., Shah, S.L., Wood, R.K., Kwok, E.K.: Predictive PID. ISA Trans. **38**, 11–23 (1999)
24. Moradi, M.H., Johnson, M.A., Katebi, M.R.: Predictive PID control. In: Johnson, M.A., Moradi, M.H. (eds.) PID Control—New Identification and Design Methods, pp. 473–524. Springer, Berlin (2005)
25. O'Dwyer, A.: Hand Book of PI and PID Controller Tuning Rules. Imperial College Press, Singapore (2003)
26. Omatu, S., Yamamoto, T. (eds.): Self-tuning Control. The Society of Instrument and Control Engineers (1996) (in Japanese)
27. Park, B.G., Kwon, W.H., Park, S.H.: Discrete two degrees of freedom PID controller based on the GPC. In: SICE '95, Sapporo, pp. 1251–1255 (1995)
28. Sato, T.: Design method of PID controller in multirate system. Dyn. Contin. Discrete Impuls. Syst. **16**(1), 123–138 (2009)
29. Sato, T.: Strongly stable GPC-based PID controller. Int. J. Adv. Mechatron. Syst. **1**(3), 183–193 (2009)
30. Sato, T.: Design of a GPC-based PID controller for controlling a weigh feeder. Control Eng. Pract. **18**(2), 105–113 (2010)
31. Sato, T., Inoue, A.: Improvement of tracking performance in self-tuning PID controller based on generalized predictive control. Int. J. Innov. Comput. Inf. Control **2**(3), 491–503 (2006)
32. Sato, T., Inoue, A.: Generalized predictive control in fast-rate single-rate and dual-rate systems. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E90-A**(11), 2616–2619 (2007)
33. Sato, T., Kameoka, K.: Self-tuning control of a weigh feeder. Trans. Soc. Instrum. Control Eng. **43**(6), 522–524 (2007) (in Japanese)

34. Sato, T., Inoue, A., Yamamoto, T., Shah, S.L.: Self-tuning PID controllers based on the strongly stable generalized minimum variance control law. In: IFAC Workshop on Digital Control: Past, Present and Future of PID Control, Terrassa, pp. 511–516 (2000)
35. Sato, T., Inoue, A., Yamamoto, T.: Improvement of tracking performance in designing a GPC-based PID controller using a time-varying proportional gain. IEEJ Trans. Electr. Electron. Eng. **1**(4), 438–441 (2006)
36. Sato, T., Inoue, A., Yamamoto, T.: Gpc-based PID controller using a stable time-varying proportional gain. In: Proc. of the 2007 IEEE Int. Conf. on Networking, Sensing and Control, pp. 536–541 (2007)
37. Sato, T., Inoue, A., Yamamoto, T.: Two-degree-of-freedom PID controller based on extended generalized minimum variance control. Int. J. Innov. Comput. Inf. Control **4**(12), 3111–3122 (2008)
38. Scattolini, R.: Multi-rate self-tuning predictive controller for multi-variable systems. Int. J. Syst. Sci. **23**(8), 1347–1359 (1992)
39. Sheng, J., Chen, T., Shah, S.L.: Multirate generalized predictive control for sampled-data systems. Dyn. Contin. Discrete Impuls. Syst., Ser. B, Appl. Algorithms **8**(4), 485–499 (2001)
40. Silva, G.J., Datta, A., Bhattacharyya, S.P.: PID Controllers for Time-Delay Systems. Birkhäuser, Boston (2005)
41. Smith, O.J.: Closer control of loops with dead-time. Chem. Eng. Prog. **53**, 217–219 (1957)
42. Smith, O.J.: A controller to overcome dead-time. ISA J. **6**, 28–33 (1959)
43. Sunan, H., King, T.K., Hen, L.T.: Applied Predictive Control. Springer, London (2002)
44. Tan, K.K., Huang, S.N., Lee, T.H.: Development of a GPC-based PID controller for unstable systems with deadtime. ISA Trans. **39**, 57–70 (2000)
45. Tan, K.K., Lee, T.H., Leu, F.M.: Predictive PI versus smith control for dead-time compensation. ISA Trans. **40**, 17–29 (2001)
46. Tan, K.K., Lee, T.H., Huang, S.N., Leu, F.M.: PID control design based on a GPC approach. Ind. Eng. Chem. Res. **41**(8), 2013–2022 (2002)
47. Tan, K.K., Lee, T.H., Leu, F.M.: Optimal smith-predictor design based on a GPC approach. Ind. Eng. Chem. Res. **41**(5), 1242–1248 (2002)
48. Tangirala, A.K., Li, D., Patwardhan, R., Shah, S.L., Chen, T.: Issues in multirate process control. In: Proc. of the American Control Conference, pp. 2771–2775 (1999)
49. Tangirala, A.T., Li, D., Patwardhan, R.S., Shah, S.L., Chen, T.: Ripple-free conditions for lifted multirate control systems. Automatica **37**(10), 1637–1645 (2001)
50. Usui, K.: Continuous flow rate control system with fuzzy controller. In: International Symposium on Measurement of Force and Mass, pp. 79–84 (1992)
51. Vermylen, J.: Controllers assure consistency in batch mixing of pasta dough. Food Process. **46**(10), 138–139 (1985)
52. Vidyasagar, M.: A Factorization Approach. Control System Synthesis. The MIT Press, Cambridge (1985)
53. Visioli, A.: Practical PID Control. Springer, London (2006)
54. Wang, X., Huang, B., Chen, T.: Multirate minimum variance control design and control performance assessment: A data-driven subspace approach. IEEE Trans. Control Syst. Technol. **15**(1), 65–74 (2007)
55. Yamamoto, T., Kaneda, M.: A design of self-tuning PID controllers based on the generalized minimum variance control law. Trans. Inst. Syst. Control Inf. Eng. **11**(1), 1–9 (1998) (in Japanese)
56. Yamamoto, T., Inoue, A., Shah, S.L.: Generalized minimum variance self-tuning pole-assignment controller with a PID structure. In: Proc. of 1999 IEEE Int Conf. on Control Applications, Hawaii, pp. 125–130 (1999)
57. Yanou, A., Inoue, A.: An extension of multivariable continuous-time generalized predictive control by using coprime factorization approach. In: Proc. of SICE Annual Conference, Fukui, pp. 3018–3022 (2003)
58. Yu, C.C.: Autotuning of PID Controllers, 2nd edn. Springer, London (2006)

Index

A
Adaptive tuning, 425
AMIGO method, 123
Analytical robust tuning, 130
Arbitrary order, 466
Auto-tuning, 486

B
Backlash, 438
Benchmark index, 410
Block diagram, 472, 473
Bode's ideal loop transfer function, 476
Bounded-input bounded-output, 471

C
Caputo's definition, 467
Cascade control, 250
Cauchy's argument principle, 472
Cauchy's formula, 466, 467
Characteristic equation, 471
Characteristic polynomial, 472
'Classical' PID controller, 5
Commensurate order, 469
Commensurate-order system, 469–471
Communication failures, 449
Complementary sensitivity function, 484
Complex Hurwitz signature lemma, 326
Complex plane, 470, 472–474
Constant phase margin, 476
Controllability index, 51
Controller design for unstable systems, 88
Controller performance index, 388, 391, 396
Controller structures, 50
Convex stability set, 328
Coprime factor uncertainty, 300
Crossover information, 288

D
Damped oscillation, 471
Data-driven PID controller, 528
Dead-time
  compensation, 241
  compensators, 432
  dominant, 423
Decoupling, 255
Degraded performance, 136
Derivative action, 328, 465, 472, 474, 475
Derivative filter, 59
Design based on equating coefficient method, 97
Design specification, 480, 484, 487
Design trade-offs, 272
Desired closed-loop, 83
Desired closed-loop response, 131
Direct PID synthesis, 331
Direct synthesis, 10, 82, 83, 104, 132, 147

E
Enhanced PID, 431, 440
Event detector, 499
Event-based control, 497, 499
Event-based sampling, 498
External reset signal, 429

F
F-Migo, 482
Feasibility of PID control, 284
Feedforward
  action, 209
  compensators, 207
  tuning, 225
Filter-bank, 308
Fine-tuning, 356
First-order plus delay model parameters, 152

R. Vilanova, A. Visioli (eds.), *PID Control in the Third Millennium*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2425-2, © Springer-Verlag London Limited 2012

Flatness of the phase curve, 487
Fractional
  calculus, 466, 467, 476
  fractional-order derivative, 467
  fractional-order integral, 467
  fractional-order lead compensator, 486
  fractional-order PID, 477, 479
  fractional-order systems, 469
Fragile, 355
Fragility, 68
  analysis, 355
  balance, 359
  evaluation, 359
  index, 351
  robustness-fragility rings, 366
Frequency loop-shaping, 289
Frequency response measurements, 319

G
Gain and phase margin, 181, 184, 258
Gain crossover frequency, 476, 484, 486
Gain margin, 117
General robustness margin, 122
Generalized minimum variance control, 568
Generalized predictive control, 553
GPC-based PID, 553, 569
Grünwald–Letnikov's definition, 468

H
$\mathcal{H}_\infty$/FB PID, 310
$\mathcal{H}_\infty$/FB estimation, 311
Half rule, 154
Hammerstein models, 535
High-frequency noise, 465, 475, 483

I
Ideal cutoff characteristic, 476
'Ideal' PID controller, 5
Ideal transfer function, 53
Initial condition, 469, 479, 485
Input signal design, 297
Input usage, 164
Integer-order, 469, 472
Integral action, 465, 472–474
Integral of absolute error, 89
Integrating and unstable processes, 79
Integrating process, 75, 79, 82, 171, 423
Internal Model Control (IMC), 481
  design based on IMC, 94
  IMC controller, 54, 259
  IMC framework, 127
  IMC methodology, 82
  IMC principle, 53, 80
  IMC-based tuning, 54
Internal stability, 179
Inverted pendulum, 292
Irrational-order, 469, 471
Iso-damping property, 484

J
Jury, 472

K
Kappa–Tau (KT) method, 122
Kharitonov's method, 99
KLκ150 method, 123

L
Lead-lag compensator, 477
Limit cycles, 512
Linear inequality, 328
Linear programming, 330
Linear time invariant, 468, 472
Load disturbance, 480, 483
Load disturbance rejection, 220
Loop dead-time, 418, 421
Loop performance, 419

M
Majhi and Atherton's Smith predictor, 243
Maximum sensitivity, 134
Maximum sensitivity based designs, 122
MBAT
  procedure, 48
  robustness in MBAT, 64
  rule, 48
McMillan degree, 178
Memory, 469
MIMO gain margin, 189
Minimising an appropriate performance criterion, 8
Model matching, 130
Model structure, 49, 68
Modified Smith predictor, 244, 245
Multirate GPC-based PID, 587
Multirate system, 580
Multivariable PID, 256

N
Neural net-based PID, 539
Non-fragile, 355
Noncausal feedforward action, 212, 217
Nonlinear constraint, 485
Nonlinear feedforward, 214
Normalised delay, 55
Normalized dead-time, 115
Nyquist plot, 476

O
Operational calculus, 467
Optimization, 484
Output disturbance rejection, 484

P
Parameter estimation, 300
Parameter separation, 327
Pareto-optimality, 165
Perfect feedforward, 222
Performance
  assessment, 384
  benchmarking, 384
  fragility, 355, 358
  indices, 82
Performance/robustness tradeoff, 134, 135, 373
Phase and gain margin, 481
Phase margin, 118, 476, 483, 484, 487
PI_{2Ms} method, 124
PI/PID controller structures, 5, 429
PID stabilizing set, 327
PID synthesis for performance, 343
Principal Riemann sheet, 471
Process reaction curve, 7
Proportional action, 465, 472, 473, 475

R
Rational-order, 469
Reference model, 60
Regulatory control, 381
Relative stability, 465, 472
Relay feedback, 263
Relay test, 486
Resilient, 355
Restricted structure optimal control, 395
Riemann surface, 470
Riemann–Liouville's definition, 467
Rise time, 472
Robust multivariable PID, 260
Robust performance, 136
Robust PID control, 114
Robustness, 12, 86, 128
  criterion, 487
  fragility, 354, 357
  measure, 117, 121, 133
Routh, 472

S
Secondary sheet, 470
Sensitivity crossover, 292
Sensitivity function, 119, 483, 484
Series PID, 148
Set-point following, 209
Set-point weighting, 86, 90, 101
Set-point weights, 116
Setpoint feedforward, 434
Setpoint weight factor, 429
Settling time, 472, 474
Ship position controller tuning, 270
Signature, 320, 323
SIMC
  procedure, 147
  tuning, 148
Smith predictor, 103, 241
Smooth control, 163
Split range, 445, 447
Stability
  stability condition, 471, 472
Stabilizing PID controllers, 327
Stabilizing set, 327, 336
Steady-state error, 484
Step response experiment, 149
Stick-slip, 437
Sticking, 511
Symbolic method, 467

T
Tight control, 161
Time delay, 3
Time domain performance, 235
Time-triggered event detector, 509
TITO processes, 181
Tracking control, 381
Transient response, 472, 476
Tuning for multiple models, 303
Tuning method, 480
Tuning rules, 7, 8, 10, 12
Two-Degrees-of-Freedom (2DOF), 62, 81, 132, 239, 250, 349
  2DOF control, 103
  2DOF controller, 6, 80
  2DOF event-based controller, 518

U
Ultimate cycle, 12
Uncertainty estimate, 298, 299
Unstable processes, 246

Y
Youla parametrization, 560

Z
Zero, 478