Studies on PID controller tuning and self-optimizing control

Hu, Wuhua

2012

Hu, W. (2012). Studies on PID controller tuning and self-optimizing control. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/48149>

<https://doi.org/10.32657/10356/48149>

STUDIES ON PID CONTROLLER TUNING AND SELF-OPTIMIZING CONTROL

WUHUA HU

WUHUA HU

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2012

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other university or institution.

Date

Signature

Acknowledgements

First of all I thank my supervisor Dr. Gaoxi (Kevin) Xiao for his patient guidance, persistent support and encouragement, and endless trust in me during my four years' work for this thesis. It is the frequent discussions with him and the strong support from him that helped me overcome the tons of difficulties, step over the boundaries of academic fields, and keep creating progress. The thesis would be impossible to be finished in four years' time without Kevin's guidance and help.

I am grateful to Dr. Vinay Kumar Kariwala from the School of Chemical and Biomedical Engineering, Nanyang Technological University (NTU). His heartfelt guidance and great collaboration contribute a lot to the results on self-optimizing control in the thesis. The thesis can never be completed as it is now without his contributions. I am lucky to have known and worked with him since the initial of April 2010. Talking to him is always helpful, from which I learned a lot beyond the insights into the research problems. His passion and vision in his specialized field is also impressive which always stimulate me to be unsatisfied and make new progress. To me, Vinay acts as a co-supervisor, more than a collaborator. I am in debt to him.

I thank Miss Lia Maisarah Umar for cooperating in deriving the results in Chapter 8. Without her contributions, the thesis would be incomplete.

I appreciate Dr. Wen-Jian Cai, from the Division of Control and Instrumentation, NTU, for the short supervision from November 2009 to April 2010, which has significantly influenced my research work later on. It is the work with him that helped promote my taste of doing interesting research and my ability in writing good academic papers. I cannot imagine the status of the thesis if the experience of working with Dr. Cai was missing.

I am also thankful to Prof. Lihua Xie from the Division of Control and Instrumentation, NTU, for treating me as a regular member of his Sensor Network Lab and for endowing me equal chances of doing presentations at the group meetings. Indeed it has been my greatest pleasures to participate in the weekly meetings and to contact the lab members from whom I have learned a lot on academics and others.

Special thanks go to the members of Sensor Network Lab, Keyou You, Nan Xiao, Shuai Liu, Jun Xu, Jingwen Hu, Wei Meng, Tingting Gao, etc., who have always been ready to discuss and help me on research problems. Their friendliness and helps contributed much to the pleasures and achievements of my four years' study in NTU.

Warm thanks also go to my office friends, Yongxu Hu, Qian Li, Jiliang Zhang, Mingyang Zhang, etc., from my previous office of Communication Lab III, and Dawei Wang, Xiaojun Yu, Yihui Li, etc., from my current office of Network Technology Research Center. They had made it possible to have comfortable working environments and also enrich my living in Singapore by means of joyful excursions, exercises and parties.

I would like to express my biggest thanks to my wife for her constant love and support. She married to me last year, after being in a relationship with me just for one month when I was still recovering from a very miserable emotional hurt. It is her deep faith and love in me that make our marriage possible and romantically sweet. I am so lucky and happy to have her around since August 16, 2010, the day we married! It is the happiness and the support from her that have made my work in the last year be fruitful. I am heavily in debt to my wife, for the limited time I have spent with and for her since our marriage.

The last but not the least, I thank my father, mother and elder brother for their solid and persistent support, to whatever situation I was subjected. I am proud of having such a good family who are always willing to help and encourage me. Their trust and love have always been reliable resources that drive me to dissolve the challenges and create a nicer future!

Abstract

This thesis consists of two parts. The first part is devoted to analytically deriving proportional-integral-derivative (PID) tuning rules with different tuning methods and the second part is devoted to reporting some new results on self-optimizing control (SOC). The two parts are connected through the controlled variables (CVs) used in control.

Firstly the problem of tuning PID controllers for integral plus time delay (IPTD) processes with specified gain and phase margins (GPMs) is approached and solved. Accurate expressions of GPMs in terms of the PID and process parameters are also obtained. Based on these results, simple PID tuning rules are then derived for typical process models. The new rules are shown to give improved disturbance rejection while maintaining the same peak sensitivities as compared to the well-known simple internal model control (SIMC) rules.

We then present a systematic approach of combining two-degrees-of-freedom (2DOF) design with direct synthesis (DS) for designing controllers which give desired closed-loop transfer functions. Explicit PID tuning rules are obtained by approximating the ideal controllers appropriately as PID controllers or PID-C controllers (i.e., PID controllers in series with lead-lag compensators). Next we investigate the very recent closed-loop setpoint response (CSR) method for tuning PI controllers in an analytical manner. A common PI tuning rule is obtained without using explicit models for both IPTD and first-order plus time delay (FOPTD) processes. The rule has a form similar to a recent one concluded from numerical experiments and turns out to give satisfactory closed-loop performance for a broad range of processes.

Conventionally, CVs are assumed to be known or given before a PID control design. The assumption, however, may neither be necessary nor be rational. It has been found in

many applications that CVs need to be selected properly for maximizing product utility when a process is perturbed or the measurements are corrupted by noises. This has motivated the proposal of SOC for selecting CVs for near optimal operation. In the second part of the thesis, we firstly investigate the local solutions of available SOC further and then deal with two new problems arising in the SOC design.

We give more complete analytical characterizations of the local solutions for SOC to minimize worst-case loss and average loss, respectively. The available solutions for SOC to minimize worst-case loss are extended in a more general form and the available solutions for SOC to minimize average loss are proved to be complete. The new results contribute to clarifying the relation between these two classes of solutions for SOC.

We then investigate the problem of SOC with tight operational constraints. For such a problem, if ideal SOC design is adopted, it will not only have to detect and distinguish the different regions of active constraints but also require frequent switching between different sets of CVs as selected for the corresponding regions. This tends to complicate the design and implementation. To keep simple, we propose a novel solution with a fixed set of CVs. The solution provides a suboptimal yet simple way to select CVs which achieve SOC.

Finally, note that available SOC design assumes a steady-state process and minimizes a cost defined for the steady state. SOC design for a dynamic process which minimizes a cost defined for the whole operation interval has been unclear so far. Such design, however, is practically important since in some applications the transient operation costs count much and are unignorable. We formulate the dynamic SOC (dSOC) problem and solve it for a local solution via perturbation control approach. A linear example is used to illustrate the usefulness of the theoretical results.

Contents

Acknowledgements.....	i
Abstract	iii
Contents	v
Chapter 1 Introduction	1
1.1 Motivations and Objectives	1
1.1.1 On PID Controller Tuning.....	1
1.1.2 On SOC Design.....	4
1.2 Organization and Contributions of the Thesis.....	5
1.2.1 Organization of the Thesis	5
1.2.2 Contributions of the Thesis	6
Chapter 2 PID Controller Tuning and SOC: A Brief Introduction	8
2.1 PID Controller Tuning	8
2.2 SOC Design.....	15
Chapter 3 PID Controller Tuning with Specified GPMs for IPTD Processes	21
3.1 Introduction.....	21
3.2 Derivation of the PI/PD/PID Tuning Formulas and the GPM Formulas ..	23
3.2.1 PI Tuning Formula and GPM-PI Formula	24
3.2.3 PID Tuning Formula and GPM-PID Formula.....	30
3.3 Application to Unifying the Existing Tuning Rules.....	35
3.4 Conclusions	37
Chapter 4 Simple Analytical PID Tuning Rules	38
4.1 Introduction.....	38
4.2 Derivation of the PID Tuning Rules	39

4.2.1 The Case of an IPTD Process.....	40
4.2.2 The Case of an FOPTD Process.....	43
4.2.3 The Case of an SOPTD Process.....	44
4.2.4 Other Processes	45
4.2.5 Choice of the Parameter k_1	46
4.3 Numerical Examples	52
4.3.1 Simulation Settings	52
4.3.2 Simulation Results	54
4.4 Conclusions.....	59
Chapter 5 PID and PID-C Controller Tuning by 2DOF-DS Approach	60
5.1 Introduction.....	60
5.2 Design Principles of 2DOF-DS.....	62
5.2.1 Design for Desired s2o Response (Method 1)	64
5.2.2 Design for Desired d2o Response (Method 2).....	66
5.3 PI/PID Controller as the Feedback Controller	68
5.3.1 PI/PID Controller Design with Method 1	68
5.3.2 PI/PID Controller Design with Method 2	73
5.4 PID-C Controller as the Feedback Controller.....	76
5.5 Numerical Examples	81
5.5.1 PI Control	83
5.5.2 PID Control	86
5.5.3 PID-C Control	91
5.6 Conclusions.....	96
Chapter 6 Analytical PI Controller Tuning Using Closed-loop Setpoint Response ...	97
6.1 Introduction.....	97
6.2 Derivation of the PI Tuning Rule	99
6.3 Simulation Results	108
6.4 Conclusions.....	114

Chapter 7 Further Results on the Local Solutions to SOC	115
7.1 Introduction	115
7.2 Local SOC	116
7.3 Main Results	118
7.4 Conclusions	123
Chapter 8 Local SOC of Constrained Processes	124
8.1 Introduction	124
8.2 Local SOC	126
8.3 Local SOC with Constraints	129
8.3.1 Exact Local Method	130
8.3.2 Measurement Subset Selection	133
8.4 Case Study: Forced Circulation Evaporator	135
8.5 Conclusions	141
Chapter 9 Selecting CVs as Optimal Measurement Combinations via Perturbation Control Approach	142
9.1 Introduction	142
9.2 Problem Formulation	144
9.3 Local Optimal Solution	148
9.3.1 Optimal Perturbation Control Law	149
9.3.2 Optimal Selection of Γ	160
9.4 Numerical Example	163
9.5 Conclusions	167
Chapter 10 Summary and Future Work	168
10.1 Summary	168
10.2 Future Work	170
10.2.1 On PID Controller Tuning	170
10.2.2 On SOC Design	171

Appendices	173
A Approximate Analytical Solutions of β for (3.11) and (3.34)	173
A.1 An Approximate Solution of (3.11).....	175
A.2 An approximate solution of (3.34).	176
B Selecting a Proper Damping Ratio ζ	180
C Deriving the Necessary Conditions for a Minimum of (9.36)	183
Author's Publications.....	187
Bibliography	189

List of Tables

Table 4.1 PID settings for typical processes.....	52
Table 4.2 PI settings and performance summary of exemplary IPTD processes	55
Table 4.3 PID settings and performance summary of exemplary FOPTD and SOPTD processes.....	56
Table 4.4 PID settings and performance summary of exemplary ILPTD processes.....	57
Table 4.5 PID settings and performance summary of exemplary DIPTD processes	58
Table 5.1 PI settings for typical process models (Method 1)	71
Table 5.2 PID settings for typical process models (Method 1)	71
Table 5.3 PI settings for typical process models (Method 2)	75
Table 5.4 PID settings for typical process models (Method 2)	76
Table 5.5 Parameter settings of the PID-C feedback controllers	80
Table 5.6 PI controller settings and performance summary for explemary processes.	85
Table 5.7 PID controller settings and performance summary for explemary processes.	89
Table 5.8 PID-C controller settings and performance summary for explemary processes.	94
Table 6.1 PI settings for Shams-Skog's and proposed rules.....	110
Table 8.1 Variables and optimal values	136
Table 8.2 Average local and nonlinear losses for the self-optimizing CV candidates.....	139
Table 9.1 Algorithm for solving a local optimal LMF gain when $W_v = 0$ and $t_f = \infty$	158

List of Figures

Figure 1.1 Organization of the thesis.	6
Figure 2.1 Block diagram of typical feedback control system.....	9
Figure 2.2 Typical setpoint response.....	10
Figure 3.1 Control system loop.	24
Figure 3.2 GPMs estimated by GPM-PI formula versus true GPMs.	27
Figure 3.3 GPMs estimated by GPM-PD formula versus true GPMs.	30
Figure 3.4 GPMs estimated by GPM-PID formula versus true GPMs.	34
Figure 3.5 Relative estimation errors of the results in Figure 3.4.	35
Figure 4.1 Block diagram of feedback control system.....	39
Figure 4.2 The true k_2 as $2/k_1 - 1 + \sqrt{(2/k_1 - 1)^2 + 1}$ v.s. its approximate as $4/k_1 - 2$	42
Figure 4.3 The relations between the margins and the tuning parameter k_1	50
Figure 4.4 Relative errors of the margins as computed by analytical formulas in (4.23) for case ii.....	50
Figure 4.5 Relations between peak sensitivities and the tuning parameter k_1	51
Figure 4.6 Responses of PI control of IPTD processes with different delays.....	55
Figure 4.7 Responses of PI control of an FOPTD process and PID control of an SOPTD process	56
Figure 4.8 Responses of PID control of ILPTD processes with different delays	57
Figure 4.9 Responses of PID control of DIPTD processes with different delays	58
Figure 5.1 2DOF control system.	63

Figure 5.2 Performance index values attained with different PI tuning rules.	84
Figure 5.3 Output responses of processes and PI controllers for processes E2 and E4	84
Figure 5.4 Performance index values attained with different PID tuning rules.	87
Figure 5.5 Output responses of processes and PID controllers for processes E5 and E8. ..	87
Figure 5.6 Output responses of processes and PID controllers for processes E12 and E15.	88
Figure 5.7 Output responses of processes and PID controllers for processes E18 and E20	88
Figure 5.8 Performance index values attained with different PID-C rules.	93
Figure 5.9 Setpoint and disturbance responses attained with different PID-C/PID rules..	93
Figure 6.1 Block diagram of feedback control system.....	99
Figure 6.2 Setpoint response with P control.....	100
Figure 6.3 $M_p - \zeta$ curve.....	103
Figure 6.4 Ouput responses for PI control of typical processes.	111
Figure 6.5 Output responses for PI control of typical processes.....	112
Figure 6.6 Effect of detuning α	113
Figure 6.7 Detuning process of the P controller gain k_{c0} using the proposed method....	114
Figure 8.1 Schematic of forced-circulation evaporator.	135
Figure 8.2 Average local losses of best CV candidates with n measurements obtained using available and proposed (explicit constraint handling) exact local methods....	138
Figure 8.3 Variation of P_2 with use of CVs obtained using available exact local method with cascade control and the proposed approach.	141

Figure 9.1 Economic cost increment ($E(\delta^2 J_0)$) as functions of the weighting factor (ρ) and the disturbance covariance (σ), under optimal LMF perturbation control.	165
Figure 9.2 Economic cost increment ($E(\delta^2 J_0)$) as functions of the weighting factor (ρ) and the disturbance covariance (σ), under optimal perturbation control with different CV feedbacks.	165
Figure 9.3 LMF control v.s. classic LQG control.	166
Figure A.1 The maximal absolute values of the relative errors of the approximate solutions, as functions of the boundary point x_b	174
Figure A.2 Typical relative estimation errors of β and A_m	180
Figure B.1 The achieved time-domain indices of system described in (4.7) as the tuning parameters ζ and k_1 change.	183

Notations

$:=$	defined as
$\equiv$	always equal to
$\square$	end of proof
$\mathfrak{R}$	field of real numbers
$\mathfrak{R}^n$	field of real vectors of dimension n
$\mathfrak{R}^{n \times m}$	field of real matrices of dimension $n \times m$
$ x $	absolute value of a real number x
$\ a\ _1$	1 norm of vector a
$\ a\ _2$ or $\ a\ $	2 (or Euclidean) norm of vector a
$\ a\ _\infty$	infinity norm of vector a
I_n (I)	identity matrix with dimension $n \times n$ (compatible dimension)
A_{ij}	entry that lies in the i -th row and j -th column of A
A^T	transpose of A
A^{-1}	inverse of A
A^{-T}	transpose of A^{-1}
$\text{rank}(A)$	rank of A
$\text{tr}(A)$	trace of A
$\ A\ _1$	1 norm of A

$\ A\ _2$ or $\ A\ $	Euclidean norm of A
$\ A\ _\infty$	infinity norm of A
$\text{diag}(a_1, a_2, \dots, a_n)$	$n \times n$ diagonal matrix with a_i as its i -th diagonal element
$X > Y$ ($X \geq Y$)	$X - Y$ is positive definite (semidefinite)
$E(\bullet)$	expectation operator
$\inf$ (min, sup, max)	infimum (minimum, supremum, maximum)
$\arg$	argument

Acronyms

CSR	closed-loop setpoint response
CVs	controlled variables
dSOC	dynamic self-optimizing control
d2o	(load) disturbance-to-output
DIPTD	double integral plus time delay
DOF	degree(s) of freedom
DS	direct synthesis
FOPTD	first-order plus time delay
GM	gain margin
GPMs	gain and phase margins
IAE	integrated absolute error
ILPTD	integrating with first-order lag plus time delay
IMC	internal model control
IPTD	integral plus time delay
LMF	linear measurement feedback
LQG	linear quadratic Gaussian
LQR	linear quadratic regulator
MCM	measurement combination matrix
MSV	minimum singular value
P	proportional
PD	proportional-derivative
PI	proportional-integral
PID	proportional-integral-derivative

PID-C	PID controller in series with a (lead-lag) compensator
PM	phase margin
sSOC	static (or steady-state) self-optimizing control
s2o	setpoint-to-output
SIMC	simple (or Skogestad's) internal model control
SOC	self-optimizing control
SOPTD	second-order plus time delay
TD	time delay
2DOF	two-degrees-of-freedom
2DOF-DS	two-degrees-of-freedom direct synthesis

Chapter 1

Introduction

1.1 Motivations and Objectives

Motivations and objectives of our studies on proportional-integral-derivative (PID) controller tuning and self-optimizing control (SOC) design are stated, respectively.

1.1.1 On PID Controller Tuning

It is well-known that many PID controllers applied in industry remain poorly tuned [1]. This is partially due to lack of simple, efficient and robust PID tuning rules. This has motivated decades' research on PID controller tuning, i.e., tuning the P, I and D gains of a PID controller for desired closed-loop performance and robustness.

Although a PID controller has only three parameters, it is very difficult to tune them properly. Since the proposal of Ziegler-Nichols rule in 1942 [2], there have been a huge number of rules proposed for tuning PID controllers in the past seven decades. In the 1980's, academic research on PID controller tuning increased as the computing power of the microprocessors advanced which allows more flexible PID controller design. The research was accelerated in 1990's and the zest in it spreads into 2000's [3]. Various methods and skills have been used to derive the rules for satisfying various specifications on the performance and robustness with different processes. Despite the flourishing results, simple, efficient and robust PID tuning rules applicable to a wide range of processes are still in exploration and highly demanded in industry. This is reflected in a recent survey of

the state-of-art applications of PID control [4]. Such demands motivate our studies on PID controller tuning in general. More specific motivations and objectives of our studies are summarized as follows.

PID controller tuning for integral plus time delay (IPTD) processes has been extensively studied in the past two decades [5]. The tunings usually rely on Taylor or Páde approximations of the time delay components and no general closed-form solution was obtained due to nonlinearity of the problems. This is the case when a PID controller is tuned to satisfy specified gain and phase margins (GPMs). Except for some special GPMs, case-by-case numerical solutions had to be used, which prevents an easy-to-use rule for applications. We will revisit this problem and solve it for an explicit solution of the PID parameters. The solution will contribute to a new way of deriving simple tuning rules for typical processes.

The aforementioned solution indicates a common form of the PI parameters, which are explicit functions of the process parameters together with two dimensionless scaling factors. By establishing a relation between these two factors for ensuring certain desired performance, it is possible to derive a simple and efficient tuning rule containing a single tuning factor. Indeed such a relation can be established by borrowing the idea of simple (or Skogetad's) internal model control (SIMC) [6] that makes the approximate damping ratio of the closed-loop system be one. This motivates us to derive a new set of simple PI/PID tuning rules as alternatives to the SIMC counterparts. The new rules will be developed based on an IPTD process model and then be extended to other typical models.

As a model-based PID tuning method, direct synthesis (DS) has a long history and has attracted continuous attention [7-8]. In the DS method, the PID controllers are obtained as appropriate approximations of the controllers that lead to specified closed-loop setpoint-to-output or disturbance-to-output transfer functions. The DS method is very general in

nature, in that any controller design can be interpreted as achieving certain closed-loop transfer functions from which the controller can be resolved. Yet there is no systematic approach to carrying out DS when the controller is restricted to a PID controller. This is also the case when a two-degrees-of-freedom (2DOF) design is required for improving setpoint following performance. This motivates us to do a detailed study and present a systematic approach to using DS for PID controller tuning, generating explicit tuning rules, while taking the 2DOF design into account at the meantime. In addition, notice that a PID controller in series with a compensator (PID-C for short) has recently been proposed as an alternate to a PID controller which may achieve improved performance [9-10]. We will also study the tuning of PID-C controllers for different process models using the DS-2DOF method and derive explicit tuning rules for them respectively.

The aforementioned studies on PID controller tuning all use certain parametric tuning methods. In contrast, very recently a novel nonparametric method, the closed-loop setpoint response (CSR) method has been proposed for PI controller tuning [11]. This method avoids the troubles due to persistent closed-loop oscillations as required by the well-known Ziegler-Nichols method [2] and relay-feedback methods [12]. In this method, a CSR experiment is carried out with a proportional controller to give an overshoot of around 30%. The data of the overshoot, the peak time, and the steady-state output change are recorded and then used to determine the PI parameters using an explicit rule. This makes PI tuning very easy. And the rule has been found to be applicable to a wide range of processes. However, the rule was concluded from numerical experiments to match the SIMC rule and no analytical derivation or explanation is available. This motivates our analytical study on the CSR method. Although the analysis will ultimately be approximate due to the existence of time delay in the process, the analytical result will provide insights into the CSR tuning method and explain the rationale of the CSR rule to some extent.

1.1.2 On SOC Design

SOC is used to select controlled variables (CVs) so that a process achieves near-optimal operation in spite of disturbances and implementation errors, when the CVs are controlled at the setpoints [13]. Alternatively we can interpret SOC as a kind of simple and suboptimal implementation of online optimal control [14]. The link between SOC and PID control is through the CVs: PID control is responsible for the system performance, given the CVs; while SOC is responsible for selecting the optimal set of CVs for best achievable economic profit under a given control (say PID control), when the process is disturbed and the control implementation involves errors. Studies on PID and SOC are therefore closely related.

In control design, it is usually assumed that the CVs are given or known a priori. This assumption, however, may neither be necessary nor be rational. The un-necessity is due to the fact that sometimes it is too difficult to know which variables should be selected as the CVs when there are a lot of candidates. This is the case in an industrial plant where there are a lot of variables to be controlled while the manipulated variables are limited and less in number. On the other hand, given a set of CVs, it may not be optimal for leading to the highest product utility (or the lowest operational cost, equivalently) when the process is perturbed or the measurements are corrupted by noises. The CVs thus should be selected to optimize the product utility in the presence of disturbances and operational constraints. This rationale motivates the concept of SOC for selecting CVs for near optimal operation [13], which is suboptimal due to setpoint constraints on the CVs as compared to ideal real-time optimization without such constraints [15].

Original SOC assumes that the operation constraints are either always active (the constraint limits are constantly touched) or always inactive (the constraint limits are constantly not touched) during the whole interval of operation [13]. Various methods have

been proposed for SOC design with a varying set of active constraints, e.g., the split-range controllers [16], the multi-parametric programming method [17] and the cascade control strategy [18], etc. The previous methods, however, all require control structures and implementations which are more complex than the original SOC. Retaining the simplicity of the design is highly expected in applications. We are interested in devising an SOC design method to resolve the difficulty when the set of active constraints vary. The novel method should keep simple the SOC design while achieving near optimal operation. We will study this in detail and propose a new method as simple as the original SOC for carrying out the SOC design subject to a changing set of active constraints.

On the other hand, we note that the existing SOC designs all assume steady-state processes and minimize cost functions defined at the steady states. Practical processes, however, are dynamic where transient operational costs may be significant. Therefore SOC design minimizing a cost defined for the whole operation interval of a dynamic process is more general and holds practical interest [19-20]. As far as we know, this problem is still open and even no complete formulation appears in literature. We shall make an attempt to formulate and solve such a design problem. As an initial step, a local solution based on linearization will be explored. Insights gained from such a solution will be discussed. The formulation and solution would contribute to more complete and practical solutions in the future.

1.2 Organization and Contributions of the Thesis

1.2.1 Organization of the Thesis

This thesis consists of one chapter of a brief introduction to PID controller tuning and SOC design, four chapters on PID controller tuning, three chapters on SOC design and one chapter on summary of the thesis together with discussions on some future work. The

organization is depicted in Figure 1.1, where the connection between PID controller tuning and SOC design is through CV selection as indicated by a dash bidirection arrow.

Figure 1.1 Organization of the thesis.

Each chapter deals with a particular problem and is almost independent of other chapters, with an exception that Chapter 4 is developed based on Chapter 3. For clarity, literature review is distributed into each chapter on the particular problems while a survey in general is made in Chapter 2. The readers are encouraged to read Chapter 2 for the general background knowledge and then go directly to the chapter that he/she is interested in.

1.2.2 Contributions of the Thesis

The contributions of the thesis are summarized chapter by chapter as follows.

In Chapter 2, a brief introduction is made to PID controller tuning and SOC design, where the concepts and developments of them are reviewed.

In Chapter 3, explicit expressions of PI/PD/PID parameters satisfying specified GPMs for an IPTD process are derived and so are accurate expressions of the GPMs attained by a given PI/PD/PID controller. The results unify a large number of existing rules into the same framework of tuning PI/PD/PID controllers based on GPM specifications.

In Chapter 4, new simple PID tuning rules are obtained for typical process models based on the PI tuning formula obtained in Chapter 3. The new rules are able to achieve

similar or better disturbance rejection while giving the same peak sensitivities as compared to the SIMC counterparts.

In Chapter 5, a 2DOF-DS method is proposed for deriving explicit PID and PID-C tunings rules for typical process models, which are shown to be advantageous over recent rules by a series of numerical examples.

In Chapter 6, a simple PI tuning rule is developed with the recent CSR method. The rule is simple to use and shown to be very efficient for a broad range of processes.

In Chapter 7, some analytical results are reported on the local solutions for SOC, which give a solution for SOC to minimize worst-case loss which is more general than the available solution and meanwhile prove the completeness of the available solutions for SOC to minimize average loss.

In Chapter 8, a new approach is proposed to dealing with SOC design of constrained processes. It treats the problem as the available SOC subject to process constraints. The problem is convex and can be solved efficiently. The proposed design results in suboptimal CVs in general but retains the important feature of simplicity of SOC.

In Chapter 9, the problem of dSOC is formulated and a local solution is obtained by adopting a perturbation control approach. It is found that the solution is essentially associated with an optimal control law as applied in practice.

Chapter 10 concludes the thesis and states the future work that could be conducted on PID controller tuning and SOC design, respectively.

Chapter 2

PID Controller Tuning and SOC: A Brief Introduction

This chapter briefly introduces the concepts and developments of PID controller tuning and SOC design. More detailed reviews of relevant existing results are left to the beginning of each chapter later on.

2.1 PID Controller Tuning

PID controllers are so far the most widely adopted controllers in industry owing to their satisfactory cost-effectiveness [1, 3, 21]. A PID controller can be expressed in a transfer function of different forms. Typical forms used in research and applications are

$$c(s) = \begin{cases} k_p + \frac{k_i}{s} + k_d s, & \text{(parallel form),} \\ K_c \left(1 + \frac{1}{T_i s} + T_d s \right), & \text{(ideal/standard/non-interacting form),} \\ k_c \left(1 + \frac{1}{\tau_i s} \right) (1 + \tau_d s), & \text{(series/interacting form).} \end{cases} \quad (2.1)$$

The generality of the forms above decreases in order. The parallel form is the most general form which allows flexible assignment of the controller parameters. The other two forms are special cases of the parallel form. An interacting form can always be converted into a non-interacting form, but the reverse is true only if $\tau_d \leq \tau_i/4$, in which case we have

$$K_c = k_c \left(1 + \frac{\tau_d}{\tau_i} \right), \quad T_i = \tau_i \left(1 + \frac{\tau_d}{\tau_i} \right), \quad T_d = \frac{\tau_d}{1 + \tau_d/\tau_i}. \quad (2.2)$$

Other forms of PID controllers exist but are less popular [1, 3, 21]. Since the derivative action is not causal, in practice it is usually implemented in series with a filter having a small time constant, e.g., T_d/N , where N typically ranges from 2 to 20 [3]. Alternatively, a filter may be added in series with the PID controller to filter the measured signals. The equivalent controller transfer function is

$$c_{eq}(s) = c(s)g_f(s) = K_c \left(1 + \frac{1}{T_i s} + T_d s \right) \frac{1}{(T_f s)^2/2 + T_f s + 1}, \quad (2.3)$$

where a second-order filter with a relative damping ratio of $1/\sqrt{2}$ is used. The filter time constant T_f is typically chosen as T_i/N for PI control or as T_d/N for PID control, where N ranges from 2 to 20 [3]. Extra studies are required to determine the value of N if the performance is sensitive to the choice [21].

Consider the control system described in Figure 2.1, where u is the manipulated control input, d the disturbance, n the measurement noise, y the controlled output, y_s the setpoint (reference) for the controlled output, $c(s)$ the PID controller transfer function, and $g(s)$ the process transfer function. The problem of PID controller tuning is basically to determine the three parameters in any of the forms in (2.1) so that desired closed-loop performance and robustness are achieved for a given process.

Figure 2.1 Block diagram of typical feedback control system.

As a first step, we need to specify the requirements on closed-loop performance and robustness. The requirements can be quantified in either the time or frequency domain. Some well-known metrics are listed below, which can be classified into metrics for

performance and metrics for robustness [3]. Note that the classification is not strict since the metrics of performance usually reflect on the robustness also, and vice versa. The metrics used most frequently are indicated in *italic font*. The variables used can be found in Figure 2.1 and Figure 2.2. Only deterministic metrics are considered, while stochastic metrics also appear in literature [21].

Figure 2.2 Typical setpoint response.

Metrics to Quantify Performance

I. Metrics Based on Setpoint or Load Disturbance Step Time Response

- *Integrated error (IE)*: $IE = \int_0^{\infty} e(t) dt$
- *Integrated absolute error (IAE)*: $IAE = \int_0^{\infty} |e(t)| dt$
- *Integrated time multiplied absolute error (ITNAE)*: $ITNAE = \int_0^{\infty} t^n |e(t)| dt$
- *Integrated squared error (ISE)*: $ISE = \int_0^{\infty} e^2(t) dt$
- *Quadratic criterion*: $QE = \int_0^{\infty} (e^2(t) + \rho u^2(t)) dt$, where ρ is a weighting scalar

II. Metrics Based on Setpoint Step Time Response

- Rise time t_r
- Peak time t_p
- *Settling time* t_s
- *Overshoot*: $M_p = (y_p - y_\infty)/y_\infty$
- Steady-state error: $e_{ss} = y_s - y_\infty$
- *Decay ratio*: the ratio between two consecutive maxima of the error for a step change in setpoint or load

III. Metrics Based on Frequency Responses of Open-loop Transfer Functions

- Phase crossover frequency: ω_{pc} , the frequency where the phase of the loop transfer function is equal to 180°
- Gain crossover frequency: ω_{gc} , the frequency where the amplitude of the loop transfer function is equal to 1

IV. Metrics Based on Frequency Responses of Closed-loop Transfer Functions

- *Peak amplitude of the transfer function from the measurement noise to the control signal*: $M_{un} = \max_{\omega} |c(j\omega)/(1 + g(j\omega)c(j\omega))|$
- Peak sensitivity frequency: ω_{ms} , the frequency where the peak sensitivity occurs
- Peak complementary sensitivity frequency: ω_{mt} , the frequency where the peak complementary sensitivity occurs
- Resonance peak: R_p , the largest value of the frequency response (which equals M_t (defined later) if unity error feedback is used)
- Peak frequency: ω_p , the frequency where the resonance peak occurs

- *Bandwidth:* ω_b , the frequency where the gain has decreased to $1/\sqrt{2}$

Metrics to Quantify Robustness

- *Gain margin:* $A_m = 1/|g(j\omega_{pc})c(j\omega_{pc})|$ (typical 2 ~ 8)
- *Phase margin:* $\phi_m = \pi + \arg(g(j\omega_{gc})c(j\omega_{gc}))$ (typically $30^\circ \sim 60^\circ$)
- *Peak sensitivity:* $M_s = \max_{\omega} |1/(1 + g(j\omega)c(j\omega))|$ (typically 1.2 ~ 2.0)
- *Peak complementary sensitivity:* $M_t = \max_{\omega} |g(j\omega)c(j\omega)/(1 + g(j\omega)c(j\omega))|$
(typically 1.0 ~ 2.0)
- *Relative delay margin:* $r_{dm} = \pi\phi_m/(180\omega_{gc}\theta)$
- *Stability margin:* $S_m = 1/M_s$ (typically 0.5 ~ 0.8)

The recommended values in design are given in the brackets. The above metrics are frequently used in control design [3]. Note that feedback control is mainly responsible for load disturbance attenuation, measurement noise rejection and robustness to process uncertainties, while setpoint following performance can be left to feedforward control [3]. When the controller is restricted to a PID controller, the metrics of interest can mainly be k_i , M_{un} , M_s and M_t [3]. A larger integral gain (k_i) is responsible for a smaller IE when disturbance response is considered. A smaller M_{un} is responsible for better rejection of measurement noise. A smaller M_s is responsible for less sensitivity to variations in process dynamics. And a smaller M_t is responsible for stronger robustness of the closed-loop system to uncertainties in the process dynamics. In a sense, both M_s and M_t capture the robustness of a control system. They can be combined to define a new robustness measure so that the Nyquist curve of the loop transfer function is ensured to lay outside a circle that includes the two circles required by M_s and M_t [3].

PID controller tuning is basically tuning the PID controllers satisfying specified indices of performance and robustness in terms of the metrics above. By using different metrics, different tuning rules may be attained. Further, different tuning methods may also lead to different tuning rules for the same specifications. According to the process information in use, PID tuning methods can roughly be divided into three classes: *parametric tuning methods*, *nonparametric tuning methods* and *model-free tuning methods* [22]. These three kinds of methods are introduced briefly as follows.

Parametric tuning methods. The parametric tuning methods are model-based methods. They assume and identify a process model captured by finite parameters and then derive the PID tuning rules in terms of the model parameters (where some tuning factors may also exist). The model is usually assumed to be IPTD, FOPTD, second-order plus time delay (SOPD), or integral with first/second-order lag plus time delay (ILPTD), double integral plus time delay (DIPTD), etc. The parametric methods comprise the main methods for PID controller tuning in literature. There are a huge number of tuning rules of this class [1, 22], such as the Ziegler-Nichols rules using setpoint response [2], the Chien-Hrones-Reswick rules [23], the Cohen-Coon rules [24], the IMC rules [25], the DS rules [7, 26], the AMIGO rules [27], and the SIMC rules [6], just to list a few. The rules may or may not be sensitive to model errors. In general the recent rules lead to better performance while ensuring similar robustness as compared to the old ones [28]. Despite tons of tuning rules obtained, there is always some room for improving the rules to achieve better tradeoff between closed-loop performance and robustness.

Nonparametric tuning methods. This class mainly consists of two methods. One uses the two parameters of ultimate gain and ultimate frequency, and the other uses the steady-state output, peak time, and overshoot of a closed-loop setpoint response with P control. The ultimate gain and ultimate frequency are identified as the gain and frequency

when the closed-loop system oscillates periodically under proportional control or relay feedback [3]. In 1940's, Ziegler and Nichols [2] first used the proportional control approach; and in 1980's, Åström and his coworkers devised the relay feedback approach [29]. The relay feedback approach has become well-known and popular since it does not require the closed loop to reach its stability limit and can identify the parameters more efficiently. With the ultimate gain and frequency identified, the PID parameters are expressed in terms of them. This has led to a rich class of PID tuning rules with wide applications [12, 22].

More recently, a novel CSR method has been proposed to give PI (or even PID) tuning rules very efficiently [11]. The method requires only to do a CSR experiment and record the values of steady-state output change (Δy_∞), peak time (t_p), and overshoot (M_p). The PI tuning rule is given in terms of the recorded quantities together with a tuning factor that controls the tradeoff between performance and robustness. This kind of tuning rules comprises a newest and very promising development for simple PID controller tuning.

Other nonparametric methods also appear such as Fourier methods and phase-locked loop methods, etc. [22]

Model-free tuning methods. This class of methods does not require any process model or priori experiments. All the tuning work is done online. These methods might seem remote from the mainstream control engineering concerns [22]. But they do have a lot of developments recently. As examples, the iterative feedback tuning [30-31] and its variant the controller parameter cycling tuning method [32] both fall into this class. This class of methods is not matured yet and requires in-depth investigations [22].

The above summarizes the methods for PID controller tuning. An important issue should be alerted is that the PID controller should be tuned mainly for desired performance of disturbance response and desired robustness to process variations and

uncertainties. Performance of setpoint response can be tuned independently by a feedforward controller. That is, a 2DOF design is usually essential to achieve desired setpoint and disturbance responses at the same time, together with required robustness to uncertainties [3]. This tends to decouple the designs for required setpoint (or servo) and disturbance (or regulatory) performances. When measurement noise is also taken into account, however, a PID controller may also have to be tuned for good setpoint following performance even if a 2DOF design is adopted. This is because that the low-frequency measurement noise or disturbance, if any, entering the feedback channel acts as a servo signal and influences the process as if it is due to setpoint changes.

2.2 SOC Design

In practice, when a process is subjected to disturbances, an ideal optimal controller repeatedly optimizes the process online [15, 33]. The repeated optimization, however, requires estimation of states and model parameters, and is also computationally costly [33-34]. To overcome these drawbacks, several approaches have recently been proposed for feedback-based optimization, such as extremum-seeking control [35-36], SOC [13, 37] and tracking necessary conditions of optimality [38-39].

The available SOC considers the selection of CVs regarding a steady-state process, where keeping the selected CVs at constant setpoints using the feedback controller automatically leads the process to acceptable operating conditions. In addition to significant reduction in computational load required for optimization, SOC offers simpler implementation policy in comparison with the use of ideal optimizing controller. The term ‘acceptable operating conditions’ in accordance to SOC concept is quantified as loss, *i.e.*, the difference between the values of the cost function, when SOC policy and the ideal optimal controller are implemented. Here, the loss depends on the selected CVs. Thus, the

main issue in SOC is to find CVs among the possible alternatives, which lead to the least loss.

CV selection based on direct evaluation of the nonlinear model and cost function requires solving large dimensional nonconvex optimization problems [40]. Thus local methods, which employ linearized process model and quadratic approximation of the loss function, are instead used to find promising CV candidates. The first local method developed to select CVs is the minimum singular value (MSV) rule [41]. The MSV rule, however, is approximate and may lead to suboptimal set of CVs [42]. More recently, exact local methods to select CVs through minimization of worst-case [40] and average loss [43] have been proposed. These methods can be used for selecting CVs as a subset or linear combinations of available measurements, where the latter approach can provide lower loss. Different approaches for finding the locally optimal combination matrix have recently been proposed [34, 43-46]. To make the application of local methods viable for large-scale processes, efficient branch and bound methods have been proposed for selecting a subset of available measurements, which can be used directly or combined as CVs [47-49].

As follows we formulate the static SOC problem from an optimization standpoint.

Problem Formulation. Some notations are defined. The variables $x \in \mathfrak{R}^{n_x}$, $u_0 \in \mathfrak{R}^{n_{u_0}}$, $y \in \mathfrak{R}^{n_y}$, $\tilde{y} \in \mathfrak{R}^{n_{\tilde{y}}}$, $c \in \mathfrak{R}^{n_c}$, $d \in \mathcal{D} \subseteq \mathfrak{R}^{n_d}$, $e \in \mathcal{E} \subseteq \mathfrak{R}^{n_e}$ and $H \in \mathcal{H} \subseteq \mathfrak{R}^{n_u \times n_y}$ denote the state, inputs (or DOF), outputs, measurements (i.e., measured outputs), CVs, disturbances, measurement noises (or implementation errors in general) and measurement combination matrix (MCM), respectively; $\mathcal{D}$, $\mathcal{E}$ and $\mathcal{H}$ are the domains or admissible sets of the variables. The scalar function J denotes the steady-state (economic) cost to be minimized for optimal operation.

SOC can be interpreted as *steady-state* optimal control with operational and setpoint constraints. The SOC design is essentially to solve the problem of optimization

$$\begin{aligned}
& \min_h E(J(x, u_0, d)) \\
& \text{s.t.}, f(x, u_0, d) = 0, \\
& \quad g(x, u_0, d) \leq 0, \\
& \quad y = f_y(x, u_0, d), \\
& \quad \tilde{y} = f_{\tilde{y}}(y, e), \\
& \quad c = h(\tilde{y}) = c_s, \\
& \quad d \in \mathcal{D}, e \in \mathcal{E}, h \in \mathcal{H}.
\end{aligned} \tag{2.4}$$

In (2.4), f is the equality constraint corresponding to the system model equation; g is the inequality constraint corresponding to physical limits in operation; $h(\tilde{y}) = c_s$ denotes the setpoint constraint, where c_s is a given constant setpoint; and $\mathcal{H}$ is the functional domain of h . In the absence of the setpoint constraint, (2.4) formulates an *optimal control problem*; if no further expectation is taken over the disturbances and noises, then (2.4) formulates a *real-time optimal control problem*. And if the objective function ‘ $E(J(x, u_0, d))$ ’ is replaced by ‘ $\max_{d,e} J(x, u_0, d)$ ’, then the SOC minimizes the worst-case cost which is not usual in practice [43].

The above SOC problem can be simplified by making appropriate assumptions. Assume that some of the active constraints (where ‘active’ means the inequality constraints take the equality) are always active. Let such active constraints be $g_i(x, u_0, d) \equiv 0$, where $g_i(\bullet)$ denotes certain components of $g(\bullet)$. Assume that some DOF are consumed to control such active constraints, leaving the rest DOF denoted as $u \in \mathfrak{R}^{n_u}$. Consequently the consumed DOF can be expressed in terms of u and d . From $f(x, u_0, d) = 0$ and $g_i(x, u_0, d) \equiv 0$, the state x can be solved in terms of u and d (which is often the case when we restrict to considering the steady state) [50]. Substituting the solved x into (2.4), we get a reduced-space SOC problem:

$$\begin{aligned}
& \min_h E(J(u, d)) \\
& \text{s.t., } y = f_y(u, d), \\
& \quad \tilde{y} = f_{\tilde{y}}(y, e), \\
& \quad c = h(\tilde{y}) = c_s, \\
& \quad z = f_z(u, d) \leq 0, \\
& \quad d \in \mathcal{D}, e \in \mathcal{E}, h \in \mathcal{H}.
\end{aligned} \tag{2.5}$$

In (2.5), the inequality constraints are the original constraints ($g(x, u_0, d) \leq 0$) excluding the always active ones. Note that some of the function names in (2.4) are overloaded in (2.5) for convenience.

Thus the SOC problem transforms into solving (2.5) for an optimal h that leads to minimal cost while satisfying the setpoint and operational constraints. To make sure that the setpoints c_s be attained under the given DOF u , the dimension of u must be at least as large as that of c_s . Without loss of generality, we assume that $n_c = n_u$.

Let the CVs be expressed as linear combinations of measurements, i.e., $c = h(\tilde{y}) = H\tilde{y}$, where $H \in \mathbb{R}^{n_u \times n_y}$ is a constant matrix to be determined. And suppose the measurements are the true outputs plus measurement noises, i.e., $f_{\tilde{y}}(y, e) = y + e$. When the functions are nonlinear, the optimization problem (2.5) is difficult to solve. To simplify, the functions are linearized around a nominal optimal operating point and a local solution is pursued. Let the nominal operating point be $(u, d, e, y, \tilde{y}, z, c) = (u^*, d^*, e^*, y^*, \tilde{y}^*, z^*, c^*)$. Define the deviation variables: $\Delta u = u - u^*$, $\Delta d = d - d^*$, $\Delta e = e - e^*$, $\Delta y = y - y^*$, $\Delta \tilde{y} = \tilde{y} - \tilde{y}^*$, $\Delta c = c - c^*$ and $\Delta z = z - z^*$. The linearized functions are obtained as

$$\Delta y = G_y \Delta u + G_y^d \Delta d, \tag{2.6}$$

$$\Delta \tilde{y} = \Delta y + \Delta e, \tag{2.7}$$

$$\Delta c = H \Delta \tilde{y} = 0. \tag{2.8}$$

$$\Delta z = G_z \Delta u + G_z^d \Delta d, \tag{2.9}$$

where $G_y := \partial f_y / \partial u$, $G_y^d := \partial f_y / \partial d$, $G_z := \partial f_z / \partial u$ and $G_z^d := \partial f_z / \partial d$, which are derivatives evaluated at the nominal point.

Define the loss function as $L(u, d) = J(u, d) - J(u^{opt}, d)$, which can be rewritten as

$$\begin{aligned} L(u, d) &= (J(u, d) - J(u^*, d)) - (J(u^{opt}, d) - J(u^*, d)) \\ &= \Delta J(u, d) - \Delta J(u^{opt}, d), \end{aligned} \quad (2.10)$$

where the point (u^{opt}, d) is a moving optimal point which solves the ideal online optimal control problem. Approximate $\Delta J(u, d)$ and $\Delta J(u^{opt}, d)$ respectively by its second order Taylor expansions and obtain a second-order approximation of the loss function as

$$L(u, d) = \frac{1}{2} (\Delta u - \Delta u^{opt})^T J_{uu} (\Delta u - \Delta u^{opt}), \quad (2.11)$$

where $J_{uu} = \partial^2 J / \partial u^2$ as evaluated at the nominal point.

Let $\Delta d = W_d \bar{d}$ and $\Delta e = W_e \bar{e}$, where the diagonal matrices W_d and W_e contain the expected magnitudes of disturbances and measurement errors, respectively. With the relations in (2.6)-(2.8) and the relation $\Delta u^{opt} = -J_{uu}^{-1} J_{ud} \Delta d$ [34, 40], the loss is explicitly obtained as

$$\bar{L}(\bar{d}, \bar{e}) := L(u, d) = \frac{1}{2} \left\| J_{uu}^{1/2} (H G_y)^{-1} H Y \begin{bmatrix} \bar{d} \\ \bar{e} \end{bmatrix} \right\|_2^2, \quad (2.12)$$

where

$$Y = [F W_d \quad W_e], \quad F = \frac{\partial y_{opt}}{\partial d} = G_y^d - G_y J_{uu}^{-1} J_{ud}. \quad (2.13)$$

Note that $H G_y$ is assumed to be nonsingular, which ensures the setpoints be attainable by manipulating the inputs. By assuming that Δd and Δe have zero means, both $\bar{d}$ and $\bar{e}$ have zero means. Let $\bar{d} \in \bar{\mathcal{D}}$ and $\bar{e} \in \bar{\mathcal{E}}$, where $\bar{\mathcal{D}}$ and $\bar{\mathcal{E}}$ are normalized domains corresponding to $\mathcal{D}$ and $\mathcal{E}$, respectively. As a result, the local SOC problem becomes to solve

$$\begin{aligned}
& \min_H \mathbb{E} \left(\frac{1}{2} \left\| J_{uu}^{1/2} (H G_y)^{-1} H Y \begin{bmatrix} \bar{d} \\ \bar{e} \end{bmatrix} \right\|_2^2 \right) \\
& \text{s.t., } \Delta z = G_z \Delta u + G_z^d W_d \bar{d} \leq 0, \\
& \bar{d} \in \bar{\mathcal{D}}, \bar{e} \in \bar{\mathcal{E}}, H \in \mathcal{H},
\end{aligned} \tag{2.14}$$

where Δu can explicitly be expressed by $\bar{d}$ and $\bar{e}$ due to (2.6)-(2.8). Therefore the optimal measurement combination matrix (H^*) is solved from (2.14) and it determines the CVs as $H^* \tilde{y}$.

The formulation of SOC in (2.14) for a steady-state process is very general. Recent studies on SOC can all be viewed as investigating (2.14) within particular domains of $\bar{\mathcal{D}}$ and $\bar{\mathcal{E}}$ and with/without the operational constraints in terms of Δz , where the objective function may be replaced by the worst-case cost function [14, 34, 40, 43-44, 46]. In addition, structural constraints on the MCM (H) may be considered, as indicated by the admissible domain $\mathcal{H}$, for practical SOC, which constitutes part of most recent investigations on SOC [45, 51-54].

Chapter 3

PID Controller Tuning with Specified GPMs for IPTD Processes

In this chapter, an *almost* closed-form solution is obtained for the problem of PID controller tuning with specified GPMs for an IPTD process. The solution indicates a general form of the PID parameters and unifies a large number of existing rules as PID controller tuning with various GPM specifications. Meanwhile, accurate expressions are also obtained for estimating the GPMs attained by a given PID controller. The GPMs realized by existing PID tuning rules are computed and documented as a reference for control engineers to tune the PID controllers.

3.1 Introduction

PID control has been widely applied in industry — more than 90% of the applied controllers are PID controllers [3, 21, 55-56]. In the absence of the derivative action, PI control is also broadly deployed, since in many cases the derivative action cannot significantly enhance the performance or may not be appropriate for noisy environment [3, 21, 55-56]. Another special form of PID control without the integral action, PD control is also applied [3, 21, 55-56]. Unlike the previous two cases, however, PD control cannot achieve zero steady-state error subject to load disturbances, which limits its applications [3, 21, 55-56].

Tuning PI/PD/PID controllers for IPTD processes has attracted a lot of attention, dating back to 1940s and lasting even today [6, 10, 55, 57-66]. Lots of results have been accumulated. There are more than fifty PI/PD/PID tuning rules for IPTD processes according to a survey made by O'Dwyer [55]. The actual number is even much higher [10, 57-58, 66-67]. Close observations reveal that many of these rules are sharing a common form. Such observations motivate our exploration of a general solution for the PI/PD/PID controller tuning on an IPTD process in this chapter.

Tuning PI/PD/PID controllers based on GPM specifications has been extensively studied in the literature [29, 55, 62, 68-72]. However, general analytic solutions of the controller parameters are not available, because of nonlinearity and solvability of such problems. Most existing solutions are limited by assuming certain constraints on GPMs or by approximations that are valid only for certain regions of process parameters [55, 62, 68-70]. As two exceptions, the graphic method proposed in [71] can derive PI parameters from an intersection of two graphs that are plotted using the frequency response of a general process, and the method proposed in [72] is able to tune PID controllers for any linear processes if the phase cross-over frequency of the loop transfer function is specified properly. The two methods are applicable to IPTD processes. However, they do not give the PI/PID parameters in terms of process parameters and hence require case-to-case numerical solutions in face of different processes even if the GPMs are specified the same.

This chapter is devoted to solving the PI/PD/PID parameters for an IPTD process with specified GPMs. Different from the existing results, *nearly* closed-form solutions are obtained for the whole domain of the process parameters. Explicit PI/PD/PID tuning formulas are obtained in terms of the process parameters. The formulas are used to unify a large number of existing rules as PI/PD/PID controller tuning with various GPM specifications. As reverse solutions, expressions of the GPMs for given PI/PD/PID settings

of an IPTD process are also obtained. These GPM formulas estimate GPMs with high accuracy and are applied to estimate the GPMs attained by each relevant PI/PD/PID tuning rule collected in [55].

The rest of the chapter is organized as follows. In Section 3.2, the solution of PI/PD/PID parameters with specified GPMs and the reverse solution of GPMs with a given PI/PD/PID setting are derived. During the derivations, numerical evaluations are employed to validate any approximations involved. In Section 3.3, the derived PI/PD/PID formulas are applied to unify the existing rules as PI/PD/PID controller tuning with different GPM specifications, and the derived GPM formulas are applied to estimate the GPMs attained by existing rules. Finally, Section 3.4 concludes the chapter.

3.2 Derivation of the PI/PD/PID Tuning Formulas and the GPM Formulas

The ideal unity-feedback control system is considered, as shown in Figure 3.1, where $G_c(s)$ denotes a PI/PD/PID controller and $G_p(s)$ denotes an IPTD process. Specifically, the transfer functions are

$$G_p(s) = K_p e^{-\tau s} / s, \quad \tau > 0, \quad (3.1)$$

where K_p is the process gain and τ the time delay, and

$$G_c(s) = \begin{cases} K_c (1 + \frac{1}{sT_i}), & \text{PI controller;} \\ K_c (1 + T_d s), & \text{PD controller;} \\ K_c (1 + \frac{1}{sT_i} + T_d s), & \text{PID controller,} \end{cases} \quad (3.2)$$

where K_c , T_i and T_d are the proportional, integral and derivative parameters respectively.

With this closed-loop system, the PI/PD/PID parameters are solved for achieving specified

GPMs. While it depends on specific design requirements, the specification of GPMs is assumed to be given throughout the chapter.

Although PI and PD controller tunings are special cases of PID controller tuning, their tuning formulas and corresponding GPM formulas are derived independently, adopting different approximations for accuracy and simplicity.

Figure 3.1 Control system loop.

3.2.1 PI Tuning Formula and GPM-PI Formula

Suppose GPMs of the closed-loop system are specified as (A_m, ϕ_m) , where A_m denotes the gain margin and ϕ_m denotes the phase margin. Given a PI controller in (3.2), the PI parameters (K_c, T_i) are to be solved. According to the GPM definitions, we have

$$\arg[G(j\omega_p)] = -\pi + \arctan(\omega_p T_i) - \omega_p \tau = -\pi, \quad (3.3)$$

$$\frac{1}{A_m} = |G(j\omega_p)| = \frac{K_c K_p \sqrt{1 + \omega_p^2 T_i^2}}{\omega_p^2 T_i}, \quad (3.4)$$

$$1 = |G(j\omega_g)| = \frac{K_c K_p \sqrt{1 + \omega_g^2 T_i^2}}{\omega_g^2 T_i}, \quad (3.5)$$

$$\phi_m = \arg[G(j\omega_g)] + \pi = \arctan(\omega_g T_i) - \omega_g \tau, \quad (3.6)$$

where ω_p and ω_g are the phase and the gain crossover frequencies, respectively. Due to nonlinearity of the equations, the four variables ω_g , ω_p , K_c and T_i are normally analytically unsolvable, preventing derivation of a general PI tuning formula [55]. By

introducing two intermediate variables, however, these variables can be solved.

Specifically, let $\alpha := \omega_g T_i$ and $\beta := \omega_p T_i$. From (3.3)-(3.6), the solution is obtained as

$$\begin{cases} \omega_g = \frac{1}{\tau} (\arctan \alpha - \phi_m), \\ \omega_p = \frac{\arctan \beta}{\tau} = \frac{\beta}{\alpha} \omega_g, \\ K_c = \frac{\alpha \omega_g}{K_p \sqrt{1 + \alpha^2}}, \\ T_i = \alpha / \omega_g, \end{cases} \quad (3.7)$$

where (α, β) is solved from

$$\begin{cases} \phi_m = \arctan \alpha - \frac{\alpha}{\beta} \arctan \beta, \\ A_m = \frac{\beta^2}{\alpha^2} \sqrt{\frac{1 + \alpha^2}{1 + \beta^2}}. \end{cases} \quad (3.8)$$

The solution (α, β) is a constant pair corresponding to a specified GPM pair which can easily be solved using a numerical solver, e.g., the solver ‘fsolve’ in Matlab. The solution is unique, if there is any, since $\alpha > \tan \phi_m$ and $\beta > 0$ which ensure positive crossover frequencies and PI parameters. The initial guess of (α, β) for the numerical solver can be any pair of large enough positive numbers, e.g., $(2 \tan \phi_m, 2 \tan \phi_m)$, $(5, 5)$ (as used in the later numeric tests), etc.

Therefore (3.7) gives explicit expressions of the PI parameters (K_c, T_i) in terms of the process parameters (K_p, τ) . For convenience, (3.7) is called as *PI tuning formula*.

Note that the crossover frequencies ω_p and ω_g are also explicitly given in (3.7).

As an inverse problem, we compute the GPMs resulting from a given PI controller for an IPTD process. Still based on (3.3)-(3.6), the expression of GPMs, namely *GPM-PI formula*, is obtained as follows:

$$\begin{cases} \omega_g = \alpha/T_i, \\ \omega_p = \beta/T_i, \\ A_m = \frac{\beta^2}{\alpha^2} \sqrt{\frac{1+\alpha^2}{1+\beta^2}}, \\ \phi_m = \arctan \alpha - \omega_g \tau, \end{cases} \quad (3.9)$$

where

$$\alpha = \sqrt{\frac{\gamma^2}{2} \left(1 + \sqrt{1 + \frac{4}{\gamma^2}} \right)}, \text{ with } \gamma := K_p K_c T_i, \quad (3.10)$$

(the negative α is omitted) and β is solved from

$$\arctan \beta = \theta \beta, \text{ with } \theta := \tau/T_i. \quad (3.11)$$

Solution (3.9) also gives expressions of the gain and phase crossover frequencies. As indicated by the above equations, the phase margin ϕ_m is explicitly expressed; however, deriving the gain margin A_m requires first solving (3.11) for β . Although a numerical solution can be used, for ease of application an approximate analytic solution is proposed. According to Appendix A.1, such a solution is

$$\begin{cases} \beta = \frac{\pi}{4\theta} \left(1 + \sqrt{1 - \frac{16\lambda_B \theta}{\pi^2}} \right), & \text{if } 0 < \theta \leq \theta_B, \\ \beta = \frac{1}{2\theta} \sqrt{-5 + \sqrt{\frac{120}{\theta} - 95}}, & \text{if } \theta_B < \theta < 1, \end{cases} \quad (3.12)$$

where $\lambda_B = 0.917$ and $\theta_B = 0.582$. The constraint $0 < \theta < 1$ is imposed to ensure a positive solution for β . With β given in (3.12), both A_m and ω_p in (3.9) are then explicitly expressed. The above solution of (α, β) meanwhile justifies the uniqueness of the solution to (3.8).

To evaluate the accuracy of (3.12) as the solution of (3.11), numeric tests are carried out. Without loss of generality, let $K_p = 1$. For different (τ, A_m, ϕ_m) , the PI parameters

are first calculated by the PI tuning formula. With these PI parameters, the realized GPMs are then estimated by the GPM-PI formula, using β 's estimated by (3.12). The estimated GPMs are compared with the originally specified GPMs correspondingly, so that the accuracy of the approximations is tested. In the computation, the parameters are chosen randomly as $\tau \in (0, 1]$ (which loses no generality since the PI tuning formula and GPM-PI formula both apply regardless of the process parameters), $A_m \in (1, 12]$ and $\phi_m \in (10^\circ, 70^\circ]$. Fifty numerical tests were done and the results are shown in Figure 3.2, where the relative estimation error (R.e.e) is defined as $R.e.e. := (the\ estimated\ value - the\ true\ value) / the\ true\ value$. Since α and ϕ_m are exactly derived by the GPM-PI formula, they are omitted in the figure, which remains the same for later discussions on PD and PID controls. The results indicate that the estimation errors of A_m 's are normally within 2% and thus validate (3.9) adopting the approximate solution of β by (3.12).

Figure 3.2 GPMs estimated by GPM-PI formula versus true GPMs: the dots denote the estimated points and the circles denote the true points.

3.2.2 PD Tuning Formula and GPM-PD Formula

Given a GPM pair (A_m, ϕ_m) , an IPTD process in (3.1) and a PD controller in (3.2), the PD parameters (K_c, T_d) are to be solved. The definitions of GPMs lead to

$$\arg[G(j\omega_p)] = -\pi/2 + \arctan \omega_p T_d - \omega_p \tau = -\pi, \quad (3.13)$$

$$\frac{1}{A_m} = |G(j\omega_p)| = K_c K_p \sqrt{1 + \omega_p^2 T_d^2} / \omega_p, \quad (3.14)$$

$$1 = |G(j\omega_g)| = K_c K_p \sqrt{1 + \omega_g^2 T_d^2} / \omega_g, \quad (3.15)$$

$$\phi_m = \arg[G(j\omega_g)] + \pi = \pi/2 + \arctan \omega_g T_d - \omega_g \tau, \quad (3.16)$$

where the variables are defined the same as those in Section 3.2.1. By introducing two new variables $\alpha' := \omega_g T_d$ and $\beta' := \omega_p T_d$ in a similar way to that for the PI case, the parameters are solved from (3.13)-(3.16) that

$$\begin{cases} \omega_g = \frac{1}{\tau} (\arctan \alpha' + \frac{\pi}{2} - \phi_m), \\ \omega_p = \frac{1}{\tau} (\arctan \beta' + \frac{\pi}{2}) = \frac{\beta'}{\alpha'} \omega_g, \\ K_c = \frac{\omega_g}{K_p \sqrt{1 + \alpha'^2}}, \\ T_d = \alpha' / \omega_g, \end{cases} \quad (3.17)$$

where the constant pair (α', β') is solved from the equations

$$\begin{cases} \phi_m = \arctan \alpha' + \frac{\pi}{2} - \frac{\alpha'}{\beta'} (\arctan \beta' + \frac{\pi}{2}), \\ A_m = \frac{\beta'}{\alpha'} \sqrt{\frac{1 + \alpha'^2}{1 + \beta'^2}}. \end{cases} \quad (3.18)$$

The solution (α', β') is unique since $\alpha' > 0$ and $\beta' > 0$ which make sure positive crossover frequencies and PD parameters. The initial guess of (α', β') for the numerical solver can be any pair of large enough positive numbers, e.g., (5, 5), (10, 10), etc.

Therefore, (3.17) gives the *PD tuning formula*.

Inversely, given an IPTD process in (3.1) and a PD controller in (3.2), the resultant GPMs and crossover frequencies of the closed-loop system are derived from (3.13)-(3.16) as

$$\begin{cases} \omega_g = \alpha'/T_d, \\ \omega_p = \beta'/T_d, \\ A_m = \frac{\beta'}{\alpha'} \sqrt{\frac{1+\alpha'^2}{1+\beta'^2}}, \\ \phi_m = \arctan \alpha' - \omega_g \tau + \pi/2, \end{cases} \quad (3.19)$$

where

$$\alpha' = \sqrt{\gamma'^2/(1-\gamma'^2)}, \text{ with } \gamma' := K_p K_c T_d, \quad (3.20)$$

and β' is solved from

$$\arctan \beta' = \theta' \beta' - \pi/2, \text{ with } \theta' := \tau/T_d. \quad (3.21)$$

Since deriving the gain margin requires solving β' from (3.21), an approximate analytic solution is proposed for it. Divide the domain of β' into two: $0 < \beta' \leq 1$ (β' being small) and $\beta' > 1$ (β' being large). In the former domain, use the approximation

$$\arctan \beta' \approx \lambda \beta', \text{ with } \lambda := \pi/4, \quad (3.22)$$

and in the latter domain use the approximation

$$\arctan \beta' = \frac{\pi}{2} - \arctan \frac{1}{\beta'} \approx \frac{\pi}{2} - \frac{\lambda}{\beta'}. \quad (3.23)$$

Solve (3.22) and (3.23) respectively, and express the applicable domains in terms of θ' , an approximate solution of (3.21) is derived as

$$\begin{cases} \beta' = \frac{\pi}{2\theta'} \left(1 + \sqrt{1 - \frac{4\theta'\lambda}{\pi^2}} \right), & \text{if } 0 < \theta' \leq \theta'_B, \\ \beta' = \frac{\pi}{2(\theta' - \lambda)}, & \text{if } \theta' > \theta'_B, \text{ where } \theta'_B := \pi/2 + \lambda. \end{cases} \quad (3.24)$$

Therefore, (3.19) gives the *GPM-PD formula*, where the intermediate variables α' and β' are expressed in (3.20) and (3.24) respectively. Meanwhile the solution of (α', β') justifies the uniqueness of the solution to (3.18) for given GPMs.

To evaluate the accuracy of (3.24) as a solution of (3.21), numeric computations are carried out to test it. The IPTD process parameters and the GPMs are specified in a similar way to those for the PI case (refer to Section 3.2.1). Analogously, the results of 50 random tests are obtained and shown in Figure 3.3, which demonstrate the accuracy of the GPM-PD formula adopting β' estimated by (3.24).

Figure 3.3 GPMs estimated by GPM-PD formula versus true GPMs: the dots denote the estimated points and the circles denote the true points.

3.2.3 PID Tuning Formula and GPM-PID Formula

Given a GPM pair (A_m, ϕ_m) , an IPTD process in (3.1) and a PID controller in (3.2), the PID parameters (K_c, T_i, T_d) are to be solved. The definitions of GPMs lead to

$$-\pi = \arg[G(j\omega_p)] = -\pi + \arctan \frac{\omega_p T_i}{1 - \omega_p^2 T_i T_d} + \mathcal{H}(1 - \omega_p^2 T_i T_d)\pi - \omega_p \tau, \quad (3.25)$$

$$\frac{1}{A_m} = |G(j\omega_p)| = \frac{K_c K_p \sqrt{(1 - \omega_p^2 T_i T_d)^2 + \omega_p^2 T_i^2}}{\omega_p^2 T_i}, \quad (3.26)$$

$$1 = |G(j\omega_g)| = \frac{K_c K_p \sqrt{(1 - \omega_g^2 T_i T_d)^2 + \omega_g^2 T_i^2}}{\omega_g^2 T_i}, \quad (3.27)$$

$$\phi_m = \arg[G(j\omega_g)] + \pi = \arctan \frac{\omega_g T_i}{1 - \omega_g^2 T_i T_d} + \mathcal{H}(1 - \omega_g^2 T_i T_d)\pi - \omega_g \tau, \quad (3.28)$$

where the function $\mathcal{H}(\bullet)$ is defined as

$$\mathcal{H}(t) := \begin{cases} 0, & \text{if } t \geq 0, \\ 1, & \text{if } t < 0. \end{cases} \quad (3.29)$$

Since there are five unknowns $(\omega_g, \omega_p, K_c, T_i, T_d)$, but only four equations, one additional condition is required for a unique solution. In the literature, normally it assumes $T_d = kT_i$ and $k \in (0, 0.5]$ [3, 55]. By defining α and β the same as those in Section 3.2.1, the parameters are solved from (3.25)-(3.28) that

$$\begin{cases} \omega_g = \frac{1}{\tau} (\arctan \frac{\alpha}{1 - k\alpha^2} + \mathcal{H}(1 - k\alpha^2)\pi - \phi_m), \\ \omega_p = \frac{1}{\tau} (\arctan \frac{\beta}{1 - k\beta^2} + \mathcal{H}(1 - k\beta^2)\pi) = \frac{\beta}{\alpha} \omega_g, \\ K_c = \frac{\alpha \omega_g}{K_p \sqrt{(1 - k\alpha^2)^2 + \alpha^2}}, \\ T_i = \alpha / \omega_g, \\ T_d = kT_i. \end{cases} \quad (3.30)$$

where (α, β) is solved from the following equations

$$\begin{cases} \phi_m = \arctan \frac{\alpha}{1-k\alpha^2} + \mathcal{H}(1-k\alpha^2)\pi \\ \quad - \frac{\alpha}{\beta} \left(\arctan \frac{\beta}{1-k\beta^2} + \mathcal{H}(1-k\beta^2)\pi \right), \\ A_m = \frac{\beta^2}{\alpha^2} \sqrt{\frac{(1-k\alpha^2)^2 + \alpha^2}{(1-k\beta^2)^2 + \beta^2}}. \end{cases} \quad (3.31)$$

The solution (α, β) is unique for ensuring positive crossover frequencies and PID parameters subject to a given k . This is justified by an explicit solution of (α, β) in terms of the PID parameters as presented later. The initial guess of (α, β) for a numerical solver to solve (3.31) can be any pair of large enough positive numbers, e.g., (5, 5), (10, 10), etc.

Equation (3.30) is the *PID tuning formula*. Note that when solving (3.31), depending on the value of k , four different cases need to be considered: 1) $1-k\alpha^2 > 0$, $1-k\beta^2 > 0$; 2) $1-k\alpha^2 > 0$, $1-k\beta^2 < 0$; 3) $1-k\alpha^2 < 0$, $1-k\beta^2 > 0$; and 4) $1-k\alpha^2 < 0$, $1-k\beta^2 < 0$. If none of these cases gives a solution, we may take (3.31) as having no solution for (α, β) and the GPMs should be re-specified to other values; or an alternative solution can be obtained such that the attained GPMs are in a certain sense (e.g., the least square sense) closest to the specified one.

Inversely, given an IPTD process in (3.1) and a PID controller in (3.2), the resultant GPMs and crossover frequencies of the closed-loop system are derived from (3.25)-(3.28) as

$$\begin{cases} \omega_g = \alpha/T_i, \\ \omega_p = \beta/T_i, \\ A_m = \frac{\beta^2}{\alpha^2} \sqrt{\frac{(1-k\alpha^2)^2 + \alpha^2}{(1-k\beta^2)^2 + \beta^2}}, \\ \phi_m = \arctan \frac{\alpha}{1-k\alpha^2} + \mathcal{H}(1-k\alpha^2)\pi - \omega_g \tau, \end{cases} \quad (3.32)$$

where α and β are the respective solutions of the two equations:

$$(\gamma^2 k^2 - 1)\alpha^4 + \gamma^2(1 - 2k)\alpha^2 + \gamma^2 = 0, \text{ and} \quad (3.33)$$

$$\arctan \frac{\beta}{1 - k\beta^2} + \mathcal{H}(1 - k\beta^2)\pi = \theta\beta, \quad (3.34)$$

where γ and θ are defined in (3.10) and (3.11). Equations (3.33)-(3.34) can be solved numerically. Alternatively, their approximate solutions can be obtained as below.

For (3.33), noticing the common conditions that $k \leq 0.5$ and $\gamma k < 1$ as adopted by a large number of existing rules [55], its unique solution (the negative solution is omitted) is obtained as

$$\alpha = \sqrt{\frac{\gamma^2}{2(1 - \gamma^2 k^2)} \left(1 - 2k + \sqrt{1 - 4k + \frac{4}{\gamma^2}} \right)}. \quad (3.35)$$

When $k = 0$, this solution reduces to (3.10), namely the solution for the case of PI control.

For (3.34), according to Appendix A.2, an approximate solution is obtained as

$$\beta = \begin{cases} \sqrt{\frac{1}{2} \left[\frac{1}{k} - \frac{3}{\theta^2} + \sqrt{\left(\frac{1}{k} + \frac{3}{\theta^2} \right)^2 - \frac{12}{k\theta^3}} \right]}, & \text{if } \beta < \beta_B; \\ \frac{\pi}{4(\theta - \lambda_B k)} \left(1 + \sqrt{1 - \frac{16\lambda_B(\theta - \lambda_B k)}{\pi^2}} \right), & \text{if } \beta_B \leq \beta < 1/\sqrt{k}; \\ \frac{\pi}{4(\theta - \lambda'_B k)} \left(1 + \sqrt{1 - \frac{16\lambda'_B(\theta - \lambda'_B k)}{\pi^2}} \right), & \text{if } 1/\sqrt{k} < \beta \leq \beta'_B; \\ -a_2/3 + U, & \text{if } \beta > \beta'_B, \end{cases} \quad (3.36)$$

where

$$\begin{aligned} \lambda_B &:= \lambda(1/x_B), \quad \beta_B := (\sqrt{1 + 4kx_B^2} - 1)/(2kx_B), \\ \lambda'_B &:= \lambda(1/x'_B), \quad \beta'_B := (\sqrt{1 + 4kx_B'^2} + 1)/(2kx'_B), \end{aligned} \quad (3.37)$$

with $x_B := 1.5$, $x'_B := 1.0$ and $\lambda(t) := (\arctan t)/t$; and

$$\begin{cases} U := \sqrt[3]{R + \sqrt{D}} + \sqrt[3]{R - \sqrt{D}}, & \text{if } D \geq 0; \\ U := 2\sqrt[6]{R^2 - D} \cos(\varphi/3), \\ \text{with } \varphi := \arctan(\sqrt{-D}/R) + \mathcal{H}(R)\pi, & \text{if } D < 0, \end{cases} \quad (3.38)$$

with

$$\begin{aligned} D &:= Q^3 + R^2, \quad Q := (3a_1 - a_2^2)/9, \\ R &:= (9a_2a_1 - 27a_0 - 2a_2^3)/54, \\ a_0 &:= \pi/(k\theta), \quad a_1 := (\lambda'_B - \theta)/(k\theta), \quad a_2 := -\pi/\theta. \end{aligned} \quad (3.39)$$

To summarize, (3.32) gives the *GPM-PID formula*, with the intermediate variables α and β being expressed by (3.35) and (3.36) respectively. By the way, the solution of (α, β) justifies the uniqueness of the solution to (3.31) for given GPMs.

Remark 3.1. a) Since the boundary conditions in (3.36) are implicit, the candidate solutions are calculated in turn until a valid one is obtained. b) Refer to the end of Appendix A.2 for a less accurate yet simpler approximate solution of (3.34).

Figure 3.4 GPMs estimated by GPM-PID formula versus true GPMs: the dots denote the estimated points and the circles denote the true points.

Figure 3.5 Relative estimation errors of the results in Figure 3.4.

Numerical computations are carried out to evaluate the accuracy of (3.36) as the solution of (3.34). The IPTD process parameters and the GPMs are specified in a similar way to those for the PI case (see Section 3.2.1). Numerical results of 50 random tests are obtained for different values of k , respectively, as shown in Figure 3.4 and Figure 3.5. Since the estimation errors are normally within 5%, the results validate the calculation of A_m in GPM-PID formula based on the β approximated by (3.36).

3.3 Application to Unifying the Existing Tuning Rules

Rules of tuning PI/PD/PID controllers for an IPTD process have been accumulated in the past decades. These rules are based on various requirements and specifications on performance and robustness of the closed-loop system and were derived with various methods [55]. However, most of them can be unified by the tuning formulas presented above. From the PI, PD, PID tuning formulas respectively in (3.7), (3.17), and (3.30), we

see that the PID parameters have a common form of

$$K_c = \frac{k_1}{K_p \tau}, \quad T_i = k_2 \tau, \quad T_d = k_3 \tau, \quad (3.40)$$

where the parameters k_1, k_2, k_3 are specifically

$$\begin{aligned} \text{PI controller: } k_1 &= \frac{\alpha(\arctan \alpha - \phi_m)}{\sqrt{1 + \alpha^2}}, \quad k_2 = \frac{\alpha}{\arctan \alpha - \phi_m}, \quad k_3 = 0; \\ \text{PD controller: } k_1 &= \frac{\arctan \alpha' + \pi/2 - \phi_m}{\sqrt{1 + \alpha'^2}}, \quad k_2 = \infty, \quad k_3 = \frac{\alpha'}{\arctan \alpha' + \pi/2 - \phi_m}; \\ \text{PID controller: } k_1 &= \frac{\alpha \varphi_\alpha}{\sqrt{(1 - k\alpha^2)^2 + \alpha^2}}, \quad k_2 = \frac{\alpha}{\varphi_\alpha}, \quad k_3 = k k_2. \end{aligned} \quad (3.41)$$

Here $\varphi_\alpha := \arctan \frac{\alpha}{1 - k\alpha^2} + \mathcal{H}(1 - k\alpha^2)\pi - \phi_m$, and α , α' and α for the PI, PD and PID controllers are determined from (3.8), (3.18) and (3.31) respectively.

The common form of PI/PD/PID parameters in (3.40) indicates that different rules employing different values of (k_1, k_2, k_3) are realizing different GPMs which consequently lead to various closed-loop performances. This gives a unified interpretation to the vast variety of PI/PD/PID tuning rules accumulated in the literature [55]. From this viewpoint, PI/PD/PID control design on an IPTD process is essentially choosing a proper GPM pair or parameter set (k_1, k_2, k_3) . The GPM pair or parameter set can be selected via performance optimization subject to design constraints. Depending on the specific performance index and design constraints, the solution may differ from case to case and particular studies are required. A summary of various designs can be found in [55]. In particular, the well-known SIMC rule [6] uses GPMs of about $(3.0, 46.9^\circ)$ and the improved SIMC rule (with enhanced disturbance rejection) [66] about $(2.9, 42.5^\circ)$ for an IPTD process, when the recommended settings are adopted for both methods.

Finally, we apply the GPM-PI/PD/PID formulas derived in the last section to estimate the GPMs realized by relevant PI/PD/PID tuning rules as collected in [55]. The

GPM-PI/PD/PID formulas indicate that any PI/PD/PID controllers with the same (k_1, k_2, k_3) in (3.40) result in the same GPMs, regardless of the process parameters. This enables numeric computation of the exact GPMs realized by each rule in the form of (3.40). To compare, GPMs attained by each rule is computed by using both the GPM-PI/PD/PID formula and the numeric approach. The results are documented in the link [73], which take more than four pages to present and hence are omitted here. The results show that various GPMs are achieved by the existing tuning rules. Note that the larger the gain margin or the smaller the phase margin is, the more aggressive yet less robust the closed-loop performance will be. The summary of such GPMs thus provides a rich reference for control engineers to tune PID controllers. Meanwhile the results verify that the GPM-PI/PD/PID formulas are accurate for GPM estimations.

3.4 Conclusions

For an IPTD process, PI/PD/PID tuning formulas with specified GPMs were obtained and so were GPM-PI/PD/PID formulas for estimating GPMs resulting from a given PI/PD/PID controller. The tuning formulas indicate a common form of the PID parameters and unify a large number of tuning rules as PI/PD/PID controller tuning with various GPM specifications. The GPM formulas accurately estimate the GPMs realized by each relevant PI/PD/PID tuning rule as collected in [55] and the results are summarized in the link [73]. The results show that a variety of GPMs are attained by the existing rules. Since the rules were developed based on various criterion and methods, the summary of their resulting GPMs provides a rich reference for control engineers to tune PID controllers, helping select a rule or a GPM pair for a specific design.

Chapter 4

Simple Analytical PID Tuning Rules

In this chapter we analytically derive simple PID tuning rules based on typical process models. With the PI tuning formula obtained in Chapter 3, a tuning rule is first obtained for IPTD processes by making the approximate damping ratio of the closed-loop system be one. Based on this rule, simple tuning rules are then obtained for other typical process models used in process control. Compared to the SIMC counterparts, the new rules lead to either the same or better disturbance rejection while achieving the same peak sensitivities.

4.1 Introduction

Despite a wealth of research on PID controller tuning, surveys show that many of the industrial PID controllers are poorly tuned and many of them use default factory settings without any specific tuning at all [1, 3-4]. This implies a gap between research and applications. A tacit reason for such gap is that simple, efficient and reliable PID tuning rules are still lacking. This motivated the proposals of PID tuning rules in [6, 27, 74-75].

Internal model control (IMC) was used to derive simple PID tuning rules for typical processes [74-75]. However, the rules give sluggish load response when a process is lag dominated (i.e., the process lag-delay ratio is large), due to zero-pole cancellation involved in deriving such PID tuning rules [6]. To solve this problem, Skogestad proposed a method for revising the integral parameter properly [6]. The resulting SIMC tuning rules keep simple in form but give improved performance when a process is lag dominated. They are

demonstrated to achieve robust and competitive performance compared to existing tuning rules while SIMC rules have a unique advantage of being very simple [6, 28].

With the results in last chapter and inspired by SIMC rules, this chapter is devoted to analytically deriving new simple PID tuning rules. This is achieved by making the closed-loop system achieve an approximate damping ratio of one. The rationale will be explained in detail. Compared to the derivation of SIMC rules, the new derivation adopts a higher order approximation of the time delay component in the process model. The new rules turn out to be able to achieve either the same or better disturbance rejection while achieving the same peak sensitivities, as compared to the SIMC counterparts. This is demonstrated by various numerical examples.

4.2 Derivation of the PID Tuning Rules

This section derives a simple PI tuning rule for IPTD processes and the derivation is then extended to FOPTD, SOPTD, ILPTD, DIPTD, and pure TD processes. The feedback control system is shown in Figure 4.1, where u is the manipulated control input, d the disturbance, y the controlled output, y_s the setpoint (reference) for the controlled output, $c(s)$ the PI/PID controller transfer function, and $g(s)$ the process transfer function.

Figure 4.1 Block diagram of feedback control system.

The PID controller takes the form of

$$c(s) = K_c \left(1 + \frac{1}{\tau_I s} \right) (1 + \tau_D s), \quad (4.1)$$

where K_c , τ_I and τ_D are the P, I and D parameters respectively. When $\tau_D = 0$, $c(s)$ corresponds to a PI controller. Here the PID controller in series form is used for simple forms of PID tuning formulas when the derivative action is included. For convenience, corresponding settings of the ideal PID controller are given as follows:

$$c(s) = K'_c \left(1 + \frac{1}{\tau'_I s} + \tau'_D s \right), \quad (4.2)$$

where

$$K'_c = K_c \left(1 + \frac{\tau_D}{\tau_I} \right), \quad \tau'_I = \tau_I \left(1 + \frac{\tau_D}{\tau_I} \right), \quad \tau'_D = \frac{\tau_D}{1 + \tau_D / \tau_I}, \quad (4.3)$$

which are the P, I and D gains respectively.

4.2.1 The Case of an IPTD Process

Consider an IPTD process

$$g(s) = k e^{-\theta s} / s, \quad (4.4)$$

where k is the process gain and θ the time delay. According to [76], in general the PI parameters are expressed as

$$K_c = \frac{k_1}{k\theta}, \quad \tau_I = k_2\theta, \quad (4.5)$$

where k_1 and k_2 are two tuning factors which uniquely determine the GPMs of the control system. Due to interlace between them, it is not easy to tune these two parameters properly. To overcome the difficulty, we propose an approach to expressing k_2 as an appropriate function of k_1 , leaving k_1 the only parameter to tune.

Given the PI controller in (4.1), the closed-loop transfer function is derived as

$$\tilde{g}(s) := \frac{g(s)c(s)}{1 + g(s)c(s)} = \frac{(k_2\theta s + 1)e^{-\theta s}}{\frac{k_2}{k_1}\theta^2 s^2 + (k_2\theta s + 1)e^{-\theta s}}. \quad (4.6)$$

Use Maclaurin expansion and approximate the numerator and denominator of $\tilde{g}(s)$ by the second-order polynomials, yielding

$$\tilde{g}(s) \approx \frac{\frac{-k_2 + 0.5}{(1/k_1 - 1)k_2 + 0.5} s^2 + \frac{k_2 - 1}{(1/k_1 - 1)k_2\theta + 0.5\theta} s + \frac{1}{(1/k_1 - 1)k_2\theta^2 + 0.5\theta^2}}{s^2 + \frac{k_2 - 1}{(1/k_1 - 1)k_2\theta + 0.5\theta} s + \frac{1}{(1/k_1 - 1)k_2\theta^2 + 0.5\theta^2}}. \quad (4.7)$$

Hence the characteristic polynomial of $\tilde{g}(s)$ is

$$f(s) := s^2 + \frac{k_2 - 1}{(1/k_1 - 1)k_2\theta + 0.5\theta} s + \frac{1}{(1/k_1 - 1)k_2\theta^2 + 0.5\theta^2}. \quad (4.8)$$

The polynomial $f(s)$ is in the standard second-order form, $s^2 + 2\zeta\omega_n s + \omega_n^2$, with

$$\omega_n = \frac{1}{\theta\sqrt{(1/k_1 - 1)k_2 + 0.5}}, \quad \zeta = \frac{k_2 - 1}{2\sqrt{(1/k_1 - 1)k_2 + 0.5}}, \quad (4.9)$$

where ζ has a physical meaning of the damping ratio [77]. Hence, ζ in (4.9) denotes an approximate damping ratio of the closed-loop system. Equation (4.9) solves k_2 as

$$k_2 = 1 - 2\zeta^2 + \frac{2\zeta^2}{k_1} + \sqrt{\left(1 - 2\zeta^2 + \frac{2\zeta^2}{k_1}\right)^2 + 2\zeta^2 - 1}. \quad (4.10)$$

Equation (4.10) indicates that the tuning parameter k_2 is an explicit function of k_1 and ζ . To release the tuning difficulty, ζ may be set as a proper constant so that k_1 is left as the only tuning parameter. According to Appendix B, a proper ζ is 1.0. Consequently, with $\zeta = 1.0$, the tuning parameter k_2 in (4.10) is simplified into

$$k_2 = 2/k_1 - 1 + \sqrt{(2/k_1 - 1)^2 + 1}, \quad (4.11)$$

which is a function singly of k_1 . Therefore, the PI tuning formula for an IPTD process is

expressed in (4.5) with k_2 being expressed as an explicit function of k_1 in (4.11).

In order to derive an *easy-to-memorize* rule, k_2 in (4.11) is approximated as $4/k_1 - 2$ (although there are more accurate alternates) with relative errors (as defined as ‘(approximate value – true value) / true value $\times 100\%$ ’) within $(-4.22\%, -0.31\%)$ for $0.2 \leq k_1 \leq 0.6$. (As will be shown later, it is sufficient to consider k_1 in the range of $[0.2, 0.6]$ so that the control system has a peak sensitivity within the range of $[1.2, 2.0]$ for robust control.) The errors of the approximation are shown in Figure 4.2, where the values of k_1 with a step of 0.001 are used in the computations. It indicates that the approximation errors are small. Therefore we use

$$\tau_I = k_2 \theta = \left(\frac{4}{k_1} - 2 \right) \theta, \quad (4.12)$$

in the PI tuning rule. Refer to Table 4.1 for the specific rule.

Figure 4.2 The true k_2 as $2/k_1 - 1 + \sqrt{(2/k_1 - 1)^2 + 1}$ v.s. its approximate as $4/k_1 - 2$.

Remark 4.1 *In the derivation of SIMC rules [6], the time delay component was ignored in deriving the characteristic polynomial. This leads to less accurate estimation of ζ as compared to the above. In consequence, the SIMC tuning rules achieve a damping ratio approximately of $\sqrt{0.5(k_1-4)^2/((k_1-4)^2-8)}$ which is dependent on k_1 . This, however, in general does not lead to better tradeoff between performance and robustness as will be shown by examples in Section 4.3.*

4.2.2 The Case of an FOPTD Process

Consider the PI control of an FOPTD process

$$g(s) = \frac{ke^{-\theta s}}{\tau_1 s + 1}. \quad (4.13)$$

The derivation is partitioned into two cases (as done in deriving the SIMC rule [6]): the delay dominated case and the lag dominated case. The basic idea is to convert the PI tuning into the one on an IPTD process which has been solved.

Case i: the FOPTD process being delay dominated. The I parameter is set as the process time constant, that is, $\tau_I = \tau_1$. In consequence, the open-loop transfer function becomes

$$g(s)c(s) = K_c g'(s) := K_c k' e^{-\theta s} / s, \quad (4.14)$$

where $k' := k/\tau_1$. This is equivalent to a P controller acting on an IPTD process $g'(s)$, with a P gain of K_c . For this P tuning problem, it is known that the closed-loop system is asymptotically stable if and only if $0 < K_c k' < 0.5\pi/\theta$ [3]. Hence, the P parameter K_c keeps the form in (4.5), with k being replaced by k' and k_1 satisfies $0 < k_1 < 0.5\pi$.

Case ii: the FOPTD process being lag dominated. The process is approximated as an IPTD process:

$$g(s) \approx k'e^{-\theta s}/s, \quad (4.15)$$

where k' is the same as that in (4.14). Hence, the PI tuning reduces to the one on an IPTD process as expressed in (4.15), which was solved in last subsection. Therefore, the PI parameters are that K_c given in (4.5), where the process gain k is replaced by k' and τ_I is given in (4.12).

Combining the above two cases, the PI tuning formula for an FOPTD process is summarized in Table 4.1. Note that, like the SIMC rules, the I parameter τ_I is taken as the minimum of the above two cases of settings for non-conservative tuning, which avoids an explicit dividing boundary for the above two cases.

Remark 4.2 *The above two-case considerations are motivated by the observation that the zero-pole cancellation using $\tau_I = \tau_1$ is only efficient for the delay dominated case, whereas it leads to sluggish load response in the lag dominated case [3, 6]. This observation can be briefly explained as follows. Suppose exact zero-pole cancellation happens between the PI controller and the FOPTD process. Then the sensitivity function, $1/(1 + g(s)c(s))$, is invariant for given time delay θ and PI parameters, independent of the process time constant τ_1 . Consequently, the disturbance-to-output transfer function, $g_{dy}(s) = g(s)/(1 + g(s)c(s))$, will have its frequency response being proportional to that of $g(s)$. This implies that the load response will become more sluggish as the process time constant τ_1 increases, as observed.*

4.2.3 The Case of an SOPTD Process

Consider an SOPTD process

$$g(s) = \frac{ke^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)}, \quad \tau_1 > \tau_2. \quad (4.16)$$

Let the PID controller be given in (4.1). Like SIMC, set $\tau_D = \tau_2$. The resultant loop transfer function becomes

$$g(s)c(s) = K_c \left(1 + \frac{1}{\tau_I s} \right) \frac{ke^{-\theta s}}{\tau_1 s + 1}, \quad (4.17)$$

which is equivalent to the loop transfer function of a PI controller cascaded with an FOPTD process. Hence, the PID tuning mathematically reduces to the PI tuning on an FOPTD process. The P and I parameters are therefore referred to those obtained in last subsection. See Table 4.1 for a summary.

4.2.4 Other Processes

The PID controller tunings of other processes, such as ILPTD and DIPTD processes, can be solved by taking certain limits of the PID tuning rule for an SOPTD process.

First, consider the PI controller tuning of an ILPTD process given in Table 4.1. By perceiving the ILPTD process as an SOPTD process with $\tau_1 \rightarrow \infty$, the PID tuning formula is obtained by taking the limit as $\tau_1 \rightarrow \infty$ in the PID tuning rule for an SOPTD process.

Similarly, a DIPTD process can be viewed as an SOPTD with $\tau_{1,2} \rightarrow \infty$. The PID tuning rule is derived by taking the limits. It turns out that the P and D parameters are obtained as those in Table 4.1 while the I parameter approaches zero. This controller gives good setpoint response for the DIPTD process, but results in steady-state error for load disturbances occurring at the process input. To remove this offset, the I parameter τ_I is revised to be expressed in (4.11), as is similarly done in SIMC [6].

Finally, consider a pure TD process. Simply an integral control is applied. That is, the controller is $c(s) = K_I/s$. The integral controller tuning on this process is then mathematically equivalent to the P controller tuning on an IPTD process as discussed in

Case *i* of Section 4.2.1, where K_c in (4.14) means K_I here. Consequently the integral tuning formula is obtained and given in Table 4.1.

4.2.5 Choice of the Parameter k_1

In general a larger k_1 leads to more aggressive setpoint response and better disturbance rejection yet less robustness. An appropriate value of k_1 should be chosen for desired tradeoff between closed-loop performance and robustness. This can be done by either tuning the parameter k_1 directly or determining the value of k_1 based on GPM or peak sensitivity specification.

Tuning k_1 Directly. According to the analysis in Appendix B, the parameter k_1 can be tuned up and down in the range of 0.1 to 1.0, or more practically 0.2 to 0.6, until a satisfactory tradeoff between performance and robustness is attained. An initial value of k_1 can be set as 0.5 which achieves a peak sensitivity of 1.765 and a peak complementary sensitivity of 1.427 when the proposed controller is applied to an IPTD process. With this particular choice, the PID tuning rule for an SOPTD process is obtained explicitly as

$$K_c = \frac{\tau_1}{2k\theta}, \quad \tau_I = \min\{\tau_1, 6\theta\}, \quad \tau_D = \tau_2. \quad (4.18)$$

The closed-loop system approximately attains a gain margin (GM) of 3.14 and a phase margin (PM) of 61.35° if $\tau_1 \leq 6\theta$, and GM of 2.91 and PM of 42.32° if $\tau_1 > 6\theta$ (which can be computed numerically or using the GPM formulas derived in Chapter 3). These are better than the typical minimum requirements $\text{GM} > 1.7$ and $\text{PM} > 30^\circ$ [6, 78]. Meanwhile, in both cases the closed-loop setpoint response approximately has an overshoot of 25%, a rise time of 2.5θ and a peak time of 5θ (see Appendix B). Note that these performance indices are independent of process parameters due to the scalability of the PID tuning rule.

Tuning k_1 Based on GPM Specification. GPMs are known to reflect the system performance and robustness [3, 69-70]. According to the results in Chapter 3, the parameter k_1 can be tuned such that the control system achieves specified GPMs. (Since there is only one degree of freedom, it is impossible to achieve flexible GM and PM simultaneously unless the two margins have certain special relations.)

Given an IPTD process in (4.4) and a PI controller in (4.1) with its parameters being expressed in (4.5), a formula accurately estimating the GPMs of a control system is given as

$$\begin{cases} A_m = \frac{\beta^2}{\alpha^2} \sqrt{\frac{1+\alpha^2}{1+\beta^2}}, \\ \phi_m = \arctan \alpha - \frac{\alpha}{k_2}, \end{cases} \quad (4.19)$$

where

$$\begin{aligned} \alpha &= \sqrt{\frac{(k_1 k_2)^2}{2} \left(1 + \sqrt{1 + \frac{4}{(k_1 k_2)^2}} \right)}, \\ \beta &= \begin{cases} \frac{\pi k_2}{4} \left(1 + \sqrt{1 - \frac{16\lambda_B}{\pi^2 k_2}} \right), & \text{if } k_2 \geq k_2^B, \\ \frac{k_2}{2} \sqrt{-5 + \sqrt{120k_2 - 95}}, & \text{if } 1 < k_2 < k_2^B, \end{cases} \end{aligned} \quad (4.20)$$

with $\lambda_B = 0.917$ and $k_2^B = 1/0.582 \approx 1.718$.

(The condition $k_2 > 1$ is necessary to ensure the existence of a solution of GPMs in (4.19).) With the proposed PI/PID tuning rules, the PI/PID control systems of the aforementioned processes (except the DIPTD process) can all be viewed as being equivalent to certain PI control systems of IPTD processes. Since the PI parameters are in the general form of (4.5), the above formulas can be applied to estimate the GPMs of the PI/PID control systems attained. For the PI control of an IPTD process, the application is straightforward by replacing k_2 in (4.20) with $4/k_1 - 2$.

Consider the PI control of an FOPTD process. Two cases are treated separately: *case i*, $\tau_I = \tau_1$; and *case ii*, $\tau_I = (4/k_1 - 2)\theta$.

In *case i*, the open-loop transfer function is given in (4.14). Thus the definitions of GPMs give

$$\begin{cases} -0.5\pi - \omega_p \theta + \pi = 0, \\ \frac{1}{A_m} = |g(j\omega_p)c(j\omega_p)| = \frac{K_c k'}{\omega_p}, \\ -0.5\pi - \omega_g \theta + \pi = \phi_m, \\ 1 = |g(j\omega_g)c(j\omega_g)| = \frac{K_c k'}{\omega_g}, \end{cases} \quad (4.21)$$

where ω_p and ω_g are the phase and the gain crossover frequencies respectively. Given K_c in Table 4.1, the equations in (4.21) solve

$$A_m = \frac{\pi}{2k_1}, \quad \phi_m = \frac{\pi}{2} \left(1 - \frac{1}{A_m} \right) = \frac{\pi}{2} - k_1. \quad (4.22)$$

Relation (4.22) indicates that the tuning parameter k_1 directly determines the GPMs of the control system. Hence, by using (4.22), k_1 can be selected for achieving desired GPMs. For a pure TD process given in Table 4.1, the GPMs are expressed the same in (4.22) and hence the tuning of k_1 is the same.

In *case ii*, the PI control of an FOPTD process can be approximated as the PI control of an IPTD process with parameters of (k', θ) (refer to (4.15)). Consequently, given k_1 , the GPMs of the system are estimated by (4.19)-(4.20), where $k_2 = 4/k_1 - 2$.

Numerically, the relations between the GPMs and the tuning parameter k_1 for the above two cases are shown in Figure 4.3. The monotonic relations between k_1 and the margins justify the simple guideline presented in the last subsection. And it is interesting to observe that *case ii* leads to a similar relation as that in *case i*. This therefore enables

accurate approximation of the GPM- k_1 relation by an analytic formula. In summary, the analytic relations are established as

$$\begin{cases} \text{case i: } A_m = \frac{\pi}{2k_1}, \phi_m = \frac{\pi}{2} - k_1; \\ \text{case ii: } A_m \approx \frac{1.596}{k_1} - 0.276, \phi_m \approx 1.350 - 1.225k_1, \end{cases} \quad (4.23)$$

where $0.2 \leq k_1 \leq 0.6$. The sound accuracy of the formula in *case ii* for approximating the margins is verified and shown in Figure 4.4, which has relative errors in the range of $(-0.3\%, +1.0\%)$.

Thus, by using (4.23), the factor k_1 can be tuned for achieving desired GPMs. (The visible relations between k_1 and GPMs as shown Figure 4.4 can be useful.) This method of tuning k_1 is applicable to all other processes given in Table 4.1, except the DIPTD process. For the exceptional DIPTD processes, we may use the direct method to tune k_1 as presented in the last subsection.

Remark 4.3 *Relation (4.23) indicates that only special GPMs can be attained by the proposed PID tuning rules. This is due to the constraint $\zeta \equiv 1$ as imposed on the closed-loop system during the derivation of the tuning rules. This constraint, however, is found to be appropriate and contributes to satisfactory closed-loop performance, which is justified by numerical examples.*

Tuning k_1 Based on Sensitivity Specification. As introduced in Section 1.1 of Chapter 1, peak sensitivity (M_s) and peak complementary sensitivity (M_t) are measures commonly used to evaluate the closed-loop robustness. Indeed they reflect on the servo (or setpoint) and regulatory (or load-disturbance) performance through the well-known tradeoff between robustness and performance. It is known that appropriate M_s and M_t are in the range of 1.2 to 2.0. Here we establish a relation between the tuning parameter

k_1 and the two peak sensitivities so that the PI/PID controller can be tuned based on sensitivity specifications.

Figure 4.3 The relations between the margins and the tuning parameter k_1 .

Figure 4.4 Relative errors of the margins as computed by analytical formulas in (4.23) for case ii.

Consider the PI tuning for an IPTD process. Since the tuning rule is scalable in the process parameters, the peak sensitivities are the same under the proposed PI control whatever the process parameters are. This justifies considering a particular process, say, e^{-s}/s , and evaluate its peak sensitivities for each given value of k_1 . In this way, relations between M_s , M_t and k_1 can be found numerically. The relation is shown in Figure 4.5. With the visible relations (which can be approximated by certain analytical expressions), the parameter k_1 can be tuned for a desired peak sensitivity or complementary sensitivity. For examples, when $k_1 = 0.43$, the peaks sensitivities are that $M_s \approx 1.59$ and $M_t \approx 1.34$; and when $k_1 = 0.5$, the peaks sensitivities are that $M_s \approx 1.76$ and $M_t \approx 1.43$. The values of k_1 around these two values may give reasonable tradeoffs between performance and robustness. The figure also shows that it is sufficient to restrict k_1 in the range of 0.2 to 0.6, so that the peak sensitivity falls into the range of 1.2 to 2.0 (roughly).

This method of tuning k_1 is approximately applicable to the processes given in Table 4.1 excluding the DIPTD process.

Figure 4.5 Relations between peak sensitivities and the tuning parameter k_1 .

Table 4.1 PID settings for typical processes ^a

$g(s)$	K_c	τ_I	τ_D
$\frac{ke^{-\theta s}}{s}$ (IPTD)	$\frac{k_1}{k\theta}$	$\left(\frac{4}{k_1} - 2\right)\theta$	0
$\frac{ke^{-\theta s}}{\tau_1 s + 1}$ (FOPTD)	$\frac{k_1 \tau_1}{k\theta}$ ^b	$\min\left\{\tau_1, \left(\frac{4}{k_1} - 2\right)\theta\right\}$	0
$\frac{ke^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)}$ (SOPTD)	$\frac{k_1 \tau_1}{k\theta}$ ^b	$\min\left\{\tau_1, \left(\frac{4}{k_1} - 2\right)\theta\right\}$	τ_2
$\frac{ke^{-\theta s}}{s(\tau_2 s + 1)}$ (ILPTD)	$\frac{k_1}{k\theta}$	$\left(\frac{4}{k_1} - 2\right)\theta$	τ_2
$\frac{ke^{-\theta s}}{s^2}$ (DIPTD)	$\frac{k_1}{k\left(\frac{4}{k_1} - 2\right)\theta^2}$	$\left(\frac{4}{k_1} - 2\right)\theta$	τ_I
$ke^{-\theta s}$ (TD)	$K_I = \frac{k_1}{k\theta}$, ($k_1 < 0.5\pi$), with the controller being $c(s) = \frac{K_I}{s}$.		

^a For the first four processes in the table, the relation between k_1 and GPMs are approximately:

$A_m = \pi/(2k_1)$, $\phi_m = \pi/2 - k_1$ if $\tau_I = \tau_1$; and $A_m \approx 1.596/k_1 - 0.276$, $\phi_m \approx 1.350 - 1.225k_1$, otherwise. And for the pure TD process, the relation is that $A_m = \pi/(2k_1)$, $\phi_m = \pi/2 - k_1$.

^b To guarantee closed-loop stability, it requires that $k_1 < 0.5\pi$ if $\tau_I = \tau_1$.

^c SIMC rules [6] can be obtained by replacing $4/k_1 - 2$ with $4/k_1$ in all places.

4.3 Numerical Examples

Numerical examples are presented to show the effectiveness of the proposed PID tuning rules. The results are compared with those attained by the SIMC counterparts.

4.3.1 Simulation Settings

The single tuning parameter in SIMC tuning rules is the closed-loop time constant τ_c .

By defining $\tau_c = (1/k_1 - 1)\theta$, the tuning parameter equivalently changes into k_1 , like the

one used in the proposed rules. Specifically, the SIMC rules are obtained by replacing $4/k_1 - 2$ with $4/k_1$ in all the places of the proposed rules. This implies that the proposed rules adopt smaller integral times and hence larger integral gains if the proportional gains are kept the same.

For fair comparison, the k_1 's are tuned to achieve the same peak sensitivity in each simulation for SIMC and the proposed rules. The peak sensitivity of $M_s = 1.76$ is selected which is the peak sensitivity achieved by the default setting of the proposed rule for IPTD processes. Comparisons of the performances are made on IPTD, FOPTD, SOPTD, ILPTD and DIPTD processes and the process gains are assumed to be one. (Note that SIMC and the proposed rules give the same results in the case of pure TD processes.) For IPTD, ILPTD and DIPTD processes, the lag dominated ($\theta/\tau_1 < 1$), the lag-delay balanced ($\theta/\tau_1 = 1$) and the delay dominated ($\theta/\tau_1 > 1$) cases are considered. For FOPTD and SOPTD processes, only the lag-dominated case is studied since in the non-lag-dominated cases, SIMC and the proposed tuning rules tend to be the same because the integral time will be both equal to the process time constant.

As the derivative mode is noncausal, it is filtered in all simulations. The PID controller is implemented in the form of

$$c(s) = K'_c \left(1 + \frac{1}{\tau'_I s} + \frac{\tau'_D s}{\mu \tau'_D s + 1} \right), \quad (4.24)$$

where μ is usually selected from $[0.1, 0.2]$ in practice [6], and K'_c , τ'_I and τ'_D are the PID parameters calculated from the series PID parameters by (4.3). The setting, $\mu = 0.1$, is applied in all simulations.

4.3.2 Simulation Results

The PID settings are obtained by SIMC and the proposed rules. The simulation results for different processes are shown in Figures 4.6-4.9 and the quantitative performances are summarized in Tables 4.2-4.5. The results indicate that compared to the SIMC counterparts, the proposed rules give better disturbance rejection while achieving the same peak sensitivity (except for DIPTD processes). This implies that the proposed rules better exploit the potentials of PID controllers. This performance gain can be understood as a result of the larger integral gains enforced by the proposed rules: A larger integral gain implies a smaller integral tracking error in response to disturbances [3]. The exceptional results observed in Figure 4.9 in face of DIPTD processes are due to the derivative modes added in ad-hoc manners for both SIMC and the proposed rules. Since the SIMC rule enforces larger derivation gains (refer to Table 4.5), it tends to give smaller overshoots when load disturbance is injected into the system. Future studies may be conducted to determine a better derivative time for the proposed rule.

The results also show that the values of k_i are close to 0.5 for achieving the peak sensitivity of 1.76 for all the processes considered. This justifies the initial value of k_i as 0.5 for the proposed rules. Also note that, for improved disturbance rejections, the proposed rules result in more aggressive setpoint responses as tradeoffs. However, this is reasonable and does not degrade the benefit since feedback control is mainly responsible for disturbance rejection. The setpoint following performance can be improved independently by feedforward control, say setpoint weighting [3].

Simulations (not shown for brevity) also indicate that for the same values of k_i , responses of the PID control systems of different processes (excluding DIPTD processes) attain similar magnitudes of overshoots, and that the rise and peak times are almost proportional to the time delays. These are consistent with the analysis in Appendix B.

Figure 4.6 Responses of PI control of IPTD processes with different delays (refer to Table 4.2 for the PI settings). Setpoint changes at $t = 0$; load disturbances of magnitudes of 3.0, 1.0 and 0.5 are injected at $t = 3, 30$ and 50 , respectively.

Table 4.2 PI settings and performance summary of exemplary IPTD processes ($M_s \approx 1.76$)

$g(s)$	Method	k_1	K_c	τ_I	Setpoint		Load disturbance	
					IAE	TV	IAE	TV
$\frac{e^{-0.1s}}{s}$	SIMC	0.524	5.240	0.763	0.39	7.78	0.44	4.80
	Proposed	0.498	4.975	0.604	0.39	7.98	0.36	5.09
$\frac{e^{-s}}{s}$	SIMC	0.524	0.524	7.641	3.81	0.78	14.58	1.60
	Proposed	0.498	0.498	6.040	3.88	0.80	12.13	1.70
$\frac{e^{-3s}}{s}$	SIMC	0.524	0.175	22.923	11.00	0.26	65.57	0.80
	Proposed	0.497	0.166	18.169	11.51	0.27	54.84	0.85

Figure 4.7 Responses of PI control of an FOPTD process and PID control of an SOPTD process (refer to Table 4.3 for the PI settings). Setpoint changes at $t = 0$; load disturbances both of a magnitude of 3.0 are injected at $t = 3$.

Table 4.3 PID settings and performance summary of exemplary FOPTD and SOPTD processes ($M_s \approx 1.76$)

$g(s)$	Method	k_1	K_c	τ_I	τ_D	Setpoint		Load disturbance	
						IAE	TV	IAE	TV
$\frac{e^{-0.1s}}{s+1}$	SIMC	0.572	5.72	0.699	0	0.26	7.44	0.37	3.96
	Proposed	0.547	5.47	0.531	0	0.29	7.49	0.29	4.27
$\frac{e^{-0.1s}}{(s+1)(0.5s+1)}$	SIMC	0.572	5.72	0.699	0.5	0.32	148.81	0.36	5.62
	Proposed	0.547	5.47	0.531	0.5	0.33	152.61	0.29	5.58

Figure 4.8 Responses of PID control of ILPTD processes with different delays (refer to Table 4.4 for the PID settings). Setpoint changes at $t = 0$; load disturbances of magnitudes of 10.0, 1.0 and 0.5 are injected at $t = 5, 30$ and 40 , respectively.

Table 4.4 PID settings and performance summary of exemplary ILPTD processes ($M_s \approx 1.76$)

$g(s)$	Method	k_1	K_c	τ_I	τ_D	Setpoint		Load disturbance	
						IAE	TV	IAE	TV
$\frac{e^{-0.1s}}{s(s+1)}$	SIMC	0.524	5.24	0.763	1	0.46	216.21	1.45	23.71
	Proposed	0.498	4.975	0.604	1	0.45	218.30	1.21	23.08
$\frac{e^{-s}}{s(s+1)}$	SIMC	0.524	0.524	7.641	1	3.87	6.82	14.59	1.64
	Proposed	0.498	0.498	6.040	1	3.93	6.67	12.13	1.73
$\frac{e^{-3s}}{s(s+1)}$	SIMC	0.524	0.175	22.923	1	10.61	2.08	66.13	0.80
	Proposed	0.497	0.166	18.169	1	11.31	2.01	55.07	0.85

Figure 4.9 Responses of PID control of DIPTD processes with different delays (refer to Table 4.5 for the PID settings). Setpoint changes at $t = 0$; load disturbances of magnitudes of 10.0, 0.2 and 0.05 are injected at $t = 3, 30$ and 50 , respectively.

Table 4.5 PID settings and performance summary of exemplary DIPTD processes ($M_s \approx 1.76$)

$g(s)$	Method	k_1	K_c	τ_i	τ_D	Setpoint		Load disturbance	
						IAE	TV	IAE	TV
$\frac{e^{-0.1s}}{s^2}$	SIMC	0.440	4.840	0.909	0.909	0.6	171.47	1.85	27.65
	Proposed	0.384	4.562	0.842	0.842	0.67	147.41	1.83	27.24
$\frac{e^{-s}}{s^2}$	SIMC	0.436	0.048	9.174	9.174	5.96	1.71	37.85	0.55
	Proposed	0.383	0.045	8.444	8.444	6.74	1.45	37.25	0.54
$\frac{e^{-2s}}{s^2}$	SIMC	0.429	0.012	18.648	18.648	11.65	0.43	76.53	0.14
	Proposed	0.384	0.011	16.833	16.833	13.63	0.35	75.63	0.14

4.4 Conclusions

Simple PID tuning rules were obtained for typical process models. Each rule contains a single scalar to control the tradeoff between closed-loop performance and robustness. Guidelines for tuning such a scalar directly, or based on GPM or peak sensitivity specification were provided. Numerical examples showed that, compared to the SIMC counterparts, the proposed tuning rules can lead to better load disturbance rejection while achieving the same peak sensitivity. This is essentially due to properly tuned up integral gains by the proposed rules. The simulations also indicate that further studies are required to determine an appropriate derivative time for PID control of a DIPTD process.

Chapter 5

PID and PID-C Controller Tuning by 2DOF-DS Approach

This chapter derives explicit tuning rules for PID and PID-C controllers by 2DOF-DS approach. The tuning rules are obtained based on typical process models. Each of the rules contains a single parameter to control the tradeoff between the closed-loop performance and robustness. The resulting 2DOF control is implemented as PID or PID-C control with setpoint weighting. The usefulness of the tuning rules is demonstrated by numerical examples and their advantages are shown over recent PID and PID-C tuning rules.

5.1 Introduction

DS has been widely used to design PID controllers [7-8, 26, 79]. In the DS approach, the closed-loop setpoint-to-output (s2o) or (load) disturbance-to-output (d2o) transfer functions are specified for desired performance while satisfying the stability conditions. The PID controllers are solved approximately with specified closed-loop transfer functions. Conventionally, the closed-loop s2o transfer function is specified for deriving a PID controller as apt for good setpoint response [8, 26, 79]. Recently it has been argued that by specifying the closed-loop d2o transfer function instead, the resulting PID controller can achieve enhanced disturbance rejection while maintain satisfactory setpoint response by

setpoint weighting [7]. Meanwhile, note that the well-known IMC design can be interpreted as DS with certain specifications of the closed-loop transfer functions.

Conventional control design involves a single feedback controller, which has a single degree of freedom (DOF) and is difficult to achieve good setpoint and disturbance responses at the same time. A prefilter provides a second DOF of control and is useful for obtaining smooth setpoint response [80]. By combining a prefilter with a feedback controller, the 2DOF design earns continuing interest in the literature [8, 81-83].

By combining the advantages of 2DOF design and DS, in this chapter we propose 2DOF-DS design. Two methods are proposed for the design, trying to realize specified closed-loop s_2o and d_2o transfer functions for desired performance, respectively. By appropriate approximations of the ideal feedback controllers, the methods result in PID controllers with parameters being explicitly expressed. This leads to new PID tuning rules.

Note that the ideal feedback controllers can alternatively be approximated by controllers with structures other than the PID form, and that more accurate approximations may lead to improved performance. Without complicating the implementation, the PID-C controller (i.e., PID controller cascaded with a lead-lag compensator) is considered as a candidate.

PID-C control was proposed to improve the performance of process control without tribulation of implementation [8, 10, 84-86]. There have been a couple of results on tuning PID-C controllers in literature: Based on the IMC principle, PID-C tuning rules have been derived for stable FOPTD processes [87], IPTD and unstable FOPTD processes [10], and stable or unstable SOPTD processes [86], respectively. And by the DS approach, PID-C tuning rules have been derived for typical process models with one or two integrating modes [8]. With the help of a setpoint filter or setpoint weighting (which are kinds of feedforward control), a plenty of examples have demonstrated that PID-C controllers can

achieve disturbance rejection and robustness both better than PID controllers [8, 10, 86-87].

To make use of the advantages of PID-C control, we extend the 2DOF-DS approach to designing the feedback controller and then approximate it as a PID-C controller. By appropriate approximations, explicit tuning rules are obtained for the PID-C controllers for typical process models.

The rest of this chapter is organized as follows. In Section 5.2, the principles of controller design by 2DOF-DS approach are presented. In Section 5.3, for typical process models, the PI/PID controllers are derived as approximates of the ideal feedback controllers. By specifying the closed-loop transfer functions properly, the prefilter and the PI/PID controller are equivalently implemented as the same PI/PID controller with setpoint weighting. Similar results are obtained when the PID controllers are replaced by PID-C controllers in Section 5.4. Series of numerical examples are given to validate the proposed PI/PID and PID-C controllers in Section 5.5. Finally, conclusions are drawn in Section 5.6.

5.2 Design Principles of 2DOF-DS

Consider the 2DOF control system described in Figure 5.1. In the figure, the notations $P(s)$, $C_1(s)$ and $C_2(s)$ denote the transfer functions of the process, the feedback controller and the prefilter, respectively; $R(s)$, $\tilde{R}(s)$, $E(s)$, $U(s)$, and $Y(s)$ denote the Laplace transforms of the reference input, the filtered input, the error signal, the manipulated variable and the plant output, respectively; $D_i(s)$, $D_o(s)$ and $D_m(s)$ denote the Laplace transforms of the input disturbance, the output disturbance and the measurement noise, respectively; and x_0 denotes the initial state of the process which acts as a disturbance.

Figure 5.1 2DOF control system.

Let the nominal process model be $P_0(s)$. In the DS approach, the closed-loop s2o and d2o transfer functions have to be specified properly in order to satisfy stability conditions [7, 26, 78-79]. It is known that the closed-loop system is internally stable if and only if the six transfer functions are stable [3, 80]:

$$\begin{aligned}
 M_1(s) &= G_{YD_o}(s) = M^{-1}(s), \\
 M_2(s) &= G_{U\tilde{R}}(s) = G_{UD_m}(s) = C_1(s)M^{-1}(s), \\
 M_3(s) &= G_{YD_i}(s) = P_0(s)M^{-1}(s), \\
 M_4(s) &= G_{Y\tilde{R}}(s) = P_0(s)C_1(s)M^{-1}(s), \\
 M_5(s) &= G_{UR}(s) = C_1(s)C_2(s)M^{-1}(s), \\
 M_6(s) &= G_{YR}(s) = P_0(s)C_1(s)C_2(s)M^{-1}(s),
 \end{aligned} \tag{5.1}$$

where $M(s) := 1 + P_0(s)C_1(s)$. The prefilter $C_2(s)$ can be designed independently for stability of $M_5(s)$ and $M_6(s)$; and the feedback controller $C_1(s)$ is concerned with stability of $M_i(s)$ ($i=1, 2, 3, 4$). For simplicity, the design of $C_1(s)$ can be focused on ensuring desired properties of $M_4(s)$ (the s2o transfer function). Such design, however, may not give a satisfactory $M_3(s)$ whose properties directly determine the ability of rejecting load disturbances. An alternative solution is to design $C_1(s)$ for achieving a desired $M_3(s)$ (the d2o transfer function) and to design $C_2(s)$ for achieving a desired $M_6(s)$. Meanwhile, satisfactory $M_1(s)$ and $M_2(s)$ can be accomplished by tuning the

parameters of the controllers properly. Two DS-based methods of control design are motivated, of which Method 1 is similar to the well-known IMC [83, 88].

5.2.1 Design for Desired s2o Response (Method 1)

For desired s2o response, the closed-loop s2o transfer function with a filtered setpoint, denoted by $G_{Y\bar{R}}(s)$, must be specified properly in order to satisfy the conditions of internal stability. The basic form of $G_{Y\bar{R}}(s)$ can be determined as follows.

From (5.1) we have

$$\begin{aligned} M_1(s) &= G_{Y\bar{R}}(s)C_1^{-1}(s)P_0^{-1}(s) = 1 - G_{Y\bar{R}}(s), \\ M_2(s) &= G_{Y\bar{R}}(s)P_0^{-1}(s), \\ M_3(s) &= G_{Y\bar{R}}(s)C_1^{-1}(s) = (1 - G_{Y\bar{R}}(s))P_0(s), \\ M_4(s) &= G_{Y\bar{R}}(s). \end{aligned} \tag{5.2}$$

Relation (5.2) implies that, in order to ensure $M_i(s)$ ($i=1, 2, 3, 4$) be stable, $G_{Y\bar{R}}(s)$ has to be specified such that three conditions are satisfied: *i)* $G_{Y\bar{R}}(s)$ is stable; *ii)* $G_{Y\bar{R}}(s)$ has zeros at any right-half plane (RHP) zeros of $P_0(s)$; and *iii)* $1 - G_{Y\bar{R}}(s)$ has zeros at any RHP poles of $P_0(s)$.

Factorize the process model as

$$P_0(s) = P_{0^+}(s)P_{0^-}(s), \tag{5.3}$$

where $P_{0^+}(s)$ contains any time delays and RHP zeros and it satisfies $P_{0^+}(0) = 1$. Then

$G_{Y\bar{R}}(s)$ can be specified as

$$G_{Y\bar{R}}(s) = \frac{P_{0^+}(s)N_1(s)}{(\lambda s + 1)^r}, \tag{5.4}$$

where λ is an adjustable parameter which controls the tradeoff between performance and robustness, and r is an integer large enough to make $G_{Y\bar{R}}(s)$ proper. And $N_1(s)$ is a

polynomial defined as

$$N_1(s) := 1 + \sum_{i=1}^m \alpha_i s^i, \quad (5.5)$$

where $m := m_+ + m_-$. Here m_+ is the total number of RHP poles of $P_0(s)$ and m_- is the number of left-half plane (LHP) poles of $P_0(s)$ which are intended to cancel (Therefore m_- is between zero and the total number of the LHP poles of $P_0(s)$). And α_i ($i=1, 2, \dots, m$) are solved from the equations:

$$\begin{aligned} & \left| 1 - G_{Y\bar{R}}(s) \right|_{s=RP_1, RP_2, \dots, RP_{m_0}} = 0, \\ & \left| \frac{d}{ds} G_{Y\bar{R}}(s) \right|_{s=RP_i} = 0, \dots, \left| \frac{d^{n_i-1}}{ds^{n_i-1}} G_{Y\bar{R}}(s) \right|_{s=RP_i} = 0, \\ & \text{for } i = 1, 2, \dots, m_0. \end{aligned} \quad (5.6)$$

Here RP_i ($i=1, 2, \dots, m_0$) denote the distinct poles among the m poles of $P_0(s)$, and n_i is the number of duplicates of pole RP_i , satisfying $m = \sum_{i=1}^{m_0} n_i$. Note that the limits at the poles may be taken in the above equations. In particular, if $C_1(s)$ takes a form of

$$C_1(s) = C_{1^+}(s)C_{1^-}(s), \quad (5.7)$$

where $C_{1^+}(s)$ contains any RHP zeros and satisfies $C_{1^+}(0)=1$, then we can let $N_1(s) := C_{1^+}(s)$.

Next, for good s2o response, the specification of $G_{YR}(s)$ is relatively flexible.

Typically it can be specified as a filter in the form of

$$G_{YR}(s) := \frac{N_2(s)}{(\lambda s + 1)^r}, \quad (5.8)$$

where λ and r are defined in (5.4), and $N_2(s)$ is a polynomial of s multiplied by the same time delay components of the process to give satisfactory setpoint tracking. With the specified $G_{Y\bar{R}}(s)$ and $G_{YR}(s)$, from (5.1) we have

$$\begin{aligned} G_{Y\bar{R}}(s) &= C_1(s)P_0(s)(1+C_1(s)P_0(s))^{-1}, \\ G_{YR}(s) &= G_{Y\bar{R}}(s)C_2(s), \end{aligned} \quad (5.9)$$

which solve

$$\begin{aligned} C_1(s) &= G_{Y\bar{R}}(s)[P_0(s)(1-G_{Y\bar{R}}(s))]^{-1}, \\ C_2(s) &= G_{YR}(s)G_{Y\bar{R}}^{-1}(s). \end{aligned} \quad (5.10)$$

Hence (5.10) gives the ideal 2DOF controllers leading to desired transfer functions

$G_{Y\bar{R}}(s)$ and $G_{YR}(s)$.

Remark 5.1 (a) The zeros of $P_0(s)$ at the origin can be classified into $P_{0^+}(s)$ (in which case $P_{0^+}(0)=1$ makes sense by omitting the augmented part with zeros of origin) and the zeros of $C_1(s)$ at the origin can be classified into $C_{1^+}(s)$. Such factorizations are recommended since they eliminate closed-loop poles of origin as are undesirable. (b) Since poles are all designated for a closed-loop system, in (5.5) m normally means the total number of poles of $P_0(s)$.

5.2.2 Design for Desired d2o Response (Method 2)

For desired d2o response, the closed-loop d2o transfer function, $G_{YD_i}(s)$, must be specified properly in order to satisfy the conditions of internal stability. The basic form of $G_{YD_i}(s)$ can be determined as follows.

From (5.1) we have

$$\begin{aligned} M_1(s) &= G_{YD_i}(s)P_0^{-1}(s), \\ M_2(s) &= G_{YD_i}(s)C_1(s)P_0^{-1}(s) = (1-G_{YD_i}(s)P_0^{-1}(s))P_0^{-1}(s), \\ M_3(s) &= G_{YD_i}(s), \\ M_4(s) &= G_{YD_i}(s)C_1(s) = 1-G_{YD_i}(s)P_0^{-1}(s). \end{aligned} \quad (5.11)$$

Relation (5.11) implies that, in order to ensure $M_i(s)$ ($i=1, 2, 3, 4$) be stable, $G_{YD_i}(s)$

has to be specified such that two conditions are satisfied: *i)* $G_{YD_i}(s)$ is stable; *ii)* both $G_{YD_i}(s)$ and $1 - G_{YD_i}(s)P_0^{-1}(s)$ have zeros at any RHP zeros of $P_0(s)$.

Factorize the process model as (5.3). Then $G_{YD_i}(s)$ can be specified in the form of

$$G_{YD_i}(s) = \frac{P_{0^+}(s)N'_1(s)}{(\lambda's + 1)^{r'}}, \quad (5.12)$$

where λ' and r' are similarly defined as λ and r in (5.4). And $N'_1(s)$ is a polynomial function of s defined as

$$N'_1(s) := 1 + \sum_{i=1}^{m'} \alpha'_i s^i, \quad (5.13)$$

where $m' := m'_+ + m'_-$. Here m'_+ is the total number of RHP zeros of $P_0(s)$ and m'_- is the number of LHP zeros of $P_0(s)$ which are intended to cancel. And α'_i ($i = 1, 2, \dots, m'_0$) are solved from the equations:

$$\begin{aligned} & \left| 1 - G_{YD_i}(s)P_0^{-1}(s) \right|_{s=RZ_1, RZ_2, \dots, RZ_{m'_0}} = 0, \\ & \left| \frac{d}{ds} (G_{YD_i}(s)P_0^{-1}(s)) \right|_{s=RZ_i} = 0, \dots, \left| \frac{d^{n'_i-1}}{ds^{n'_i-1}} (G_{YD_i}(s)P_0^{-1}(s)) \right|_{s=RZ_i} = 0, \\ & \text{for } i = 1, 2, \dots, m'_0. \end{aligned} \quad (5.14)$$

Here RZ_i ($i = 1, 2, \dots, m'_0$) denote the distinct zeros among the m' zeros of $P_0(s)$, and n'_i is the number of duplicates of zero RZ_i , satisfying $m' = \sum_{i=1}^{m'_0} n'_i$.

For good setpoint response, $G_{YR}(s)$ can be specified the same as (5.8) with λ and r replaced by λ' and r' respectively. With the specified $G_{YD_i}(s)$ and $G_{YR}(s)$, from (5.1) we have

$$\begin{aligned} G_{YD_i}(s) &= P_0(s)(1 + C_1(s)P_0(s))^{-1}, \\ G_{YR}(s) &= G_{YD_i}(s)C_1(s)C_2(s), \end{aligned} \quad (5.15)$$

which solve

$$\begin{aligned} C_1(s) &= G_{YD_i}^{-1}(s) - P_0^{-1}(s), \\ C_2(s) &= G_{YR}(s)(C_1(s)G_{YD_i}(s))^{-1}. \end{aligned} \quad (5.16)$$

Hence (5.16) gives the ideal 2DOF controllers leading to desired transfer functions $G_{YR}(s)$ and $G_{YD_i}(s)$.

Remark 5.2 *The zeros of $P_0(s)$ at the origin can be classified into $P_{0^+}(s)$ (in which case $P_{0^+}(0)=1$ makes sense by omitting the augmented part of zeros of origin); and the poles of $C_1(s)$ at the origin can be added as a factor of the numerator of $G_{YD_i}(s)$. The modified factorizations are recommended since they eliminate the closed-loop poles of origin as are undesirable.*

5.3 PI/PID Controller as the Feedback Controller

The ideal feedback controller $C_1(s)$ in (5.10) or (5.16) usually does not have a simple structure. To obtain a simple feedback controller, $C_1(s)$ is approximated by a PI or PID controller. The ideal PID controller in a standard form is considered:

$$C(s) = K_c \left(1 + \frac{1}{T_i s} + T_d s \right), \quad (5.17)$$

where K_c , T_i and T_d are the proportional (P), integral (I) and derivative (D) parameters respectively. When $T_d = 0$, (5.17) corresponds to a PI controller

5.3.1 PI/PID Controller Design with Method 1

A general way of deriving the PI/PID controllers with Method 1 is firstly presented. Then the PI/PID parameters are obtained explicitly for typical process models.

$C_1(s)$ being a PI controller. To illustrate the design, let us consider a time-delay process model with a single pole and no RHP zeros. The PI controller is derived to match

a desired s2o transfer function approximately. According to (5.4) and (5.8), the desired closed-loop transfer functions are specified as follows:

$$G_{Y\tilde{R}}(s) := \frac{(\alpha s + 1)e^{-\tau s}}{(\lambda_1 s + 1)^2}, \quad G_{YR}(s) := \frac{(\gamma_p \alpha s + 1)e^{-\tau s}}{(\lambda_1 s + 1)^2}, \quad (5.18)$$

where α is a constant to be determined, τ the time delay of the process model, λ_1 the time constant that controls the tradeoff between performance and robustness, and γ_p a proper weighting scalar. Substituting (5.18) into (5.10) gives

$$C_1(s) = \frac{P_0^{-1}(s)(\alpha s + 1)e^{-\tau s}}{(\lambda_1 s + 1)^2 - (\alpha s + 1)e^{-\tau s}} =: \frac{P_0^{-1}(s)(\alpha s + 1)e^{-\tau s}}{sD(s)} =: \frac{f(s)}{s}. \quad (5.19)$$

$D(s)$ can be interpreted as a polynomial of s by Maclaurin expansion of the denominator of $C_1(s)$. In order to approximate $C_1(s)$ in (5.19) as a PI controller, expand $f(s)$ as a Maclaurin series:

$$C_1(s) = \frac{1}{s} \left(f(0) + f^{(1)}(0)s + \frac{f^{(2)}(0)}{2!}s^2 + \dots \right). \quad (5.20)$$

The derivatives are obtained as the limits with $s \rightarrow 0$. Consequently, the PI parameters are obtained as

$$K_c = f^{(1)}(0), \quad T_i = f^{(1)}(0)/f(0). \quad (5.21)$$

The intermediate variable α is solved by requiring $D(s)$ to have a zero at the pole of $P_0(s)$ (which becomes that $\lim_{s \rightarrow 0} D(s) = 0$ if the pole is zero). From (5.10) the prefilter is obtained as

$$C_2(s) = \frac{\gamma_p \alpha s + 1}{\alpha s + 1}. \quad (5.22)$$

By similar procedures, the PI settings for two typical process models are obtained and summarized in Table 5.1. In both cases, the prefilters, $C_2(s)$'s, keep the form of (5.22).

$C_1(s)$ **being a PID controller.** Consider a time-delay process model with a single pole and no RHP zeros. The PID parameters can be obtained by truncating the Maclaurin series in (5.20) to the second order, which gives

$$K_c = f^{(1)}(0), T_i = f^{(1)}(0)/f(0), T_d = f^{(2)}(0)/[2f^{(1)}(0)]. \quad (5.23)$$

Next, consider a time-delay process model with two poles and no RHP zeros. The PID feedback controllers can similarly be obtained. According to (5.4) and (5.8), specify the desired closed-loop transfer functions as (which have referred to the forms of IMC filters used in [83, 89])

$$G_{YR}(s) = \frac{(\alpha_2 s^2 + \alpha_1 s + 1)e^{-\tau s}}{(\lambda_1 s + 1)^4}, \quad G_{YR}(s) = \frac{(\gamma_d \alpha_2 s^2 + \gamma_p \alpha_1 s + 1)e^{-\tau s}}{(\lambda_1 s + 1)^4}. \quad (5.24)$$

Here $\alpha_{1,2}$ are constants to be determined and the other parameters are defined similarly to those in (5.18). Substituting (5.24) into (5.10) gives

$$C_1(s) = \frac{P_0^{-1}(s)(\alpha_2 s^2 + \alpha_1 s + 1)e^{-\tau s}}{(\lambda_1 s + 1)^4 - (\alpha_2 s^2 + \alpha_1 s + 1)e^{-\tau s}} =: \frac{P_0^{-1}(s)(\alpha_2 s^2 + \alpha_1 s + 1)e^{-\tau s}}{sD(s)} =: \frac{f(s)}{s}. \quad (5.25)$$

Expand $f(s)$ as a Maclaurin series and it gives the expression of (5.20). As a result, the PID parameters are obtained and expressed in the same form of (5.23). The variables $\alpha_{1,2}$ are solved by requiring $D(s)$ to have zeros at the poles of $P_0(s)$ (If the two poles are zeros, the requirement will be that $\lim_{s \rightarrow 0} D(s) = 0$ and $\lim_{s \rightarrow 0} D^{(1)}(s) = 0$.). And from (5.10) the prefilter is obtained as

$$C_2(s) = \frac{\gamma_d \alpha_2 s^2 + \gamma_p \alpha_1 s + 1}{\alpha_2 s^2 + \alpha_1 s + 1}. \quad (5.26)$$

In a similar way, the PID settings for FOPTD and SOPTD process models are obtained and summarized in Table 5.2. In all cases the prefilters keep the form of (5.26).

Table 5.1 PI settings for typical process models (Method 1)

$G_{YR}(s)$ ^a	$P_0(s)$	α	KK_c	T_i
A	$\frac{Ke^{-\tau s}}{s}$	$2\lambda_1 + \tau$	$\frac{T_i}{\lambda_1^2 + \tau\alpha - 0.5\tau^2}$	α
A	$\frac{Ke^{-\tau s}}{T_1 s + 1}$	$T_1 \left[1 - (1 - \lambda_1/T_1)^2 e^{-\tau/T_1} \right]$	$\frac{T_i}{2\lambda_1 + \tau - \alpha}$	$\alpha + T_1 + \frac{0.5\tau^2 - \alpha\tau - \lambda_1^2}{2\lambda_1 + \tau - \alpha}$

^a “A” denotes the desired filtered s2o transfer function and $A := (\alpha s + 1)e^{-\tau s} / (\lambda_1 s + 1)^2$. And the corresponding desired s2o transfer function is specified as $G_{YR}(s) := (\gamma_p \alpha s + 1)e^{-\tau s} / (\lambda_1 s + 1)^2$.

Table 5.2 PID settings for typical process models (Method 1)

$G_{YR}(s)$ ^b	$P_0(s)$	α_1, α_2	KK_c	T_i	T_d
A	$\frac{Ke^{-\tau s}}{T_1 s + 1}$	$T_1 [1 - (1 - \lambda_1/T_1)^2 e^{-\tau/T_1}],$ 0	$\frac{T_i}{2\lambda_1 + \tau - \alpha_1}$	$\alpha_1 + T_1 + \frac{0.5\tau^2 - \alpha_1\tau - \lambda_1^2}{2\lambda_1 + \tau - \alpha_1}$	$\frac{T_i\alpha_1}{T_i} + \frac{0.5\alpha_1\tau^2 - \tau^3/6}{T_i(2\lambda_1 + \tau - \alpha_1)} + \frac{0.5\tau^2 - \alpha_1\tau - \lambda_1^2}{2\lambda_1 + \tau - \alpha_1}$
B	$\frac{Ke^{-\tau s}}{(T_1 s + 1)(T_2 s + 1)}$	$\frac{\left\{ \begin{array}{l} T_2^2 [(1 - \lambda_1/T_2)^4 e^{-\tau/T_2} - 1] \\ - T_1^2 [(1 - \lambda_1/T_1)^4 e^{-\tau/T_1} - 1] \end{array} \right\}}{T_1 - T_2},$ ^c $T_1\alpha_1 + T_1^2 [(1 - \lambda_1/T_1)^4 e^{-\tau/T_1} - 1].$	$\frac{T_i}{4\lambda_1 + \tau - \alpha_1}$	$\alpha_1 + T_1 + T_2 + \frac{0.5\tau^2 - \alpha_1\tau + \alpha_2 - 6\lambda_1^2}{4\lambda_1 + \tau - \alpha_1}$	$\frac{\alpha_2 + T_1 T_2 + (T_1 + T_2)\alpha_1}{T_i} - \frac{\tau^3/6 - 0.5\alpha_1\tau^2 + \alpha_2\tau + 4\lambda_1^3}{T_i(4\lambda_1 + \tau - \alpha_1)} + \frac{0.5\tau^2 - \alpha_1\tau + \alpha_2 - 6\lambda_1^2}{4\lambda_1 + \tau - \alpha_1}$

^b A and B denote the desired filtered s2o transfer functions and $A := (\alpha_1 s + 1)e^{-\tau s} / (\lambda_1 s + 1)^2$ and $B := (\alpha_2 s^2 + \alpha_1 s + 1)e^{-\tau s} / (\lambda_1 s + 1)^4$. While the filtered s2o transfer functions are A and B, the desired s2o transfer functions are specified as $G_{YR}(s) := (\gamma_p \alpha_1 s + 1)e^{-\tau s} / (\lambda_1 s + 1)^2$ and $G_{YR}(s) := (\gamma_d \alpha_2 s^2 + \gamma_p \alpha_1 s + 1)e^{-\tau s} / (\lambda_1 s + 1)^4$, respectively.

^c If $T_1 = T_2$, then $\alpha_1 = 2T_1 - [4\lambda_1(1 - \lambda_1/T_1)^3 + (2T_1 + \tau)(1 - \lambda_1/T_1)^4]e^{-\tau/T_1}$ and α_2 keeps the same.

Implementation. Thanks to appropriate DS, the above 2DOF control, consisting of $C_1(s)$ (a PI/PID controller) and $C_2(s)$ (a first/second order prefilter), can be implemented as the same PI/PID control with setpoint weighting. This is explained below.

Consider the case of $C_1(s)$ being a PID controller as an example. The closed-loop s2o transfer function is expressed as

$$G_{YR}(s) = \frac{(T_i T_d s^2 + T_i s + 1)e^{-\tau s}}{\frac{T_i}{KK_c} s(T_1 s + 1)(T_2 s + 1) + (T_i T_d s^2 + T_i s + 1)e^{-\tau s}}. \quad (5.27)$$

Suppose that approximation of the ideal feedback controller in (5.25) by a PID controller is accurate. By comparing (5.27) with the ideal $G_{YR}(s)$ in (5.24), it implies that

$$T_i \approx \alpha_1 \text{ and } T_i T_d \approx \alpha_2. \quad (5.28)$$

Hence the prefilter in (5.26) can be approximated as

$$C_2(s) \approx \frac{\gamma_d T_i T_d s^2 + \gamma_p T_i s + 1}{T_i T_d s^2 + T_i s + 1}. \quad (5.29)$$

The controller (5.29) behaves equivalently as setpoint weighting on the PID controller $C_1(s)$, with a weight of γ_d on the setpoint for the derivative action and a weight of γ_p on the setpoint for the proportional action [3]. Therefore, the 2DOF control can be implemented as the same PID control with setpoint weighting, which is expressed in the time domain as

$$u_{pid}(t) = K_c \left((\gamma_p r(t) - y(t)) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{d(\gamma_d r(t) - y(t))}{dt} \right), \quad (5.30)$$

where $e(t) := r(t) - y(t)$. Usually γ_p and γ_d both take values in the range of $[0, 1]$. It is often to set γ_d as zero to avoid derivative kick [3]. Note that the larger γ_p is, the more aggressive the setpoint response will be. Empirical values of γ_p are in the range of 0.4 to 0.6 [8, 86]. Similar implementation applies when $C_1(s)$ is a PI controller.

5.3.2 PI/PID Controller Design with Method 2

This subsection derives $C_1(s)$ as PI/PID controllers for typical processes using Method 2 introduced in Section 5.2.2. Exemplary procedures are given to illustrate the derivations of PI and PID controller parameters, respectively. Consider an FOPTD process of the form

$$P_0(s) = \frac{Ke^{-\tau s}}{(T_1s+1)(T_2s+1)}, \quad (5.31)$$

where K is the process gain, $\tau (\geq 0)$ the time delay, and $T_{1,2} (\geq 0)$ the time constants of the process, of which at least one is nonzero. To simplify expressions, the normalized parameters are used when necessary:

$$\theta := \tau/T_1, \quad T := T_2/T_1, \quad \lambda := \lambda_1/T_1. \quad (5.32)$$

The rationale of such normalizations can be referred to [90-91].

$C_1(s)$ being a PI controller. Consider the process model (5.38) with $T_2 \equiv 0$. The PI parameters are derived to match a desired d2o transfer function approximately. According to Section 5.2.2, specify the desired closed-loop transfer functions as

$$G_{YD_i}(s) := \frac{T_i s e^{-\tau s}}{K_c (\lambda_1 s + 1)^2}, \quad G_{YR}(s) := \frac{(\gamma_p T_i s + 1) e^{-\tau s}}{(\lambda_1 s + 1)^2}. \quad (5.33)$$

From (5.15) we have

$$G_{YD_i}(s) = \frac{T_i s e^{-\tau s} / K_c}{T_i s (T_1 s + 1) / (K K_c) + e^{-\tau s} (T_i s + 1)}. \quad (5.34)$$

Expand the denominator of $G_{YD_i}(s)$ in (5.34) as a Maclaurin series and compare it with the denominator of $G_{YD_i}(s)$ in (5.33). By equating the coefficients of the polynomials of s for the first two orders, the PI parameters are solved and given in Table 5.3. Then, from

(5.15) the prefilter is obtained as (5.22) with $\alpha := T_i$, which behaves equivalently as setpoint weighting on the PI control.

To summarize, the 2DOF-DS design derives the PI feedback controller $C_1(s)$ with its parameters being explicitly given in Table 5.3 and the prefilter $C_2(s)$ as a filter given in (5.22) with $\alpha := T_i$. The two controllers are together implemented as the same PI controller with setpoint weighting as expressed in (5.30), where $T_d \equiv 0$.

Similarly, the PI setting for an IPTD process is obtained and given in Table 5.3. The corresponding prefilter $C_2(s)$ keeps the form of (5.22) with $\alpha := T_i$. And the 2DOF control is implemented as a setpoint-weighted PI control expressed in (5.30).

$C_1(s)$ being a PID controller. Consider the process model (5.38) with nonzero T_1 and T_2 . The controllers $C_1(s)$ and $C_2(s)$ are obtained in a similar way. Specify the desired closed-loop transfer functions as

$$G_{YD_i}(s) := \frac{T_i s e^{-\tau s}}{K_c (\lambda_1 s + 1)^3}, \quad G_{YR}(s) := \frac{(\gamma_d T_i T_d s^2 + \gamma_p T_i s + 1) e^{-\tau s}}{(\lambda_1 s + 1)^3}. \quad (5.35)$$

According to (5.15) we have

$$G_{YD_i}(s) = \frac{T_i s e^{-\tau s} / K_c}{\frac{T_i}{KK_c} s(T_1 s + 1)(T_2 s + 1) + e^{-\tau s} (T_i T_d s^2 + T_i s + 1)}. \quad (5.36)$$

Expand the denominator of $G_{YD_i}(s)$ in (5.36) as a Maclaurin series and compare it with the denominator of $G_{YD_i}(s)$ in (5.35). By equating the coefficients of the polynomials of s for the first three orders, the PID parameters are solved and given in Table 5.4. Then from (5.15) the prefilter is obtained as (5.26).

To summarize, the 2DOF-DS design derives the PID feedback controller $C_1(s)$ with its parameters being explicitly given in Table 5.4 and the prefilter $C_2(s)$ as a filter given

in (5.26). The two controllers are together implemented as a PID controller with setpoint weighting as expressed in (5.30).

In similar manners, the PID settings for typical process models are obtained and summarized in Table 5.4. The prefilters, $C_2(s)$'s, always take the form of (5.26). And the 2DOF controls are all implemented as setpoint-weighted PID controls expressed in (5.30).

Finally, note that the parameter λ_1 can be tuned in a way similar to those of the existing DS-based PI/PID [7] or IMC-PI/PID [6, 74, 89] controllers. Usually a larger λ_1 leads to stronger robustness yet more sluggish response, and vice versa. However, it should be cautioned that such relation is not always true. Specific situations when such relation does not hold seem too complicated and are unclear so far [21]. Simulations indicate that, to be conservative, λ_1 can initially be set as three or two times of the process time delay and then be tuned up or down until satisfactory performance is attained.

Remark 5.3 *Ideally the PI/PID settings are also applicable to unstable processes by replacing the parameter K or T_1 or T_2 in the Tables 1-4 with $-K$ or $-T_1$ or $-T_2$, respectively. The applicability, however, is restricted since the approximation errors involved may become unignorable and cause instability when the process is unstable.*

Table 5.3 PI settings for typical process models (Method 2)

$G_{D,Y}(s)$ ^d	$P_0(s)$	KK_c	T_i
A	$\frac{Ke^{-\tau s}}{s}$	$\frac{2(\tau + 2\lambda_1)}{\tau^2 + 4\lambda_1\tau + 2\lambda_1^2}$	$\tau + 2\lambda_1$
A	$\frac{Ke^{-\tau s}}{T_1s + 1}$	$\frac{\theta^2 + 2\theta + 4\lambda - 2\lambda^2}{\theta^2 + 4\theta\lambda + 2\lambda^2}$	$\frac{T_1}{2} \frac{\theta^2 + 2\theta + 4\lambda - 2\lambda^2}{\theta + 1}$

^d “A” denotes the reference transfer function and it is specified as $A := T_i s e^{-\tau s} / [K_c (\lambda_1 s + 1)^2]$. The desired s2o transfer function is always specified as $G_{YR}(s) := (\gamma_p T_i s + 1) e^{-\tau s} / (\lambda_1 s + 1)^2$.

Table 5.4 PID settings for typical process models (Method 2)

$G_{DY}(s)$ ^e	$P_0(s)$	KK_c	T_i	T_d
B	$\frac{Ke^{-\tau s}}{s}$	$\frac{12\tau(\tau + 6\lambda_1)}{\tau^3 + 6\lambda_1(3\tau^2 + 6\lambda_1\tau + 4\lambda_1^2)}$	$\frac{\tau}{2} + 3\lambda_1$	$\frac{5\tau^3 + 6\lambda_1(3\tau^2 + 6\lambda_1\tau + 4\lambda_1^2)}{12\tau(\tau + 6\lambda_1)}$
C	$\frac{Ke^{-\tau s}}{s^2}$	$\frac{6(\tau + 3\lambda_1)}{\tau^3 + 3\lambda_1(3\tau^2 + 6\lambda_1\tau + 2\lambda_1^2)}$	$\tau + 3\lambda_1$	$\frac{\tau^2 + 6\lambda_1\tau + 6\lambda_1^2}{2(\tau + 3\lambda_1)}$
C	$\frac{Ke^{-\tau s}}{s(T_1s + 1)}$	$\frac{6}{T_1} \frac{(\theta + 1)(\theta + 3\lambda)}{\theta^3 + 3\lambda(3\theta^2 + 6\lambda\theta + 2\lambda^2)}$	$(\theta + 3\lambda)T_1$	$\frac{T_1}{6} \frac{\begin{bmatrix} 2\theta^3 + 3(1 + 3\lambda)\theta^2 \\ + 6\lambda(3\theta + 3\lambda - \lambda^2) \end{bmatrix}}{(\theta + 1)(\theta + 3\lambda)}$
B	$\frac{Ke^{-\tau s}}{T_1s + 1}$	$\frac{\begin{bmatrix} 5\theta^3 + 6\theta^2(2 + 3\lambda) \\ + 36\lambda\theta(2 - \lambda) - 24\lambda^3 \end{bmatrix}}{\theta^3 + 6\lambda(3\theta^2 + 6\lambda\theta + 4\lambda^2)}$	$\frac{T_1}{12} \frac{\begin{bmatrix} 5\theta^3 + 6\theta^2(2 + 3\lambda) \\ + 36\lambda\theta(2 - \lambda) - 24\lambda^3 \end{bmatrix}}{\theta(\theta + 2)}$	$T_1 \frac{\begin{bmatrix} 2\theta^4 + 5\theta^3 + 18\lambda\theta^2 \\ + 12\lambda^2\theta(3 - 2\lambda) - 24\lambda^3 \end{bmatrix}}{\begin{bmatrix} 5\theta^3 + 6\theta^2(2 + 3\lambda) \\ + 36\lambda\theta(2 - \lambda) - 24\lambda^3 \end{bmatrix}}$
C	$\frac{Ke^{-\tau s}}{(T_1s + 1)(T_2s + 1)}$	$\frac{\begin{bmatrix} \theta^3/3 + \theta^2(T + 1) \\ + \theta(T + 3\lambda T + 3\lambda - 3\lambda^2) \\ + \lambda(3T - \lambda^2) \end{bmatrix}}{2 \frac{\theta^3/3 + \lambda(3\theta^2 + 6\lambda\theta + 2\lambda^2)}{\theta^3/3 + \lambda(3\theta^2 + 6\lambda\theta + 2\lambda^2)}}$	$T_1 \frac{\begin{bmatrix} \theta^3/3 + \theta^2(T + 1) \\ + \theta(T + 3\lambda T + 3\lambda - 3\lambda^2) \\ + \lambda(3T - \lambda^2) \end{bmatrix}}{0.5\theta^2 + \theta(T + 1) + T}$	$\frac{T_1}{12} \frac{\begin{bmatrix} \theta^4 + 4(1 + T)\theta^3 + 6\theta^2(T + 3\lambda T) \\ + 3\lambda - 3\lambda^2 + 12\lambda\theta(3T - \lambda^2) \\ + 12\lambda^2(3T - \lambda T - \lambda) \end{bmatrix}}{\begin{bmatrix} \theta^3/3 + \theta^2(T + 1) + \theta(T + 3\lambda T) \\ + 3\lambda - 3\lambda^2 + \lambda(3T - \lambda^2) \end{bmatrix}}$
C	$\frac{Ke^{-\tau s}}{T_1^2s + 2\zeta T_1s + 1}$	$\frac{\begin{bmatrix} \theta^3/3 + 2\zeta\theta^2 \\ + \theta(1 + 6\zeta\lambda - 3\lambda^2) \\ + \lambda(3 - \lambda^2) \end{bmatrix}}{2 \frac{\theta^3/3 + \lambda(3\theta^2 + 6\lambda\theta + 2\lambda^2)}{\theta^3/3 + \lambda(3\theta^2 + 6\lambda\theta + 2\lambda^2)}}$	$T_1 \frac{\begin{bmatrix} \theta^3/3 + 2\zeta\theta^2 \\ + \theta(1 + 6\zeta\lambda - 3\lambda^2) \\ + \lambda(3 - \lambda^2) \end{bmatrix}}{0.5\theta^2 + 2\zeta\theta + 1}$	$\frac{T_1}{12} \frac{\begin{bmatrix} \theta^4 + 8\zeta\theta^3 + 6\theta^2(1 + 6\zeta\lambda) \\ - 3\lambda^2 + 12\lambda\theta(3 - \lambda^2) \\ + 12\lambda^2(3 - 2\zeta\lambda) \end{bmatrix}}{\begin{bmatrix} \theta^3/3 + 2\zeta\theta^2 + \theta(1 + 6\zeta\lambda) \\ - 3\lambda^2 + \lambda(3 - \lambda^2) \end{bmatrix}}$

^e B and C denote the reference transfer functions and they are specified as $B := T_i s(0.5\tau s + 1)e^{-\tau s} / [K_c(\lambda_1 s + 1)^3]$ and $C := T_i s e^{-\tau s} / [K_c(\lambda_1 s + 1)^3]$. The desired s2o transfer function is always specified as $G_{yr}(s) := (\gamma_d T_i T_d s^2 + \gamma_p T_i s + 1)e^{-\tau s} / (\lambda_1 s + 1)^3$.

5.4 PID-C Controller as the Feedback Controller

In this section, Method 1 for 2DOF-DS is adopted to determine the ideal feedback controller as in (5.10) or (5.16). The controller is then approximated by a PID-C controller. The PID-C controller takes the form of

$$C_1(s) := K_c \left(1 + \frac{1}{T_i s} + T_d s \right) \frac{as + 1}{bs + 1}, \quad (5.37)$$

where K_c , T_i and T_d are the PID parameters, and a and b are the parameters of a lead-lag compensator. In the following, we firstly illustrate the derivation of a PID-C controller with an exemplary process model, and then present the PID-C controllers derived for typical process models.

Consider an exemplary nominal process

$$P_0(s) = \frac{Ke^{-\tau s}}{(T_1s+1)(T_2s+1)}. \quad (5.38)$$

By the 2DOF-DS approach, specify the desired closed-loop transfer functions as

$$\begin{aligned} G_{Y\bar{R}}(s) &= \frac{\alpha_2 s^2 + \alpha_1 s + 1}{(\lambda_1 s + 1)^3} e^{-\tau s}, \\ G_{YR}(s) &= \frac{(\gamma_d \alpha_2 s^2 + \gamma_p \alpha_1 s + 1)e^{-\tau s}}{(\lambda_1 s + 1)^3}, \end{aligned} \quad (5.39)$$

where γ is a proper scalar. According to (5.10), the controllers are solved as

$$\begin{aligned} C_1(s) &= \frac{(T_1s+1)(T_2s+1)(\alpha_2 s^2 + \alpha_1 s + 1)}{K[(\lambda_1 s + 1)^3 - e^{-\tau s}(\alpha_2 s^2 + \alpha_1 s + 1)]}, \\ C_2(s) &= \frac{\gamma_d \alpha_2 s^2 + \gamma_p \alpha_1 s + 1}{\alpha_2 s^2 + \alpha_1 s + 1}. \end{aligned} \quad (5.40)$$

In order to designate desired poles for the closed-loop system, α_1 and α_2 are solved to cancel the poles of $P_0(s)$ namely $s = -1/T_1$ and $s = -1/T_2$. This requires that the denominator of $C_1(s)$ in (5.40) have zeros at these two poles, which solves

$$\begin{aligned} \alpha_1 &= \frac{T_2^2[(1 - \lambda_1/T_2)^3 e^{-\tau/T_2} - 1] - T_1^2[(1 - \lambda_1/T_1)^3 e^{-\tau/T_1} - 1]}{T_1 - T_2}, \\ \alpha_2 &= T_1 \alpha_1 + T_1^2[(1 - \lambda_1/T_1)^3 e^{-\tau/T_1} - 1]. \end{aligned} \quad (5.41)$$

If $T_1 = T_2$, α_2 remains the same as that in (5.41) and α_1 is instead solved as

$$\alpha_1 = 2T_1 - [3\lambda_1(1 - \lambda_1/T_1)^2 + (2T_1 + \tau)(1 - \lambda_1/T_1)^3]e^{-\tau/T_1}, \quad (5.42)$$

which is the limit of α_1 in (5.41) as $T_2 \rightarrow T_1$.

With the solved α_1 and α_2 , rewrite $C_1(s)$ in (5.40) as follows

$$C_1(s) = \frac{(\alpha_2 s^2 + \alpha_1 s + 1)(as + 1)}{D(s)}, \quad (5.43)$$

where

$$D(s) := \frac{K[(\lambda_1 s + 1)^3 - e^{-\tau s}(\alpha_2 s^2 + \alpha_1 s + 1)](as + 1)}{(T_1 s + 1)(T_2 s + 1)}. \quad (5.44)$$

Since the denominator of $D(s)$ is cancelled by factors in the numerator due to appropriate α_1 and α_2 , the term $D(s)$ is essentially a polynomial of s with a zero constant if $e^{-\tau s}$ is expanded as a Maclaurin series. That is, $D(s)$ has a form of $\sum_{i=1}^{\infty} \eta_i s^i$, where η_i are proper constants. The values of η_1 and η_2 are of interest for deriving the PID-C controller parameters in (5.37), and they are solved as

$$\begin{aligned} \eta_1 &= \left. \frac{\partial D(s)}{\partial s} \right|_{s=0} = K(3\lambda_1 + \tau - \alpha_1), \\ \eta_2 &= \left. \frac{\partial^2 D(s)}{\partial s^2} \right|_{s=0} = \eta_1(b_0 + a), \end{aligned} \quad (5.45)$$

where

$$b_0 := \frac{3\lambda_1^2 - 0.5\tau^2 + \tau\alpha_1 - \alpha_2}{3\lambda_1 + \tau - \alpha_1} - T_1 - T_2. \quad (5.46)$$

As a consequence, the PID parameters are obtained as

$$K_c = \frac{\alpha_1}{\eta_1}, \quad T_i = \alpha_1, \quad T_d = \frac{\alpha_2}{\alpha_1}. \quad (5.47)$$

And from (5.45) we have

$$b = \eta_2 / \eta_1 = b_0 + a, \quad (5.48)$$

which is a function of the compensator parameter a . Therefore, to derive the compensator parameters a and b , the parameter a has first to be determined. Note

that in (5.43) a is a flexible parameter, and it is intentionally introduced to achieve improved performance compared to the case with $a \equiv 0$. To determine a explicitly, various methods may be used for certain optimizations. The following presents a simple method to determine a .

Noticing the 1/1 Páde expansion that $e^{-\tau s} \approx (1 - 0.5\tau s)/(1 + 0.5\tau s)$, in (5.43) we may take $a := 0.5\tau$ as tends to attain good approximation. The actual a is consequently taken as

$$a = \tilde{a} := \max\{0.5\tau, -b_0\}, \quad (5.49)$$

where ‘max’ is used to ensure a positive b (see (5.48)). The adoption of a in (5.49) has been validated by series of simulations, as referred to next section for examples.

To summarize, the 2DOF-DS approach derives the feedback controller $C_1(s)$ as a PID-C controller in (5.37) for achieving desired closed-loop transfer functions in (5.39), approximately. Of the PID-C controller, the PID parameters are given in (5.47) and the lead-lag compensator parameters are given in (5.48)-(5.49).

With the solved $C_1(s)$, together with the specified $G_{YR}(s)$ in (5.39), the prefilter is derived as

$$C_2(s) = \frac{\gamma_d \alpha_2 s^2 + \gamma_p \alpha_1 s + 1}{\alpha_2 s^2 + \alpha_1 s + 1} e^{-\tau s} \approx \frac{\gamma_d T_i T_d s^2 + \gamma_p T_i s + 1}{T_i T_d s^2 + T_i s + 1} e^{-\tau s}. \quad (5.50)$$

There are two ways to implement the 2DOF controllers consisting of the PID-C feedback controller and the prefilter. One way is to implement them as a setpoint-weighted PID controller in series with a lead-lag compensator, where the PID controller and the compensator are implemented separately. The other way is to implement them as they are, i.e., as the PID-C feedback controller and the prefilter. In the first way, the PID controller requires a filtered derivative action as usual. In the second way, no additional filtering is required; whereas, implementing the designed prefilter is necessary.

Table 5.5 Parameter settings of the PID-C feedback controllers ($C_1(s)$)

$G_{KY}(s)$ ^a	Process model	$\alpha_1,$ $\alpha_2.$	$\frac{\eta_1}{K}$	$\frac{\eta_1}{K}b_0$
A	$\frac{Ke^{-\tau s}}{s}$	$2\lambda_1 + \tau,$ 0.	$\tau\alpha_1 + \lambda_1^2 - \frac{\tau^2}{2}$	$-\frac{\tau^2\alpha_1}{2} + \frac{\tau^3}{6}$
B	$\frac{Ke^{-\tau s}}{s^2}$	$3\lambda_1 + \tau,$ $3\lambda_1^2 + \frac{\tau^2}{2} + 3\tau\lambda_1.$	$\lambda_1^3 + \frac{\tau^3}{6} - \frac{\tau^2\alpha_1}{2} + \tau\alpha_2$	$\frac{\tau^2}{24}(-\tau^2 + 4\tau\alpha_1 - 12\alpha_2)$
B	$\frac{Ke^{-\tau s}}{s(T_1s+1)}$	$3\lambda_1 + \tau,$ $T_1\alpha_1 + T_1^2 \left[(1 - \lambda_1/T_1)^3 e^{-\tau/T_1} - 1 \right].$	$-\frac{\tau^2}{2} + \tau(\alpha_1 - T_1) - \alpha_2$ $+ 3\lambda_1^2 - 3\lambda_1T_1 + \alpha_1T_1$	$\frac{\tau^3}{6} - \frac{\tau^2\alpha_1}{2} + \tau\alpha_2 + \lambda_1^3 - \frac{T_1\eta_1}{K}$
A	$\frac{Ke^{-\tau s}}{T_1s+1}$	$T_1 \left[1 - (\lambda_1/T_1 - 1)^2 e^{-\tau/T_1} \right],$ 0.	$2\lambda_1 + \tau - \alpha_1$	$-\frac{\tau^2}{2} + \alpha_1\tau + \lambda_1^2 - \frac{T_1\eta_1}{K}$
B	$\frac{Ke^{-\tau s}}{(T_1s+1)(T_2s+1)}$	$\frac{\left\{ T_2^2 \left[(1 - \lambda_1/T_2)^3 e^{-\tau/T_2} - 1 \right] \right\}}{T_1 - T_2},$ $\frac{\left\{ -T_1^2 \left[(1 - \lambda_1/T_1)^3 e^{-\tau/T_1} - 1 \right] \right\}}{T_1\alpha_1 + T_1^2 \left[(1 - \lambda_1/T_1)^3 e^{-\tau/T_1} - 1 \right].}$ ^b	$3\lambda_1 + \tau - \alpha_1$	$-\frac{\tau^2}{2} + \alpha_1\tau - \alpha_2 + 3\lambda_1^2 - \frac{(T_1+T_2)\eta_1}{K}$

^a A and B denote the desired transfer functions and they are specified as $A := (\alpha_1s+1)/(\lambda_1s+1)^2$ and

$$B := (\alpha_2s^2 + \alpha_1s + 1)/(\lambda_1s+1)^3.$$

^b If $T_1 = T_2$, then $\alpha_1 = 2T_1 - [3\lambda_1(1 - \lambda_1/T_1)^2 + (2T_1 + \tau)(1 - \lambda_1/T_1)^3]e^{-\tau/T_1}$ and α_2 keeps the same.

Remark 5.4 If the numerators of λ_1s+1 with power four are used in (5.39), then a new PID-C controller will be obtained as the same as that reported in [86].

Similarly, the PID-C controllers for other process models can be obtained. The results are summarized in Table 5.5, which give explicit expressions of the intermediate variables for deriving the PID-C parameters. (The expressions were obtained by using the symbolic Toolbox in MATLAB (version R2006a).) With the intermediate variables, the PID-C parameters for an FOPTD process are obtained as

$$K_c = \frac{\alpha_1 + \tilde{a}}{\eta_1}, T_i = \alpha_1 + \tilde{a}, T_d = \frac{\alpha_1 \tilde{a}}{\alpha_1 + \tilde{a}}, \quad (5.51)$$

$$a = 0, b = b_0 + \tilde{a}.$$

And the PID-C parameters for an SOPTD process are obtained as

$$K_c = \frac{\alpha_1}{\eta_1}, T_i = \alpha_1, T_d = \frac{\alpha_2}{\alpha_1}, \quad (5.52)$$

$$a = \tilde{a}, b = b_0 + a.$$

In (5.51)-(5.52), the $\tilde{a}$'s are both given in (5.49).

As in the PID case, the PID-C settings may also apply to unstable processes by replacing the parameter K or T_1 or T_2 in the table with $-K$ or $-T_1$ or $-T_2$, respectively. The applicability, however, is restricted since the approximation or model errors may become unignorable and cause instability when the process is unstable.

5.5 Numerical Examples

In this section, simulation results with various processes are presented to validate the proposed 2DOF PI/PID and PID-C controllers, and the results are compared with those obtained with recent methods. The process models with lag dominated ($\min\{T_1, T_2\}/\tau > 1$), lag-delay balanced ($\min\{T_1, T_2\}/\tau = 1$) and delay dominated ($\min\{T_1, T_2\}/\tau < 1$) are considered. PID controllers with filtered derivative modes are applied in all the simulations, that is, the PID controllers take the form of

$$C_1(s) = K_c \left(1 + \frac{1}{T_i s} + \frac{T_d}{\mu T_d s + 1} s \right), \quad (5.53)$$

where μ is a scalar selected from $[0.1, 0.2]$ [6, 80]. To avoid biasing much the ideal design, $\mu = 0.05$ is used. And for consistent and fair comparisons, setpoint weights with $\gamma_p = 0.4$ (as suggested in [7-8, 81, 89]) and $\gamma_d = 1.0$ are applied to any PI/PID or PID-C tuning methods in the simulations.

The PI/PID or PID-C designs with different methods are tuned to achieve the same peak sensitivity. And two indices are calculated to evaluate the performance [6]:

$$\text{IAE} := \int_0^{\infty} |r(t) - y(t)| dt, \text{ and} \quad (5.54)$$

$$\text{TV} := \sum_{k=1}^{\infty} |u(k+1) - u(k)|. \quad (5.55)$$

IAE (integrated absolute error) measures the deviations of the output from the given setpoint, and TV (total variation) measures the ‘smoothness’ of the control signal $u(t)$. For best performance, these two indices should both be as small as possible.

Further, to visually show the performance differences, normalized IAE’s and TV’s and a comprehensive index ρ are defined:

$$\begin{aligned} \overline{\text{IAE}}_s &:= \frac{\text{IAE}_s}{\text{IAE}_s^{\circ}}, \quad \overline{\text{TV}}_s := \frac{\text{TV}_s}{\text{TV}_s^{\circ}}, \\ \overline{\text{IAE}}_d &:= \frac{\text{IAE}_d}{\text{IAE}_d^{\circ}}, \quad \overline{\text{TV}}_d := \frac{\text{TV}_d}{\text{TV}_d^{\circ}}, \\ \rho &:= \frac{1}{4} (\overline{\text{IAE}}_s + \overline{\text{TV}}_s + \overline{\text{IAE}}_d + \overline{\text{TV}}_d), \end{aligned} \quad (5.56)$$

where the footnote ‘_s’ means for the step setpoint response and ‘_d’ for the step load disturbance response when certain tuning method is applied; and IAE_s° , TV_s° , IAE_d° and TV_d° are performance indices attained by a reference tuning method. The smaller an index is, the better the performance is in the sense of the particular index. If an index value is larger than 1.0 then the performance is interpreted as being worse than that attained by the reference method, regard to this particular index; and vice versa. On statistics of the results, if a method does not apply to a process model or if it applies but fails to give a feasible control (i.e., the closed-loop system is unstable), the indices above are defined as infinity and denoted by a maximal value of 2.0. To differentiate a large value from infinity, any index values larger than 1.2 are normalized as $1.2 + 0.6M/M^{\max}$. Here M is any index

value calculated in (5.56) and $M^{\max}$ is the largest M among the M 's computed for different methods as applied to the same process.

The above constitutes complete definitions of the performance indices to be used. The simulation results are compared for PI, PID and PID-C controls, respectively.

5.5.1 PI Control

The PI tuning rules obtained with Method 1 and Method 2 are named as 'Prop. 1' and 'Prop. 2', respectively. They are compared with the rules proposed by Skogestad in [6] (named as 'SIMC') and by Chen and Seborg in [7] (named as 'C-S'). Simulations are carried out on IPTD and FOPTD processes with different parameter configurations and the results are summarized in Table 5.6, where the setpoint references are unit-step signals and the load disturbances are step signals with magnitudes of d_i as indicated in the table. The performance indices are calculated and shown in Figure 5.2. The results indicate that the proposed tuning rules with Method 1 and Method 2 are both applicable to all the exemplary processes but neither are SIMC nor C-S rules. Overall the proposed rules achieve most competitive performances, resulting in minimum ρ in almost each case. Note that such gains in performance are at a cost of robustness, which is indicated by higher peak complementary sensitivities (M_r 's) as compared to those attained by SIMC rules. The rules, however, almost always enable larger robustness margins (as indicated by smaller values of M_r) while achieving similar performances as compared to C-S rules. For each of the exemplary processes, the PI tuning rules with Method 1 and Method 2 achieve similar performance and robustness and it seems that neither method is obviously advantageous than the other. Typical simulation results are shown in Figure 5.3.

Figure 5.2 Performance index values attained with different PI tuning rules.

Figure 5.3 Output responses of processes and PI controllers for processes E2 and E4 in Table 5.6, subject to unit-step inputs and step disturbances with magnitudes of -0.3 and -2.0, respectively.

Table 5.6 PI controller settings and performance summary for exemplary processes.

Process	Method	M_s	M_t	λ_1	K_c	T_i	Setpoint		Disturbance		
							IAE	TV	d_i	IAE	TV
E1: $\frac{e^{-0.2s}}{s}$	Prop. 1	2.0	1.57	0.411	2.892	1.022	0.61	2.35	-1.0	0.35	1.89
	Prop. 2		1.57	0.411	2.892	1.022	0.61	2.35		0.35	1.89
	SIMC		1.46	0.126	3.068	1.304	0.78	2.28		0.43	1.79
	C-S		2.00	0.384	2.839	0.968	0.58	2.39		0.34	1.91
E2: $\frac{e^{-s}}{s}$	Prop. 1	2.0	1.57	2.055	0.578	5.111	3.07	0.47	-0.2	1.77	0.38
	Prop. 2		1.57	2.055	0.578	5.111	3.07	0.47		1.77	0.38
	SIMC		1.46	0.630	0.614	6.520	3.87	0.46		2.12	0.36
	C-S		2.00	1.919	0.568	4.838	2.90	0.48		1.70	0.38
E3: $\frac{e^{-5s}}{s}$	Prop. 1	2.0	1.57	10.276	0.116	25.553	15.32	0.09	-0.05	11.00	0.09
	Prop. 2		1.57	10.276	0.116	25.553	15.32	0.09		11.00	0.09
	SIMC		1.46	3.150	0.123	32.600	19.35	0.09		13.19	0.09
	C-S		2.01	9.594	0.114	24.188	14.51	0.10		10.60	0.10
E4: $\frac{e^{-0.2s}}{s+1}$	Prop. 1	2.0	1.36	0.306	3.204	0.662	0.60	2.47	-1.0	0.21	1.62
	Prop. 2		1.39	0.306	3.126	0.616	0.57	2.55		0.20	1.65
	SIMC		1.26	0.085	3.506	1.000	0.89	2.33		0.29	1.52
	C-S		2.00	0.326	3.338	0.769	0.69	2.38		0.23	1.57
E5: $\frac{e^{-s}}{s+1}$	Prop. 1	2.0	1.00	0.776	0.822	1.291	2.35	1.06	-1.0	1.59	1.43
	Prop. 2		1.00	0.893	0.807	1.244	2.29	1.09		1.58	1.44
	SIMC		1.26	0.426	0.701	1.000	2.33	1.32		1.71	1.52
	C-S		--	--	--	--	--	--		--	--
E6: $\frac{e^{-5s}}{s+1}$	Prop. 1	2.0	1.00	1.517	0.389	2.738	8.91	1.02	-1.0	7.88	1.32
	Prop. 2		1.00	2.403	0.391	2.755	8.91	1.01		7.87	1.32
	SIMC		1.26	2.131	0.140	1.000	11.31	1.46		10.69	1.51
	C-S		--	--	--	--	--	--		--	--
E7: $\frac{e^{-0.2s}}{s-1}$	Prop. 1	2.0	2.01	0.784	2.622	2.935	0.64	2.41	-1.0	1.12	2.61
	Prop. 2		2.07	0.732	2.568	2.726	0.67	2.50		1.06	2.65
E8: $\frac{e^{-0.4s}}{s-1}$	Prop. 1	3.2	3.33	1.693	1.646	9.930	1.97	2.66	-1.0	6.03	4.70
	Prop. 2		3.62	1.610	1.549	10.217	2.43	2.74		6.60	4.93

5.5.2 PID Control

The PID tuning rules obtained with Method 1 and Method 2 are named as ‘Prop. 1’ and ‘Prop. 2’, respectively. They are compared with the rules proposed by Skogestad in [6] (named as ‘SIMC’), Chen and Seborg in [7] (named as ‘C-S’), and Shamsuzzoha and Lee in [89] (denoted as ‘S-L’). Simulations are carried out on various processes and the results are summarized in Table 5.7, where the setpoint references are unit-step signals and the load disturbances are step signals with magnitudes of d_i as indicated in the table. The resulting performance indices are computed and shown in Figure 5.4. The results indicate that the proposed rules with either method and the S-L rules are applicable to most of the processes and give most competitive performances while achieving similar robustness in terms of peak sensitivities and complementary sensitivities (M_s ’s and M_t ’s), and the best tuning rule depends on the process in face. Overall the proposed rules with Method 2 lead to smallest peak complementary sensitivities when the same peak sensitivities are attained, implying the most robust controls the rules can provide. Typical simulation results that produce Figure 5.4 are shown in Figures 5.5-5.7, which include step setpoint and disturbance responses.

Note that the PID settings obtained by S-L rules were optimal IMC-PID settings [89]. The above simulation results imply that in many cases the proposed methods work as well as or even better than the optimal IMC-PID rules. This justifies the efficiency of the proposed rules. In addition, numerical results (not shown for brevity) indicate that for delay dominated or open-loop unstable processes it is difficult for the proposed PID tuning rules to give robust closed-loop performance and stability. For these challenging cases, other realizations of the ideal feedback controllers have to be explored, or more sophisticated control strategies have to be considered [3, 78].

Figure 5.4 Performance index values attained with different PID tuning rules.

Figure 5.5 Output responses of processes and PID controllers for processes E5 and E8 in Table 5.7, subject to unit-step inputs and step disturbances with magnitudes of -1.0 and -0.2, respectively.

Figure 5.6 Output responses of processes and PID controllers for processes E12 and E15 in Table 5.7, subject to unit-step inputs and step disturbances with magnitudes of -0.1 and -1.0, respectively.

Figure 5.7 Output responses of processes and PID controllers for processes E18 and E20 in Table 5.7, subject to unit-step inputs and step disturbances with magnitudes of -1.0 and -8.0, respectively.

Table 5.7 PID controller settings and performance summary for exemplary processes.

Process	Rule	M_s	M_t	λ_1	K_c	T_i	T_d	Setpoint		Disturbance		
								IAE	TV	d_1	IAE	TV
E1: $\frac{e^{-0.2s}}{s}$	Prop. 1	2.0	1.50	0.254	4.174	0.774	0.063	0.47	107.11		0.19	1.88
	Prop. 2		1.41	0.239	4.269	0.817	0.075	0.49	116.54	-1	0.19	1.86
	C-S		1.43	0.236	4.266	0.807	0.072	0.49	114.74		0.19	1.86
	S-L		1.53	0.292	4.255	0.688	0.072	0.42	115.23		0.17	1.96
E2: $\frac{e^{-s}}{s}$	Prop. 1	2.0	1.51	1.262	0.834	3.821	0.313	2.29	21.32		4.59	1.90
	Prop. 2		1.41	1.195	0.854	4.084	0.377	2.45	23.26	-1	4.79	1.86
	C-S		1.43	1.179	0.853	4.036	0.361	2.42	22.92		4.73	1.87
	S-L		1.54	1.452	0.850	3.401	0.360	2.08	22.81		4.12	1.97
E3: $\frac{e^{-5s}}{s}$	Prop. 1	2.0	1.56	6.182	0.165	17.979	1.543	10.82	4.20		10.95	0.19
	Prop. 2		1.41	5.973	0.171	20.419	1.886	12.25	4.67	-0.1	11.94	0.19
	C-S		1.43	5.894	0.171	20.181	1.803	12.11	4.60		11.80	0.19
	S-L		1.60	7.080	0.169	16.155	1.750	10.04	4.49		10.05	0.20
E4: $\frac{e^{-0.2s}}{s+1}$	Prop. 1	2.0	1.33	0.195	4.483	0.542	0.063	0.45	117.82		0.12	1.70
	Prop. 2		1.20	0.210	4.503	0.598	0.078	0.49	127.68	-1	0.13	1.69
	C-S		1.23	0.205	4.533	0.587	0.073	0.48	125.82		0.13	1.67
	S-L		1.35	0.230	4.509	0.508	0.067	0.42	120.83		0.11	1.74
E5: $\frac{e^{-s}}{s+1}$	Prop. 1	2.0	1.10	0.523	1.157	1.306	0.289	1.92	32.26		1.13	1.58
	Prop. 2		1.00	0.779	1.119	1.498	0.374	2.24	34.23	-1	1.34	1.71
	C-S		1.00	0.734	1.158	1.449	0.319	2.12	33.49		1.25	1.61
	S-L		1.11	0.604	1.155	1.287	0.288	1.89	32.19		1.12	1.58
E6: $\frac{e^{-5s}}{s+1}$	Prop. 1	1.75	1.00	1.382	0.418	2.826	0.744	8.46	11.86		6.88	1.32
	Prop. 2		1.00	2.581	0.484	3.342	1.253	9.02	19.92	-1	6.97	2.53
	C-S		1.00	2.682	0.326	2.594	0.090	9.51	7.55		7.98	1.07
	S-L		1.00	1.285	0.407	2.788	0.709	8.53	11.32		6.97	1.27
E7: $\frac{e^{-0.2s}}{s^2}$	Prop. 1 or S-L	3.0	2.71	0.301	5.899	1.428	0.580	1.34	182.68		0.24	4.76
	Prop. 2		2.61	0.481	5.858	1.644	0.610	1.44	190.65	-1	0.28	4.85
	SIMC		0.58	0.091	5.894	2.330	0.583	1.79	186.53		0.39	4.72
	C-S		--	--	--	--	--	--	--		--	--
E8: $\frac{e^{-s}}{s^2}$	Prop. 1 or S-L	3.0	2.75	1.519	0.235	7.187	2.857	6.64	7.15		6.16	0.94
	Prop. 2		2.61	2.406	0.234	8.218	3.052	7.17	7.60	-0.2	7.05	0.97
	SIMC		2.91	0.456	0.236	11.651	2.913	8.94	7.45		9.87	0.94
	C-S		--	--	--	--	--	--	--		--	--
E9: $\frac{e^{-5s}}{s^2}$	Prop. 1 or S-L	3.0	2.97	8.241	0.009	37.776	13.335	33.76	0.26		42.02	0.05
	Prop. 2		2.54	12.030	0.009	41.089	15.262	35.33	0.28	-0.01	45.75	0.04
	SIMC		2.57	2.282	0.009	58.254	14.563	44.1	0.27		64.64	0.04
	C-S		--	--	--	--	--	--	--		--	--
E10: $\frac{e^{-0.2s}}{s(s+1)}$	Prop. 1 or S-L	2.5	2.11	0.277	7.820	1.328	0.418	0.95	197.17	-1	0.17	3.06
	Prop. 2		1.99	0.450	7.713	1.551	0.450	1.08	202.03		0.20	3.05

	SIMC		1.87	0.068	7.209	2.072	0.517	1.39	201.62		0.29	3.08
	C-S		2.12	0.398	7.818	1.395	0.413	0.98	196.75		0.18	3.05
E11: $\frac{e^{-s}}{s(s+1)}$	Prop. 1 or S-L		1.89	0.900	0.889	4.728	0.908	2.86	19.76		2.68	1.24
	Prop. 2	2.5	1.77	1.487	0.887	5.460	0.995	3.28	20.18	-0.5	3.08	1.22
	SIMC		1.87	0.340	0.885	6.362	0.843	3.82	19.54		3.59	1.22
	C-S		2.04	1.236	0.842	4.708	0.786	2.84	18.15		2.80	1.27
	Prop. 1 or S-L		2.00	2.799	0.170	16.357	1.820	9.88	3.60		10.17	0.24
E12: $\frac{e^{-5s}}{s(s+1)}$	Prop. 2	2.5	1.70	4.671	0.182	19.014	2.305	11.42	4.00	-0.1	10.46	0.22
	SIMC		1.88	1.702	0.155	27.809	0.964	16.59	3.24		17.87	0.23
	C-S		2.09	5.010	0.120	20.029	0.414	14.84	2.54		18.67	0.25
	Prop. 1 or S-L		1.64	0.299	14.254	1.310	0.387	0.88	327.97		0.92	21.01
E13: $\frac{e^{-0.2s}}{(s+1)(2s+1)}$	Prop. 2	2.0	1.47	0.492	13.881	1.563	0.435	1.06	330.87	-10	1.13	20.12
	SIMC		1.26	0.085	10.518	3.000	0.667	2.10	285.92		2.85	20.06
	C-S		1.57	0.442	14.224	1.426	0.400	0.96	331.24		1.00	20.61
	Prop. 1 or S-L		1.24	0.733	2.115	2.793	0.725	3.00	45.01		2.65	3.14
E14: $\frac{e^{-s}}{(s+1)(2s+1)}$	Prop. 2	2.0	1.15	1.227	2.076	3.159	0.826	3.42	44.94	-2	3.05	3.08
	SIMC		1.26	0.426	2.104	3.000	0.667	3.23	44.47		2.85	3.16
	C-S		1.34	0.982	2.042	2.648	0.622	2.89	42.67		2.61	3.27
	Prop. 1 or S-L		1.00	1.168	0.703	4.696	1.473	9.54	15.63		6.95	1.32
E15: $\frac{e^{-5s}}{(s+1)(2s+1)}$	Prop. 2	2.0	1.00	2.403	0.704	5.042	1.692	10.23	15.92	-1	7.28	1.39
	SIMC		1.26	2.131	0.421	3.000	0.667	11.68	9.68		10.19	1.52
	C-S		1.26	1.708	0.409	2.937	0.639	11.83	9.45		10.34	1.52
	Prop. 1		1.82	0.374	3.832	1.370	0.062	0.47	95.88		0.36	2.27
E16: $\frac{e^{-0.2s}}{s-1}$	Prop. 2	2.0	1.80	0.293	3.932	1.314	0.067	0.46	101.66	-1	0.33	2.25
	S-L		1.84	0.401	4.020	1.055	0.086	0.40	112.66		0.28	2.31
	Prop. 1		2.44	0.509	2.613	2.561	0.153	0.67	79.38		0.98	3.61
E17: $\frac{e^{-0.4s}}{s-1}$	Prop. 2	3.0	2.25	0.495	2.632	2.718	0.167	0.71	84.15	-1	1.03	3.57
	S-L		2.34	0.654	2.528	2.110	0.209	0.68	92.15		0.97	3.88
	Prop. 1		1.97	0.400	9.120	2.009	0.548	1.14	220.2		1.10	12.55
E18: $\frac{e^{-0.2s}}{(s+1)(2s-1)}$	Prop. 2	2.0	1.71	0.656	9.130	2.435	0.613	1.36	231.1	-5	1.33	11.9
	S-L		1.94	0.381	9.141	2.160	0.547	1.21	221.26		1.18	12.34
	Prop. 1		2.34	0.637	4.062	3.942	0.763	1.94	98.08		0.97	3.09
E19: $\frac{e^{-0.4s}}{(s+1)(2s-1)}$	Prop. 2	2.5	2.10	1.018	4.137	4.556	0.823	2.17	104.01	-1	1.10	3.07
	S-L		2.36	0.698	4.082	3.479	0.778	1.85	98.99		0.89	3.13
	Prop. 1 or S-L	2.2	1.00	0.261	15.177	1.065	0.667	1.18	529.71	-10	0.70	41.19
E20: $\frac{e^{-0.1s}}{(s-1)(2s-1)}$	Prop. 2		1.00	0.433	15.551	1.315	0.717	1.33	602.70		0.85	42.32

Notes on Tables 5.6-5.7. *i)* If a PI/PID tuning method does not apply, it is omitted for comparison; *ii)* if a method applies but fails to give a feasible control the notations, ‘--’s, are used to indicate the results; *iii)* in the case of PID control, some of the processes are approximated by other processes so that the tuning methods can be applied. Specifically, E1-3 are approximated by $100e^{-\tau s}/(100s+1)$ when applying Prop. 1 and S-L rules, E7-9 by $100e^{-\tau s}/[s(100s+1)]$ when applying C-S rules and by $10000e^{-\tau s}/[(100s+1)(100s+1)]$ when applying Prop. 1 and S-L¹ rules, and E10-12 by $100e^{-\tau s}/[(100s+1)(s+1)]$ when applying Prop. 1 and S-L rules; *iv)* the stabilizability of unstable processes with time delay by PI/PID control is considered in preparing the exemplary processes in Tables 5.6-5.7, of which the conditions on the process parameters are referred to [90].

5.5.3 PID-C Control

The proposed PID-C control are compared with the proposed PID control (with Method 1) and recent PID-C controls. Recently, Rao, et al., derived tuning rules for PID-C control of a class of integrating processes [8]. The rules are named as ‘R-R-C’ (using acronyms of the author names) for short. Shamsuzzoha and Lee derived PID-C tuning rules for IPTD [10], stable/unstable FOPTD [87], and stable/unstable SOPTD processes [86]. These rules altogether are named as ‘S-L’. The proposed PID control with Method 1 keeps the name of ‘Prop. 1’. And the proposed PID-C tuning rules are named as ‘Prop. 3’. The PID control is implemented as that in (5.53) with setpoint weighting, and the PID-C control is implemented as a weighted PID control given in (5.53) in series with a lead-lag compensator.

¹ The case of $T_1 = T_2$ was skipped in the original paper. But it can be handled, where the intermediate variables, α_1 and α_2 , are solved as the limits of the given expressions as $T_2 \rightarrow T_1$.

Simulations are carried out for all the processes given in Table 5.7. The controller parameters and simulation results are summarized in Table 5.8, where the setpoint references and load disturbances are the same as those for obtaining Table 5.7. And the resulting performance indices are computed and shown in Figure 5.8. The results indicate that: *i)* R-R-C rules are limited to integrating processes and achieve performances similar to those achieved by the proposed PID-C rules; *ii)* S-L rules achieve better setpoint and disturbance responses as compared to the proposed PID-C rules in the cases of $E\{1-6, 16-17\}$, at costs of more control efforts; *iii)* Prop. 1 rules attain similar performance to S-L rules in each case except that their disturbance responses are a bit worse than those with S-L rules in general (while, as tradeoffs the Prop. 1 rules require less control efforts); and *iv)* overall, the performances attained by R-R-C, S-L, and Prop. 1 controls fluctuate around those attained by the proposed PID-C controls, and the superiority of the setpoint or disturbance response is usually gained at a cost of more control efforts. The results also indicate that the proposed PID-C rules always lead to the most robust control, in terms of the lowest peak complementary sensitivities (M_t 's), in comparison with the other rules. Typical responses of the tested processes are shown in Figure 5.9, as simulated with the controller settings given in Table 5.8.

To summarize, the proposed PID-C control can achieve similar performance with enhanced robustness as compared to the recent PID-C controls and the proposed PID control. The proposed PID-C control can be good alternates for control of processes in practice. The results summarized in Table 5.8 and shown in Figure 5.8 can be referred to when engineers are selecting tuning rules in control design.

Figure 5.8 Performance index values attained with different PID-C rules.

Figure 5.9 Setpoint and disturbance responses attained with different PID-C/PID rules. The R-R-C rules apply only to E{2, 8, 11} and achieve similar responses to the Prop. 3 rules.

Table 5.8 PID-C controller settings and performance summary for exemplary processes.

Process	Rule	M_s	M_t	λ_1	K_c	T_i	T_d	a	b	Setpoint		Disturbance	
										IAE	TV	IAE	TV
E1	Prop. 3	2.0	1.43	0.321	3.749	0.941	0.089	0	0.038	0.58	15.33	0.25	1.77
	R-R-C		1.44	0.324	3.721	0.949	0.089	0	0.041	0.58	14.27	0.25	1.78
	S-L		1.72	0.133	0.873	0.133	0.050	0.600	0.079	0.55	327.79	0.16	2.15
	Prop. 1		1.50	0.254	4.174	0.774	0.063	0	0	0.47	107.11	0.19	1.88
E2	Prop. 3	2.0	1.43	1.603	0.750	4.707	0.447	0	0.191	2.83	3.07	6.27	1.77
	R-R-C		1.43	1.622	0.744	4.744	0.447	0	0.206	2.86	2.85	6.38	1.78
	S-L		1.73	0.659	0.175	0.667	0.250	3.016	0.391	2.73	65.43	3.90	2.17
	Prop. 1		1.51	1.262	0.834	3.821	0.313	0	0	2.29	21.32	4.59	1.90
E3	Prop. 3	2.0	1.43	8.017	0.150	23.533	2.234	0	0.957	14.13	0.61	15.69	0.18
	R-R-C		1.43	8.109	0.149	23.719	2.236	0	1.032	14.24	0.57	15.92	0.18
	S-L		1.81	3.223	0.035	3.333	1.250	15.622	1.886	13.85	13.07	9.59	0.23
	Prop. 1		1.56	6.182	0.165	17.979	1.543	0	0	10.82	4.20	10.95	0.19
E4	Prop. 3	1.5	1.05	0.430	2.560	0.834	0.088	0	0.056	0.84	6.32	0.33	1.12
	S-L		1.11	0.347	0.411	0.100	0.033	0.651	0.048	0.50	258.98	0.24	1.25
	Prop. 1		1.09	0.373	2.717	0.727	0.044	0	0	0.72	116.83	0.27	1.19
E5	Prop. 3	1.5	1.00	1.061	0.706	1.499	0.333	0	0.265	3.03	1.6	2.12	1.00
	S-L		1.00	0.840	0.296	0.500	0.167	0.991	0.208	2.01	65.43	1.69	1.09
	Prop. 1		1.00	0.925	0.684	1.267	0.198	0	0	2.62	29.9	1.85	1.01
E6	Prop. 3	1.75	1.00	2.040	0.432	3.493	0.711	0	1.083	10.19	1.03	8.19	1.08
	S-L		1.00	1.086	0.405	2.500	0.833	1.000	0.476	8.05	61.19	6.36	3.78
	Prop. 1		1.00	1.383	0.418	2.826	0.744	0	0	8.47	20.38	6.88	1.32
E7	Prop. 3	2.5	1.87	0.405	7.439	1.414	0.533	0.100	0.030	1.14	1193.3	0.19	3.18
	R-R-C		1.91	0.412	7.226	1.437	0.541	0.100	0.035	1.17	994.07	0.20	3.23
	S-L		2.31	0.305	5.678	1.420	0.576	0.100	0.076	1.22	354.21	0.25	3.53
	Prop. 1		2.33	0.345	4.748	1.593	0.647	0	0	1.35	232.00	0.34	3.62
E8	Prop. 3	2.5	1.88	2.023	0.298	7.069	2.666	0.500	0.151	5.68	47.57	4.75	0.64
	R-R-C		1.91	2.062	0.289	7.185	2.705	0.500	0.176	5.79	39.48	4.98	0.65
	S-L		2.36	1.574	0.220	7.291	2.878	0.500	0.402	6.16	12.96	6.64	0.71
	Prop. 1		2.38	1.749	0.189	8.038	3.144	0	0	6.69	9.15	8.52	0.72
E9	Prop. 3	2.5	1.88	10.114	0.012	35.343	13.330	2.500	0.756	28.43	1.92	29.51	0.03
	R-R-C		1.94	10.309	0.012	35.926	13.526	2.500	0.881	29.36	1.66	30.02	0.03
	S-L		2.62	13.953	0.004	59.276	19.270	2.500	4.087	43.99	0.12	148.03	0.04
	Prop. 1		2.56	9.956	0.007	43.840	14.891	0	0	34.34	0.33	62.62	0.04
E10	Prop. 3	2.5	1.75	0.332	10.897	1.196	0.368	0.100	0.029	0.87	1792.49	0.11	2.77
	R-R-C		1.79	0.337	10.658	1.212	0.371	0.100	0.034	0.88	1486.72	0.11	2.81
	S-L		2.08	0.236	9.535	1.143	0.369	0.100	0.063	0.86	685.75	0.12	2.98
	Prop. 1		2.11	0.277	7.820	1.328	0.418	0	0	0.96	356.57	0.17	3.06
E11	Prop. 3	2.5	1.68	1.074	1.066	4.223	0.763	0.5	0.135	2.56	182.32	1.98	1.16
	R-R-C		1.71	1.089	1.051	4.268	0.766	0.5	0.159	2.58	150.51	2.03	1.17
	S-L		1.86	0.777	0.995	4.066	0.754	0.5	0.253	2.5	87.34	2.08	1.20

	Prop. 1		1.89	0.900	0.889	4.728	0.908	0	0	2.87	37.89	2.68	1.24
	Prop. 3		1.78	3.870	0.166	16.610	0.930	2.500	0.984	9.99	17.68	10.01	0.22
E12	R-R-C	2.5	1.76	4.388	0.178	18.163	1.862	2.500	2.065	10.91	9.07	10.21	0.22
	S-L		1.85	2.918	0.153	15.651	0.942	2.500	1.465	10.68	12.28	11.04	0.22
	Prop. 1		2.00	2.799	0.170	16.357	1.82	0	0	9.89	7.07	10.17	0.24
	Prop. 3		1.42	0.391	17.891	1.301	0.373	0.100	0.039	0.88	2161.21	0.73	19.23
E13	S-L	2.0	1.64	0.287	15.03	1.263	0.376	0.100	0.088	0.86	740.97	0.84	20.95
	Prop. 1		1.64	0.299	14.254	1.310	0.387	0	0	0.89	618.45	0.92	21.01
	Prop. 3		1.14	0.930	2.263	2.629	0.620	0.500	0.165	2.75	304.78	2.32	2.92
E14	S-L	2.0	1.25	0.672	2.191	2.533	0.607	0.500	0.275	2.69	170.8	2.31	3.08
	Prop. 1		1.24	0.733	2.115	2.793	0.725	0	0	3.01	88.15	2.65	3.14
	Prop. 3		1.00	1.601	0.440	2.996	0.666	2.500	0.701	8.70	66.66	7.13	1.31
E15	S-L	2.0	1.09	1.207	0.438	2.992	0.666	2.500	0.846	8.88	54.66	7.34	1.33
	Prop. 1		1.00	1.168	0.703	4.696	1.473	0	0	9.55	29.97	6.95	1.32
	Prop. 3		1.86	0.306	4.364	1.182	0.092	0	0.030	0.45	24.54	0.27	2.58
E16	S-L	2.5	2.24	0.107	0.985	0.133	0.050	0.655	0.061	0.59	614.35	0.14	3.46
	Prop. 1		1.92	0.235	4.872	0.940	0.072	0	0	0.38	238.73	0.19	2.69
	Prop. 3		2.42	0.679	2.353	3.404	0.188	0	0.051	0.78	16.41	1.45	3.57
E17	S-L	3.0	2.83	0.211	0.433	0.267	0.100	1.647	0.109	1.46	435.41	0.65	5.20
	Prop. 1		2.44	0.509	2.613	2.561	0.153	0	0	0.68	132.73	0.98	3.61
	Prop. 3		1.65	0.503	12.705	1.856	0.515	0.100	0.041	1.11	1425.81	0.73	11.52
E18	S-L	2.0	1.96	0.453	7.760	2.307	0.598	0.100	0.128	1.27	273.49	1.49	12.47
	Prop. 1		1.97	0.400	9.120	2.009	0.548	0	0	1.15	406.31	1.1	12.55
	Prop. 3		1.89	0.714	5.868	3.064	0.679	0.200	0.060	1.71	927.19	0.52	2.86
E19	S-L	2.5	2.31	0.558	4.666	3.351	0.709	0.200	0.151	1.79	277.03	0.72	3.05
	Prop. 1		2.34	0.637	4.062	3.942	0.763	0	0	1.95	180.96	0.97	3.09
	Prop. 3		1.00	0.260	36.827	0.857	0.413	0.05	0.017	0.81	5274.7	0.23	37.42
E20	S-L	2.4	1.00	0.281	13.389	1.138	0.734	0.05	0.064	1.34	653.38	0.86	48.13
	Prop. 1		1.00	0.232	18.932	0.974	0.581	0	0	1.09	1062.78	0.52	45.02

Notes on Table 5.8: *i)* A method is omitted if it does not apply; *ii)* if a method applies but fails to give a feasible control, the notations ‘--’s are used to indicate the results; *iii)* some of the processes are approximated by other processes so that the S-L and Prop. 3 rules are applicable. Specifically, in order to apply the S-L and Prop. 3 rules, E1-3, E7-9 and E10-12 are approximated by $100e^{-\tau s}/(100s+1)$, $10000e^{-\tau s}/[(100s+1)(100s+1)]$, $100e^{-\tau s}/[(100s+1)(s+1)]$ and $100e^{-\tau s}/[(100s+1)(s-1)]$, where τ ’s are the time delays.

5.6 Conclusions

Principles of designing 2DOF controllers by DS approach were presented. Based on these principles, explicit tuning formulas for PI/PID and PID-C controllers as feedback controllers were respectively obtained based on typical process models. For simplicity, the prefilter is implemented approximately as setpoint weighting on the PI/PID and PID-C controller. The derived rules may apply to unstable processes. The application, however, is limited by the approximation errors involved in the design. A series of numerical examples demonstrated the usefulness of the proposed 2DOF PI/PID and PID-C controller design.

Chapter 6

Analytical PI Controller Tuning Using Closed-loop Setpoint Response

Recently Shamsuzzoha and Skogestad proposed a PI tuning rule for a wide range of unidentified processes. The rule relies on a CSR of a process and was developed from extensive numerical experiments. This chapter analytically derives a similar PI tuning rule using the CSR method. Simulations indicate that the two rules perform similarly if the tuning parameter is selected properly for the analytical rule. Meanwhile, a guideline is proposed for choosing the P controller gain for the CSR experiment to result in a proper overshoot for obtaining good PI settings. Numerical examples are used to demonstrate the usefulness of the theoretical results.

6.1 Introduction

PI controller tuning has been extensively studied in the last decades, generating a large number of PI tuning rules [1-3, 78]. Conventional PI controller tuning, however, requires trials and may experience instability during tuning experiment or process modeling, and the resulting closed-loop performance is satisfactory only for particular classes of processes [11]. To overcome these problems, recently Shamsuzzoha and Skogestad proposed using CSR to set the PI parameters, which requires a single closed-loop experiment and gives fast and robust performance for a broad range of processes typical for process control [11]. An earlier CSR method was considered by Yuwana and Seborg in

1982 [92], but their method leads to a more complicated solution.

In the CSR method, one carries out a closed-loop experiment with a single P controller and then utilizes the response information to derive the PI settings. Shamsuzzoha and Skogestad considered a special case of the SIMC tuning rule with its single tuning parameter, the closed-loop time constant τ_c , set as $\tau_c = \theta$, where θ is the effective time delay of a process [6, 25]. They derived a PI tuning rule by relating the closed-loop response quantities with the SIMC settings, including the peak time, overshoot and steady-state offset. The resulting PI tuning rule was tested on a broad range of processes and demonstrated to give comparable performance as the SIMC tuning rule.

While Shamsuzzoha and Skogestad developed the PI tuning rule from series of numerical experiments, analytical derivation of a similar rule using CSR method is of interest in this chapter. The derivation is based on an IPTD process and then extended to an FOPTD process. The main idea is as follows. With the CSR method, a single P controller is applied to the process and a step test of setpoint change is performed. From the closed-loop response, the steady-state offset, peak time, and overshoot *or* rise time are recorded. These quantities, together with the applied proportional gain and setpoint change, are used to estimate the process parameters and consequently express the SIMC tuning rule in a new manner. The resulting PI tuning rule has a single tuning parameter α which controls the trade-off between performance and robustness. This rule is tested on a broad range of processes typical for process control applications. The results indicate that the analytical rule gives comparable performance to Shamsuzzoha-Skogestad's PI tuning rule [11] if the detuning parameter α is chosen properly. In a sense, the analysis and derived rule provide some kind of insight and support to the PI tuning rule proposed by Shamsuzzoha and Skogestad.

6.2 Derivation of the PI Tuning Rule

The control system is described in Figure 6.1, where u is the manipulated control input, d the disturbance, y the controlled output, y_s the setpoint (reference) for the controlled output, $c(s)$ the PI controller transfer function, and $g(s)$ the process transfer function. The PI controller takes the form of

$$c(s) = k_c \left(1 + \frac{1}{\tau_I s} \right), \quad (6.1)$$

where k_c and τ_I are the proportional (P) gain and the integral (I) time constant respectively.

Figure 6.1 Block diagram of feedback control system.

CSR Experiment. When a single P controller ($c(s) = k_{c0}$) is applied to the process, a setpoint change is made. From the CSR experiment (see Figure 6.2), we record the following values [11]:

Δy_s : Setpoint change

Δy_p : Peak output change

Δy_∞ : Steady-state output change after setpoint step test

t_r : Time from setpoint change to reach steady-state output for the first time

t_p : Time from setpoint change to reach peak output

k_{c0} : Controller gain used in experiment.

From this data, the following parameters are calculated

$$M_p = \frac{\Delta y_p - \Delta y_\infty}{\Delta y_\infty}, \quad b = \frac{\Delta y_\infty}{\Delta y_s}. \quad (6.2)$$

Figure 6.2 Setpoint response with P control [11].

As recommended in (Shamsuzzoha and Skogestad, 2010) [11], for deriving good PI settings the experiment should make M_p be larger than 10% and best around 30%. In case that it takes a long time for the response to settle down, one may simply record the output, Δy_u , when the response reaches its first minimum and compute Δy_∞ as $\Delta y_\infty = 0.45(\Delta y_p + \Delta y_u)$ [11].

The analytical derivation of the PI tuning rule proceeds as follows. Consider an IPTD process

$$g(s) = \frac{ke^{-\theta s}}{s}, \quad (6.3)$$

where k is the process gain and θ the time delay. With $c(s) = k_{c0}$ applied to the process, the closed-loop transfer function is obtained as

$$\tilde{g}(s) := \frac{g(s)c(s)}{1 + g(s)c(s)} = \frac{e^{-\theta s}}{s/K + e^{-\theta s}}, \quad (6.4)$$

where $K := kk_{c0}$. The time delay component in Eq. (6.4) is usually approximated by Padé approximation or Maclaurin expansion. Although Padé approximation is normally more

accurate, Maclaurin expansion is adopted for the purpose of deriving a simple analytical PI tuning rule. Use Maclaurin expansion and approximate the numerator and denominator of $\tilde{g}(s)$ by the second-order polynomials respectively, yielding

$$\tilde{g}(s) \approx \frac{s^2 - \frac{1}{0.5\theta}s + \frac{1}{0.5\theta^2}}{s^2 + \left(\frac{1}{0.5K\theta^2} - \frac{1}{0.5\theta}\right)s + \frac{1}{0.5\theta^2}}. \quad (6.5)$$

Hence the characteristic polynomial of $\tilde{g}(s)$ is

$$f(s) := s^2 + \left(\frac{1}{0.5K\theta^2} - \frac{1}{0.5\theta}\right)s + \frac{1}{0.5\theta^2}. \quad (6.6)$$

The above $f(s)$ is in the standard second-order form, $s^2 + 2\zeta\omega_n s + \omega_n^2$, with

$$\omega_n = \frac{\sqrt{2}}{\theta}, \quad \zeta = \frac{1}{\sqrt{2}} \left(\frac{1}{K\theta} - 1 \right). \quad (6.7)$$

In Eq. (6.7), ζ has a physical meaning of being the damping ratio of the closed-loop system [77]. Equation (6.7) solves K as

$$K = \frac{1}{(\sqrt{2}\zeta + 1)\theta}. \quad (6.8)$$

Therefore the unit step setpoint response is

$$Y(s) = \tilde{g}(s)y_s(s) \approx \frac{s^2 - \frac{1}{0.5\theta}s + \frac{1}{0.5\theta^2}}{s^2 + \left(\frac{1}{0.5K\theta^2} - \frac{1}{0.5\theta}\right)s + \frac{1}{0.5\theta^2}} \frac{1}{s} = \frac{a}{s} + \frac{b(s + \sigma) + c\omega_d}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \quad (6.9)$$

where $\sigma := \zeta\omega_n$ and $\omega_d := \omega_n\sqrt{1-\zeta^2}$, and the parameters ζ and ω_n are those defined in Eq. (6.7), and a , b and c are given by

$$a = 1, \quad b = 0, \quad c = -\frac{1}{0.5K\omega_d\theta^2} = -\frac{2\zeta + \sqrt{2}}{\sqrt{1-\zeta^2}}. \quad (6.10)$$

Assume that the initial states of the system and their derivatives are zero. By inverse Laplace transform, from Eq. (6.9) the time-domain response is derived as

$$y(t) \approx 1 + ce^{-\sigma t} \sin \omega_d t. \quad (6.11)$$

From Eq. (6.11), the time-domain performance indices such as the rise time t_r , the peak time t_p , and the overshoot M_p can all be estimated. Let the rise time be defined as the time for $y(t)$ reaching the steady-state value of one for the first time. This means

$$y(t_r) = 1 = 1 + ce^{-\sigma t_r} \sin \omega_d t_r. \quad (6.12)$$

Equation (6.12) solves t_r as

$$t_r = \frac{\pi}{\omega_d} = \frac{\pi\theta}{\sqrt{2(1-\zeta^2)}}. \quad (6.13)$$

With $dy(t)/dt|_{t=t_p} = 0$, the peak time t_p is solved as

$$t_p = \frac{1}{\omega_d} (\pi + \arccos \zeta) = \frac{\theta}{\sqrt{2(1-\zeta^2)}} (\pi + \arccos \zeta). \quad (6.14)$$

Consequently the overshoot M_p , which is defined in Eq. (6.2), is computed as

$$\begin{aligned} M_p &= (y(t_p) - 1) \times 100\% = ce^{-\sigma t_p} \sin \omega_d t_p \times 100\% \\ &= (2\zeta + \sqrt{2}) \exp \left(-\frac{\zeta}{\sqrt{1-\zeta^2}} (\pi + \arccos \zeta) \right) \times 100\%. \end{aligned} \quad (6.15)$$

Note that the dimensionless scalars t_r/θ , t_p/θ and M_p are all functions of ζ . The relationship between M_p and ζ are shown in Figure 6.3. Hence ζ can be read from the M_p - ζ curve once M_p is measured from the CSR experiment. Or it can be solved from Eqs. (6.13) and (6.14) as

$$\zeta = \left| \cos \frac{\pi t_p}{t_r} \right|. \quad (6.16)$$

Figure 6.3 M_p - ζ curve: relation between the overshoot M_p and the damping ratio ζ .

Although $\cos \frac{\pi t_p}{t_r} < 0$ comes from $\pi \leq \frac{\pi t_p}{t_r} \leq \frac{3\pi}{2}$ as deduced from Eqs. (6.13) and (6.14), the absolute is taken to ensure a positive ζ since t_p and t_r are measured values and the analysis here is approximate. Consequently θ can be estimated from Eq. (6.14) as

$$\theta = \frac{t_p \sqrt{2(1-\zeta^2)}}{\pi + \arccos \zeta}. \quad (6.17)$$

(The parameter θ may also be estimated from Eq. (6.13) in terms of t_r . The estimation in Eq. (6.17) is recommended since it avoids the measurement of t_r if the damping ratio is read from the M_p - ζ curve.) The process gain is therefore solved from Eq. (6.8) as

$$k = \frac{1}{(\sqrt{2}\zeta + 1)\theta k_{c0}}. \quad (6.18)$$

Note that the SIMC tuning rule [6] for an IPTD process is equivalently expressed by

$$k_c = \frac{\alpha}{k\theta}, \quad \tau_I = \frac{4\theta}{\alpha}, \quad (6.19)$$

where α is a tuning parameter that corresponds to $\theta/(\tau_c + \theta)$ in the original SIMC rule. Applying the estimated processes parameters θ and k respectively in Eqs. (6.17) and (6.18), the SIMC tuning rule for an IPTD process becomes

$$\begin{aligned} k_c &= \alpha(\sqrt{2}\zeta + 1)k_{c0}, \\ \tau_I &= \frac{4}{\alpha} \frac{\sqrt{2(1-\zeta^2)}}{\pi + \arccos \zeta} t_p, \\ \text{where } \zeta &= \left| \cos \frac{\pi t_p}{t_r} \right|. \end{aligned} \quad (6.20)$$

Alternatively, ζ can be read from the M_p - ζ curve shown in Figure 6.3. Equation (6.20) is the new PI tuning rule which requires no modeling of the process dynamics but only the peak time and rise time or overshoot as recorded from a single CSR experiment. This eases the PI controller tuning in practice.

Next, consider an FOPTD process

$$g(s) = \frac{ke^{-\theta s}}{\tau s + 1}, \quad (6.21)$$

where k is the process gain, θ the time delay and τ the process time constant. During the transient of a setpoint response, since it involves mainly high frequency response, the transfer function can be approximated as

$$g(s) \approx \frac{k'e^{-\theta s}}{s}, \quad k' := \frac{k}{\tau}. \quad (6.22)$$

As the transient dynamics is of main interest, where quantities such as the rise time, peak time and overshoot are measured, approximate analysis can be made similarly to that for an IPTD process. Therefore the time delay θ is estimated in Eq. (6.17) and the gain k' is estimated in Eq. (6.8) with $K := k'k_{c0}$. At the steady state, the process gain k satisfies

$$kk_{c0} = \frac{b}{1-b}, \quad (6.23)$$

where b is given in Eq. (6.2). Equation (6.23), together with Eq. (6.8), solves

$$\tau = \frac{kk_{c0}}{K} = \frac{b}{1-b}(\sqrt{2}\zeta + 1)\theta. \quad (6.24)$$

The SIMC tuning rule [6] for an FOPTD process is equivalently expressed as

$$k_c = \frac{\alpha}{k\theta}, \quad \tau_I = \min\left\{\tau, \frac{4\theta}{\alpha}\right\}, \quad (6.25)$$

where α is a tuning parameter. Applying the process parameter estimated in Eqs. (6.8), (6.17) and (6.24), the SIMC tuning rule for an FOPTD process is rewritten as

$$\begin{aligned} k_c &= \alpha(\sqrt{2}\zeta + 1)k_{c0}, \\ \tau_I &= \min\left\{\left|\frac{b}{1-b}\right|(\sqrt{2}\zeta + 1), \frac{4}{\alpha}\right\} \times \frac{\sqrt{2(1-\zeta^2)}}{\pi + \arccos\zeta} t_p, \\ \text{where } \zeta &= \left|\cos\frac{\pi t_p}{t_r}\right|. \end{aligned} \quad (6.26)$$

The damping ratio ζ can alternatively be read from the M_p - ζ curve as shown in Figure 6.3. In Eq. (6.26), the absolute are taken to ensure positive values in the presence of measurement and approximation errors. The tuning rule (6.26) covers the tuning rule (6.20), since $b/(1-b) \rightarrow \infty$ for an IPTD process. Thus for either an IPTD or an FOPTD process, the PI tuning rule is given in Eq. (6.26).

The single tuning parameter α controls the trade-off between closed-loop performance and robustness. Hence an appropriate choice of α is important. Though it is difficult to derive any analytical guideline for determining α properly, it is clear that a larger α leads to more aggressive closed-loop response yet less robustness and vice versa. In applications, α can be detuned from a small (conservative) value until satisfactory performance is achieved. Extensive simulations indicate that it is almost sufficient to start α as 0.4 (which is conservative in most cases) and tune it at a step of 0.05 or 0.1 up if the response is too sluggish and down otherwise. The simulations also indicate that an acceptable α normally falls into the range of [0.2, 0.6].

Remark 6.1 If the time delay (θ) is estimated from Eq. (6.13) in terms of t_r , instead of t_p , then the PI tuning rule can be derived as

$$\begin{aligned} k_c &= \alpha(\sqrt{2}\zeta + 1)k_{c0}, \\ \tau_I &= \min \left\{ \left| \frac{b}{1-b} \right| (\sqrt{2}\zeta + 1), \frac{4}{\alpha} \right\} \times \frac{\sqrt{2(1-\zeta^2)}}{\pi} t_r, \\ \text{where } \zeta &= \left| \cos \frac{\pi t_p}{t_r} \right|. \end{aligned} \quad (6.27)$$

The parameters have the same meanings as those in (6.26). The main shortcoming of using (6.27) is that measuring t_r becomes necessary. The rule (6.26), however, does not require t_r if ζ is read from the M_p - ζ curve as shown in Figure 6.3.

In comparison, using the CSR method, Shamsuzzoha and Skogestad concluded a similar PI tuning rule from series of numerical experiments that aims to match the SIMC rule.[11] The rule takes the form of

$$\begin{aligned} k_c &= 2\alpha A k_{c0}, \\ \tau_I &= \min \left\{ 0.86A \left| \frac{b}{1-b} \right| t_p, \frac{1.22}{\alpha} t_p \right\}, \\ \text{where } A &= 1.152M_p^2 - 1.607M_p + 1.0, \end{aligned} \quad (6.28)$$

where α is a tuning parameter similar to the one in Eq. (6.26), which corresponds to $1/(2F)$ as adopted in the original tuning rule[11]. Comparing it with the new rule in Eq. (6.26), we see that these two rules are similar in form: in the proportional gains, the coefficient $2A$ in Eq. (6.28) is a function only of M_p and so is the coefficient $\sqrt{2}\zeta + 1$ in Eq. (6.26) (as refers to the M_p - ζ curve in Figure 6.3 or Eq. (6.15)); and the integral gains are both functions of t_p . Nevertheless, the new rule does not give the same relation between $\sqrt{2}\zeta + 1$ and M_p as its counterpart $2A$ and M_p . Another difference is that the rule (6.28) adopts approximate relations of $\theta = 0.43t_p$ and $\theta = 0.305t_p$ (when M_p

varying from 0.1 to 0.6) in the first and second components of the $\min\{\bullet, \bullet\}$ function respectively, whereas the new rule (6.26) uses a common estimate of θ for both components as given in Eq. (6.17). Indeed similar approximate relations between θ and t_p may be established using the $M_p - \zeta$ curve shown in Figure 6.3, subject to $\alpha = 0.5$.

Choice for the P Controller Gain k_{c0} . An overshoot of around 30% is recommended for the CSR experiment giving good PI settings.[11] This is confirmed by the simulations with the proposed PI tuning rule (The simulation results are not shown for brevity.). Normally such an overshoot is achieved by detuning the P controller gain k_{c0} via trials and errors. The detuning process can be time consuming and may disturb the process much as are undesirable in applications. Therefore an efficient way for determining k_{c0} is important for the CSR experiment. We present a method to generate k_{c0} 's that can reduce the number of times of detuning k_{c0} . The method is developed based on the PI tuning rule proposed by Shamsuzzoha and Skogestad, avoiding the errors involved in the above analysis.

The method requires a foregoing CSR experiment. Suppose that we apply a P controller gain of k_{c0}^0 in a CSR experiment and it results in an overshoot M_p^0 that is larger than 10% but not around 30%. Let the target overshoot be M_p^* and the target P controller gain be k_{c0} . Note that Shamsuzzoha-Skogestad's PI tuning rule aims to match the SIMC rule which keeps a constant P gain k_c regardless of the overshoot resulted from the CSR experiment. Ideally, k_c should be the same as determined with different overshoots from various CSR experiments. That is, it should have

$$\begin{aligned} & 2\alpha(1.152(M_p^0)^2 - 1.607M_p^0 + 1.0)k_{c0}^0 \\ & = 2\alpha(1.152(M_p^*)^2 - 1.607M_p^* + 1.0)k_{c0}, \end{aligned} \quad (6.29)$$

which solves

$$k_{c0} = \frac{1.152(M_p^0)^2 - 1.607M_p^0 + 1.0}{1.152(M_p^*)^2 - 1.607M_p^* + 1.0} k_{c0}^0. \quad (6.30)$$

Equation (6.30) gives a general guideline for choosing the P controller gain for the next CSR experiment. If M_p^* is set as 30%, then the gain for the next CSR experiment is recommended as

$$k_{c0} = 1.609 \times [1.152(M_p^0)^2 - 1.607M_p^0 + 1.0] k_{c0}^0. \quad (6.31)$$

If the gain does not result in a desired overshoot, the formula (6.31) can be applied repeatedly until the overshoot reaches around 30%. Such a repeating process converges and ultimately gives a P controller gain that results in the exact overshoot of 30%. With the monotonic relationship between M_p and k_{c0} , this can be understood from Eq. (6.30): The gain k_{c0} will be adjusted until $M_p^0 \rightarrow M_p^*$ and thus $k_{c0} \rightarrow k_{c0}^0$. The observation has been justified by extensive simulations and typical results are shown in the next section.

6.3 Simulation Results

Although the PI tuning rule in Eq. (6.26) was derived for IPTD and FOPTD processes, it turns out to be effective for a wide range of processes. Simulations were carried out on various processes and typical results are summarized in Table 6.1. In the table, the PI settings and peak sensitivities in regular and italic fonts were obtained by the proposed method with ζ 's as computed by the formula and read from the M_p - ζ curve, respectively; and the values of F (as adopted in the Shams-Skog's rule) are equal to $1/(2\alpha)$. (For all the processes being studied, we adopt the same numbering as that in (Shamsuzzoha and Skogestad, 2010) [11] to achieve good consistency and easy reference.) In the simulations, the damping ratios ζ 's were read from the M_p - ζ curve or

computed using the rise time t_r 's and the peak time t_p 's. Typical simulation results are shown in Figure 6.4, where in each case a unit step change was applied in both the setpoint and the disturbance.

The results indicate that the proposed PI tuning rule leads to similar closed-loop performance and robustness (in terms of peak sensitivity) in each case as compared to the Shams-Skog's rule, if a proper α is chosen. It is observed that for each process, the PI settings work well in both situations when the damping ratios ζ 's are computed by the formula and read from the $M_p - \zeta$ curve. Overall when ζ 's were read from the $M_p - \zeta$ curve, the PI settings are more aggressive, giving rise to faster setpoint responses with larger overshoots and faster load responses yet less deviations. This is also reflected from the larger peak sensitivities given in Table 6.1.

The PI tuning rule in (6.27) is also tested. In the tests, the pairs of P controller gain (k_{c0}) and tuning factor (α) for $E\{6, 8, 17, 21, 24, 33\}$ are $\{0.8, 0.5\}$, $\{0.58, 0.5\}$, $\{4.0, 0.4\}$, $\{0.3, 0.25\}$, $\{0.8, 0.5\}$ and $\{4.0, 0.6\}$, respectively. And the same PI settings with Shams-Skog's rule as in Table 6.1 were applied. The simulation results are shown in Figure 6.5, from which we observe that both the setpoint and disturbance responses are similar to those shown in Figure 6.4 as obtained with the rule in (6.26). We conclude that the rule in (6.27) can work as well as the rule in (6.26) if the rise time is measured from a CSR experiment.

The closed-loop response normally changes smoothly as α changes. The simulations indicate that it is good to start α as 0.4 and then adjust it, say, at a step of 0.05 or 0.1, until a satisfactory response is attained. Typical closed-loop responses for the proposed PI tuning rule when different α 's were applied are shown in Figure 6.6. The results confirm that a larger α leads to more aggressive response with less robustness and vice versa. In comparison, Shams-Skog's rule has an advantage that a constant value of α at 0.5 is

almost sufficient to give satisfactory closed-loop performance for various processes, which can be seen from the PI settings in Table 6.1 and the closed-loop responses shown in Figure 6.6.

Table 6.1 PI settings for Shams-Skog's (short for Shamsuzzoha-Skogestad's) and proposed rules.

Case	Process model	k_{c0}	M_p	t_r	t_p	b	Method	α	F	k_c	τ_I	M_s
E1	$\frac{1}{(s+1)(0.2s+1)}$	15.0	0.327	0.227	0.373	0.938	Shams-Skog	0.5	1.0	8.968	0.910	1.75
							Proposed	0.4	1.25	9.687	1.115	1.74
										9.467	1.122	1.73
E6	$\frac{(0.17s+1)^2}{s(s+1)^2(0.028s+1)}$	0.80	0.301	3.187	5.002	1.000	Shams-Skog	0.5	1.0	0.496	12.205	1.77
							Proposed	0.5	1.0	0.522	12.293	1.80
										0.642	11.979	1.99
E8	$\frac{1}{s(s+1)^2}$	0.58	0.307	4.038	6.210	1.000	Shams-Skog	0.5	1.0	0.357	15.152	1.75
							Proposed	0.5	1.0	0.338	15.187	1.71
										0.463	14.890	2.01
E11	$\frac{(-s+1)e^{-s}}{(6s+1)(2s+1)^2}$	1.40	0.344	9.391	13.674	0.583	Shams-Skog	0.5	1.0	0.817	9.609	1.59
							Proposed	0.35	1.43	0.585	7.004	1.49
										0.765	9.012	1.56
E12	$\frac{(6s+1)(3s+1)e^{-0.3s}}{(10s+1)(8s+1)(s+1)}$	15.0	0.310	0.609	0.844	0.938	Shams-Skog	0.5	1.0	9.191	2.059	1.74
							Proposed	0.45	1.11	10.125	2.281	1.86
										10.769	2.250	1.95
E13	$\frac{(2s+1)e^{-s}}{(10s+1)(0.5s+1)}$	4.75	0.302	1.687	2.183	0.826	Shams-Skog	0.5	1.0	2.942	5.327	1.76
							Proposed	0.35	1.43	3.082	5.332	1.82
										2.665	4.978	1.64
E17	$\frac{e^{-s}}{5s+1}$	4.00	0.298	2.123	3.024	0.800	Shams-Skog	0.5	1.0	2.493	6.484	1.56
							Proposed	0.4	1.25	2.133	4.954	1.48
										2.573	5.819	1.59
E21	$e^{-s} *$	0.30	0.300	1.001	2.001	0.231	Shams-Skog	0.5	1.0	0.186	0.321	1.53
							Proposed	0.4	1.25	-	-	-
										0.193	0.288	1.66
E24	$\frac{e^{-s}}{s}$	0.80	0.302	2.282	3.282	1.000	Shams-Skog	0.5	1.0	0.496	8.008	1.70
							Proposed	0.5	1.0	0.509	8.064	1.72
										0.642	7.861	2.03
E29	$\frac{(-s+1)e^{-2s}}{(s+1)^5}$	0.4	0.304	8.812	11.981	0.286	Shams-Skog	0.5	1.0	0.247	2.546	1.70
							Proposed	0.35	1.43	0.225	2.301	1.72
										0.224	2.299	1.72
E32	$\frac{(s^2+2s+9)(-2s+1)(s+1)e^{-2s}}{(s^2+0.5s+1)(5s+1)^2}$	0.12	0.301	10.623	15.055	0.519	Shams-Skog	0.5	1.0	0.074	8.674	1.55
							Proposed	0.4	1.25	0.065	6.806	1.61
										0.077	7.809	1.64
E33	$\frac{e^{-s}}{5s-1}$	4.00	0.300	2.527	3.677	1.333	Shams-Skog	0.5	1.0	2.487	7.866	2.33
							Proposed	0.6	0.83	2.878	5.402	2.92
										3.855	7.070	3.24

* For a pure time delay process, the analytical ζ is zero and hence invalid. For this case, ζ has to be read from the

M_p - ζ curve.

Figure 6.4 Output responses for PI control of typical processes: solid black line—Shams-Skog's rule, dotted red line—proposed rule (6.26) with ζ being computed by the formula, dashdot green line—proposed rule (6.26) with ζ being read from the M_p - ζ curve. The x-axes are times and the y-axes are output responses.

Figure 6.5 Output responses for PI control of typical processes: solid black line—Shams-Skog's rule, dotted red line—proposed rule (6.27) with ζ being computed by the formula, dashdot green line—proposed rule (6.27) with ζ being read from the M_p - ζ curve. The x-axes are times and the y-axes are output responses.

Finally, four examples are presented to validate the method proposed for choosing P controller gain for the CSR experiment. The target overshoot is set as $M_p^* = 30\%$. Hence the P controller gain k_{c0} is recommended as that in the formula (6.31). The formula was applied repeatedly to update k_{c0} until the overshoot of the CSR converges to 30%. The four examples are with the processes E1, E17, E21 and E24 as given in Table 6.1. As reported in (Shamsuzzoha and Skogestad, 2010) [11], for processes E17 and E24, the P

gains of the PI settings are almost the same in spite of the CSR's having various overshoots; whereas for processes E1 and E21, the P gains vary significantly when CSR having different overshoots. Note that the former and latter cases correspond to the cases that are consistent and inconsistent with the assumption of the analysis that led to the proposed formula (6.31). With a target overshoot of 30%, the CSR experiments were carried out by applying formula (6.31) repeatedly and the results are shown in Figure 6.7, where the arrows indicate the detuning directions of k_{c0} 's relative to their initial values. From the results, we see that in the cases of E17 and E24, both P controller gains converge quickly to the ideal ones giving target overshoots of 30%. In either case, it requires only one round of detuning k_{c0} before reaching an overshoot within 25%-35%. In contrast, in the cases of E1 and E21, both P controller gains converge much more slowly but to the ideal values ultimately. It takes four and six rounds of detuning k_{c0} before reaching an overshoot within 25%-35% for E1 and E21, respectively. Nevertheless, the number of rounds of detuning k_{c0} remains acceptably small. These results demonstrate the effectiveness and usefulness of the proposed method in determining a proper P controller gain for the CSR experiment.

Figure 6.6 Effect of detuning α : output responses for PI control of $g(s) = 1/[(s+1)(0.2s+1)]$, with unit-step setpoint change at $t = 0$ and unit-step load disturbance at $t = 5$.

Figure 6.7 Detuning process of the P controller gain k_{c0} using the proposed method.

6.4 Conclusions

An analytical PI tuning rule was derived for IPTD and FOPTD processes using the CSR method. The rule expresses the PI parameters in terms of the steady-state offset, peak time, and overshoot or rise time as recorded in a CSR experiment. The rule turns out to be applicable to a broad range of processes typical for process control, and it gives comparable performance to the PI tuning rule proposed in (Shamsuzzoha and Skogestad, 2010) [11] when a tuning parameter is properly chosen. Meanwhile, a method was proposed for choosing the P controller gain for the CSR experiment to result in a preferred overshoot of around 30%. The presented analysis and derived rule provide some insight and support to the PI tuning rule proposed by Shamsuzzoha and Skogestad.

Chapter 7

Further Results on the Local Solutions to SOC

This chapter revisits the local solutions for SOC to minimize worst-case loss and average loss and derives more complete characterizations for each of them. Specifically, a more general form of the solution for SOC minimizing worst-case loss is found and the available solution for SOC minimizing average loss is proved to be complete. The results contribute to a better understanding of these two classes of solutions and their relations.

7.1 Introduction

Various methods have been proposed for SOC which selects CVs by minimizing steady-state economic loss in the presence of disturbances and implementation errors. The methods include the qualitative rules [13], minimum singular value rule [40-41], null space method [46], exact local method [34, 40, 43-44], gradient function [18], etc. Among them, the exact local method gives a general local solution to the SOC problem. This chapter reports some further results with this method.

Let the CVs be expressed as linear combinations of available measurements. Exact local method formulates the SOC problem as solving for the optimal MCM, denoted by H , leading to minimal local worst-case loss [40, 44] or average loss [43]. Originally the solutions were found to minimize worst-case loss by solving a nonlinear optimization problem [40]. To improve the efficiency and guarantee global optimality, solutions

involving slight computations were proposed in [44]. Later, solutions were proposed for minimizing average loss, which minimize worst-case loss simultaneously [34, 43]. This argues for the favor of minimizing average instead of worst-case loss for SOC. Indeed case studies indicated that there are H 's minimizing worst-case but not average loss, although simultaneous minimizations happen sometimes [43]. These observations imply that the form of H 's presented in [43] for minimizing worst-case loss is not complete. In the meanwhile, it is unclear whether the available solution to SOC minimizing average loss is complete or not.

This chapter extends the aforementioned results, establishing more complete characterizations of the solutions of H 's for SOC to minimize worst-case and average losses, respectively. The renewed characterizations extend the solution of H minimizing worst-case loss and give further insight into the solution of H minimizing average loss. The new results also contribute to revealing a clear relation between the solutions of the two kinds of SOC problems.

The rest of the chapter is organized as follows. Section 7.2 summarizes the solutions of SOC minimizing worst-case and average losses, respectively. Section 7.3 presents the new results we obtain. Finally Section 7.4 concludes the chapter.

7.2 Local SOC

Let $G_y \in \mathbb{R}^{n_y \times n_u}$, $J_{uu} \in \mathbb{R}^{n_u \times n_u}$, $J_{ud} \in \mathbb{R}^{n_u \times n_d}$, $G_y^d \in \mathbb{R}^{n_y \times n_d}$, $W_d \in \mathbb{R}^{n_d \times n_d}$ and $W_n \in \mathbb{R}^{n_y \times n_y}$ be given matrices about the process, of which the details are referred to [34, 40, 43]. Define several key matrices as

$$M_H = J_{uu}^{0.5} (HG_y)^{-1} HY, \quad (7.1)$$

$$A_\gamma = \gamma^2 G_y J_{uu}^{-1} G_y^T - YY^T, \quad (7.2)$$

$$A_X = G_y J_{uu}^{-0.5} X J_{uu}^{-0.5} G_y^T - Y Y^T, \quad (7.3)$$

$$Z = J_{uu}^{-0.5} G_y^T (Y Y^T)^{-1} G_y J_{uu}^{-0.5}, \quad (7.4)$$

where

$$Y = [(G_y J_{uu}^{-1} J_{ud} - G_y^d) W_d \quad W_n], \quad (7.5)$$

which is assumed to have full rank.

The SOC problem for minimizing worst-case loss can be formulated as an optimization problem [43-44]:

$$\begin{aligned} \min_{H, \gamma} \quad & \gamma \\ \text{s.t.,} \quad & H A_\gamma H^T \geq 0, \\ & \gamma \geq 0, \\ & \text{rank}(H G_y) = n_u. \end{aligned} \quad (7.6)$$

Similarly the SOC problem for minimizing average loss can be formulated as [43]:

$$\begin{aligned} \min_{H, X} \quad & \text{tr}(X) \\ \text{s.t.,} \quad & H A_X H^T \geq 0, \\ & X \geq 0, \\ & \text{rank}(H G_y) = n_u. \end{aligned} \quad (7.7)$$

For brevity, the coefficients in the objective functions are omitted. Note that the rank conditions, $\text{rank}(H) = n_u$, given in [43-44] are inaccurate.

Instead of solving the above optimization problems directly, explicitly expressed optimal solutions were derived as [43]:

$$\text{Minimizing worst-case loss: } \begin{cases} \gamma = \gamma^* := \sqrt{\bar{\lambda}_{Z^{-1}}} \\ H = H^* := C V_{A_{\gamma^*}, n_u}^T \end{cases} \quad (7.8)$$

$$\text{Minimizing average loss: } \begin{cases} X = X^* := Z^{-1}, \\ H = H^* := C V_{A_X^*, n_u}^T \end{cases} \quad (7.9)$$

where $\bar{\lambda}_{Z^{-1}}$ is the largest eigenvalue of Z^{-1} , columns of $V_{A_{\gamma^*}, n_u}$ (or $V_{A_{X^*}, n_u}$) are the (right) mutually orthogonal eigenvectors associated with the first n_u largest eigenvalues of A_{γ^*} (or A_{X^*}), and C is any nonsingular matrix. The rank conditions of $\text{rank}(HG_y) = n_u$ were implicitly assumed to be satisfied for the solutions in (7.8) and (7.9). In practice, this is almost always true if $\text{rank}(H) = n_u$ when the solutions are derived numerically. Hereafter we keep this implicit assumption.

The following knowledge is useful for the proofs of later results.

Definition 7.1 [93] *Let A be a square matrix and let C be nonsingular and of the same order as A . Then $C^T A C$ is called a congruence transformation of A .*

Lemma 7.1 [93] *Assume that A is symmetric, and let C be nonsingular. Then $C^T A C$ has the same number of positive eigenvalues, the same number of negative eigenvalues, and the same number of zero eigenvalues as A .*

Normally congruence transformation does not preserve the eigenvalues [93]. This implies mistakes of related statements in [43] although they do not affect the conclusions.

Lemma 7.2 [43] *For $A \in \Re^{m \times n}$, $m \leq n$, the largest m eigenvalues of $AA^T - I_m$ and $A^T A - I_n$ are the same.*

(There are a few mistakes in the statements of the original proof of Lemma 7.2, but they do not affect the proof much and the lemma keeps true.)

7.3 Main Results

Let the columns of $V_{A_{\gamma}}$ and V_{A_X} be mutually orthogonal eigenvectors of A_{γ} and A_X , respectively. The main results are summarized in lemmas and theorems.

Lemma 7.3 *Let $\gamma^* = \sqrt{\bar{\lambda}_{Z^{-1}}}$. If A_{γ^*} has m nonnegative eigenvalues, then $m \geq n_u$ and A_{γ^*} has a zero eigenvalue with equal algebraic and geometric multiplicities of $m - n_u$.*

Proof. It was proved in [43] that $\gamma^* = \sqrt{\bar{\lambda}_{Z^{-1}}}$ entails the n_u -th largest eigenvalue of A_{γ^*} be zero. This implies that A_{γ^*} has at least n_u nonnegative eigenvalues and hence $m \geq n_u$. It also implies that the $(n_u + 1)$ -th to m -th largest nonnegative eigenvalues must be zero if there were any. Hence the algebraic multiplicity of the zero eigenvalue is $m - n_u$. Since A_{γ^*} is symmetric which is diagnosable, it is necessary that the geometric multiplicity of the zero eigenvalue equals its algebraic multiplicity and hence $m - n_u$. $\square$

Theorem 7.1 *H solves the problem (7.6) if $H = H^* := CV_{A_{\gamma^*,m}}^T$, where $\gamma^* = \sqrt{\bar{\lambda}_{Z^{-1}}}$, and columns of $V_{A_{\gamma^*,m}}$ are m mutually orthogonal eigenvectors associated with the total m ($\geq n_u$) nonnegative eigenvalues of A_{γ^*} , and C is an $n_u \times m$ matrix with full row rank.*

Proof. The proof is similar to that for validating the solution in (7.8) and the detail is referred to [43]. The only difference is that the rows of H are now combinations of eigenvectors for all the m nonnegative eigenvalues, instead of the first n_u largest nonnegative eigenvalues of A_{γ^*} . $\square$

Theorem 7.1 extends the solution of problem (7.6) as given in (7.8). However, it remains to provide a sufficient but not necessary solution. Note that solutions of H are all those satisfying $HA_{\gamma^*}H^T \geq 0$. We have

$$\begin{aligned}
HA_{\gamma}^*H^T \geq 0 &\Rightarrow HV_{A_{\gamma}^*} \begin{bmatrix} \Lambda_{+,0} & \\ & \Lambda_- \end{bmatrix} V_{A_{\gamma}^*}^T H^T \geq 0 \\
&\Rightarrow H \begin{bmatrix} V_{A_{\gamma}^*,m} & V_{A_{\gamma}^*,(n_y-m)} \end{bmatrix} \begin{bmatrix} \Lambda_{+,0} & \\ & \Lambda_- \end{bmatrix} \begin{bmatrix} V_{A_{\gamma}^*,m}^T \\ V_{A_{\gamma}^*,(n_y-m)}^T \end{bmatrix} H^T \geq 0 \\
&\Rightarrow HV_{A_{\gamma}^*,m} \Lambda_{+,0} V_{A_{\gamma}^*,m}^T H^T + HV_{A_{\gamma}^*,(n_y-m)} \Lambda_- V_{A_{\gamma}^*,(n_y-m)}^T H^T \geq 0,
\end{aligned} \tag{7.10}$$

where the diagonals of $\Lambda_{+,0}$ and Λ_- consist of the m nonnegative and $n_y - n_u$ negative eigenvalues of A_{γ}^* respectively, columns of $V_{A_{\gamma}^*,m} \in \Re^{n_y \times m}$ and $V_{A_{\gamma}^*,(n_y-m)} \in \Re^{n_y \times (n_y-m)}$ are mutually orthogonal eigenvectors associated with the eigenvalues of A_{γ}^* . For the inequality in (7.10) to be true, it is not necessary to require $HV_{A_{\gamma}^*,(n_y-m)} \Lambda_- V_{A_{\gamma}^*,(n_y-m)}^T H^T = 0$. This implies that the solution given in Theorem 7.1 is sufficient but not necessary.

To illustrate, let us see an example. Let $n_u = 2$, $n_y = 3$ and $A_{\gamma}^* = \text{diag}\{1, 0, -1\}$ (a diagonal matrix whose elements lie on the diagonal in order). Hence $V_{A_{\gamma}^*,2} = [1 \ 0; 0 \ 1; 0 \ 0]$, where the element pairs denote the rows in order. Thus $H = [h_{ij}]_{2 \times 3}$, for $\forall h_{12}, h_{22}$ and $\forall h_{13}/h_{11} = h_{23}/h_{21} =: \alpha$ satisfying $|\alpha| \leq 1$ and $\text{rank}(H) = 2$, is a solution to the problem (7.6), which satisfies $HA_{\gamma}^*H^T \geq 0$. If $h_{13}, h_{23} \neq 0$, then H is a solution but not expressible by $CV_{A_{\gamma}^*,2}^T$ for any nonsingular $C \in \Re^{2 \times 2}$.

A similar but stronger conclusion holds for the solutions of H for problem (7.7). Before presenting the conclusion, we give another lemma.

Lemma 7.4 *Let $X^* = Z^{-1}$. A_{X^*} has $n_y - n_u$ negative eigenvalues and a zero eigenvalue with equal algebraic and geometric multiplicities of n_u .*

Proof. Let R be an upper triangular matrix satisfying $YY^T = R^T R$ (Cholesky factorization) and let $Q = R^{-T} G_y J_{uu}^{-0.5}$, giving $Q^T Q = Z$. By congruence transformation it follows that the matrices A_{X^*} and $QX^*Q^T - I_{n_y}$ have eigenvalues with the same signs (refer to Lemma 7.1). The solution $X^* = Z^{-1}$ to problem (7.7) implies $X^{*0.5} Z X^{*0.5} - I_{n_u} = 0$, entailing that all the eigenvalues of $X^{*0.5} Z X^{*0.5} - I_{n_u}$ are zero. Since the first n_u largest eigenvalues of $QX^*Q^T - I_{n_y}$ are the same as the n_u eigenvalues of $X^{*0.5} Z X^{*0.5} - I_{n_u}$, it follows that the first n_u largest eigenvalues of $QX^*Q^T - I_{n_y}$ are all zero. Consequently the first n_u largest eigenvalues of A_{X^*} are zero. Note that $Q \in \mathbb{R}^{n_y \times n_u}$. By singular value decomposition it is easy to see that the rest $n_y - n_u$ eigenvalues of $QX^*Q^T - I_{n_y}$ are negative. Thus A_{X^*} also has $n_y - n_u$ negative eigenvalues.

Since A_{X^*} is symmetric, it follows that the geometric multiplicity of the zero eigenvalue equals its algebraic multiplicity and hence n_u . $\square$

Theorem 7.2 H solves problem (7.7) if and only if $H = H^* := C V_{A_{X^*}, n_u}^T$, where $X^* := Z^{-1}$, and columns of $V_{A_{X^*}, n_u}$ are n_u mutually orthogonal eigenvectors associated with the zero eigenvalue of A_{X^*} , and C is a nonsingular $n_u \times n_u$ matrix.

Proof. The sufficiency is easily proved by combining Lemma 7.4 with the result in (7.9). We prove the necessity. Lemma 7.4 indicates that A_{X^*} has n_u zero eigenvalues and $n_y - n_u$ negative eigenvalues. Since H solves problem (7.7), we have

$$\begin{aligned}
HA_{X^*}H^T \geq 0 &\Rightarrow HV_{A_{X^*}} \begin{bmatrix} 0_{n_u} & \\ & \Lambda_- \end{bmatrix} V_{A_{X^*}}^T H^T \geq 0 \\
&\Rightarrow H \begin{bmatrix} V_{A_{X^*}, n_u} & V_{A_{X^*}, (n_y - n_u)} \end{bmatrix} \begin{bmatrix} 0_{n_u} & \\ & \Lambda_- \end{bmatrix} \begin{bmatrix} V_{A_{X^*}, n_u}^T \\ V_{A_{X^*}, (n_y - n_u)}^T \end{bmatrix} H^T \geq 0 \\
&\Rightarrow HV_{A_{X^*}, (n_y - n_u)} \Lambda_- V_{A_{X^*}, (n_y - n_u)}^T H^T \geq 0 \Rightarrow HV_{A_{X^*}, (n_y - n_u)} = 0 \\
&\Rightarrow h_i^T \in \text{Null}(V_{A_{X^*}, (n_y - n_u)}^T), \quad i = 1, 2, \dots, n_u, \quad H = [h_1^T \ h_2^T \ \dots \ h_{n_u}^T]^T, \\
&\Rightarrow H = CV_{A_{X^*}, n_u}^T.
\end{aligned} \tag{7.11}$$

In the above, the diagonal of Λ_- consists of the $n_y - n_u$ negative eigenvalues of A_{X^*} , columns of $V_{A_{X^*}, n_u} \in \mathfrak{R}^{n_y \times n_u}$ and $V_{A_{X^*}, (n_y - n_u)} \in \mathfrak{R}^{n_y \times (n_y - n_u)}$ are mutually orthogonal eigenvectors associated with the eigenvalues of A_{X^*} , and $C \in \mathfrak{R}^{n_u \times n_u}$ is nonsingular.

This establishes the necessity. $\square$

Based on Lemma 7.4 and Theorem 7.2, we have the following result.

Corollary 7.1 *H solves problem (7.7) if and only if*

$$H = H^* := CG_y^T(YY^T)^{-1}, \tag{7.12}$$

for a nonsingular $C \in \mathfrak{R}^{n_u \times n_u}$.

Proof. Note that $(YY^T)^{-1}G_y$ has the same rank as G_y and thus has full rank. With Theorem 7.2, it is sufficient to prove that the columns of $(YY^T)^{-1}G_y$ are eigenvectors associated with the zero eigenvalue of A_{X^*} , i.e., $A_{X^*}(YY^T)^{-1}G_y = 0$, which can be established as follows:

$$\begin{aligned}
A_{X^*}(YY^T)^{-1}G_y &= (G_y J_{uu}^{-0.5} X^* J_{uu}^{-0.5} G_y^T - YY^T)(YY^T)^{-1}G_y \\
&= \left(G_y \left(G_y^T (YY^T)^{-1} G_y \right)^{-1} G_y^T - YY^T \right) (YY^T)^{-1} G_y \\
&= 0.
\end{aligned} \tag{7.13}$$

This completes the proof. $\square$

Note that the explicit solution of H reported in [34] is equivalent to the one in (7.12), by taking $C = C' J_{uu}^{0.5} (G_y^T (Y Y^T)^{-1} G_y)^{-1}$ with a nonsingular $C' \in \Re^{n_u \times n_u}$. Given the solution in (7.12), the minimal objective function is obtained as $\text{tr}(X)$ which involves only matrix multiplications and additions. This property might be used to improve the BAB algorithm proposed in [49] for CV selection, avoiding solving for eigenvalues of matrices as required in the original algorithm.

Based on Corollary 7.1, a relation can be established between the solutions of SOC for minimizing worst-case loss and average loss, respectively.

Corollary 7.2 *A solution, denoted by H^* , of problem (7.6) is a solution of problem (7.7) if and only if there exists a nonsingular matrix C such that $H^* = C G_y^T (Y Y^T)^{-1}$.*

Proof. It follows directly from Corollary 7.1. $\square$

Corollary 7.2, together with the fact that ‘A solution of problem (7.7) is also a solution of problem (7.6)’ [43], gives a clear characterization of the relation between the solutions of the two kinds of SOC problems.

7.4 Conclusions

More complete characterizations of the local solutions for SOC to minimize worst-case and average losses were obtained. The solution for minimizing worst-case loss extends the previous one by allowing for combinations of eigenvectors associated with the additional zero eigenvalues (if any), beyond the first largest n_u nonnegative eigenvalues, of the key matrix A_{γ^*} . And a complete characterization of the solution for SOC minimizing average loss was obtained which reveals that the solutions reported in [34, 43] are complete for the same SOC problem. Altogether the results contribute to clearer descriptions of the two classes of solutions and also their relations (as referred to Corollary 7.2).

Chapter 8

Local SOC of Constrained Processes

The available methods for selection of CVs using the concept of SOC have been developed under the restrictive assumption that the set of active constraints remains unchanged for all the allowable disturbances and implementation errors. To track the changes in active constraints, the use of split-range controllers and parametric programming has been suggested in literature. An alternative heuristic approach to maintain the variables within their allowable bounds involves the use of cascade controllers. In this chapter, we propose a different strategy, where CVs are selected as linear combinations of measurements to minimize the local average loss, while ensuring that all the constraints are satisfied over the allowable set of disturbances and implementation errors. This result is extended to select a subset of the available measurements, whose combinations can be used as CVs. In comparison with the available methods, the proposed approach offers simpler implementation of operational policy for processes with tight constraints. We use the case study of forced-circulation evaporator to illustrate the usefulness of the proposed method.

8.1 Introduction

Local methods, which employ linearized process model and quadratic approximation of the loss function, have been used to find promising CV candidates [34, 40, 43-44, 46]. An assumption involved in the development of exact local methods is that the set of active

constraints does not change during the operation. This assumption is not always satisfied in practice, where it may be optimal to keep different sets of variables at their limits for different disturbance scenarios. For heat exchanger networks described using linear models, Lersbamrungsuk *et al.* [16] suggested the use of split-range controllers to track the set of active input constraints. For the general case involving the input and output constraints, Manum [94] proposed the use of multi-parametric programming [17] to identify the regions with different sets of active constraints and to select CVs for each region separately. However, this approach requires switching between different regions, which can be difficult in the presence of measurement noise. As an alternate approach, Cao [18] proposed the use of cascade control strategy to keep the variables within their allowable bounds. In this approach, the CVs identified based on the concept of SOC are placed in the outer loop and the variable likely to violate the constraint in the inner loop of the cascade controller. The use of cascade control strategy is heuristic, as the presence of constraints is not accounted for during the CV selection. Furthermore, this approach is only applicable, when the number of constraints, which are likely to be active or inactive depending on the disturbance scenario, is not more than the number of CVs. In summary, the available approaches for handling changes in active constraint set are either not general enough or their practical usage is difficult. The use of split-range controllers, multi-parametric programming or cascade controllers also contradicts the goal of SOC of devising ‘simple’ implementation policy for near-optimal operation of the process.

This chapter proposes a fundamentally different approach for handling the possible changes in active constraint set. Instead of tracking the active constraint set, we aim at finding CVs, whose control ensures that the variables are always kept within their allowable bounds for all disturbance and implementation error scenarios. The resulting ‘passive’ approach maintains the simplicity of the control structure and can be seen as a

viable alternative to the use of the available strategies [16, 18, 94], where the penalty (measured in terms of loss) of not tracking the optimal set of active constraint set is not very high.

We present an exact local method, where linear combinations of measurements are selected as CVs such that the local average loss is minimized subject to process constraints. It has earlier been noted in literature that the use of combinations of a few measurements as CVs can often provide similar loss as the case where combinations of all available measurements are used [34, 43-46]. We extend the proposed approach to identify the locally optimal subset of available measurements, whose linear combinations can be used as CVs. The resulting formulation is a mixed integer cone program and can be solved efficiently by available software, *e.g.* using the *bnb* function in YALMIP [95], which implements a branch and bound algorithm. The case study of forced-circulation evaporator [43, 96] is used to demonstrate the usefulness of the proposed approach.

The rest of the chapter is organized as follows: A brief overview of the available exact local method for SOC is presented in Section 8.2. The exact local method is extended for handling constraints in Section 8.3. The case study of forced-circulation evaporator is presented in Section 8.4. Finally, conclusions are drawn in Section 8.5.

8.2 Local SOC

We consider that the optimal operation of the process requires solving the following steady-state optimization problem:

$$\min_u J(u, d), \quad (8.1)$$

where $u \in \mathfrak{R}^{n_u}$ and $d \in D$ denote the inputs (or degrees of freedom) and disturbances, respectively, and D is the domain of d . The scalar J refers to the (economic) cost function, which needs to be minimized. The optimization problem in (8.1) implicitly

assumes that all the constraints remain active for all $d \in D$ and are controlled, and the internal states of the process have been eliminated using these constraints and model equations. In this sense, u denotes the ‘remaining’ degrees of freedom; see Skogestad [13] for further details.

For every $d \in \mathcal{D}$, the optimization problem in (8.1) can be solved online to update u . An alternative and simpler approach to update u in the presence of disturbances involves the use of a feedback controller to hold the CVs (c) at setpoint (c_s), *i.e.*,

$$c = h(\tilde{y}) = c_s. \quad (8.2)$$

In (8.2), $\tilde{y}$ denotes the measured outputs given as $\tilde{y} = y + e$, where

$$y = f_y(u, d), \quad (8.3)$$

and $e \in \mathcal{E}$ denotes the implementation error arising due to measurement error. The use of feedback-based policy results in a loss, which is given as

$$L(d, e) = J_c(d, e) - J_{opt}(d), \quad (8.4)$$

where $J_{opt}(d)$ and $J_c(d, e)$ denote the values of the objective function obtained by solving the optimization problem in (8.1) and by holding c at c_s , respectively. The loss depends on the choice of c and the aim of SOC is to find appropriate CVs which minimize the loss.

In the local methods, the process model is linearized around the nominal optimal operating point (u^*, d^*) to obtain

$$\Delta y = G_y \Delta u + G_y^d \Delta d, \quad (8.5)$$

$$\Delta \tilde{y} = \Delta y + \Delta e, \quad (8.6)$$

where G_y and G_y^d are $\partial f_y / \partial u$ and $\partial f_y / \partial d$, evaluated at the nominal operating point, respectively, and Δ denotes the deviation variables. The deviation in CVs (Δc) is given as

$$\Delta c = H\Delta\tilde{y}, \quad (8.7)$$

with $H \in \mathcal{R}^{n_u \times n_y}$ being a selection or combination matrix. Here, HG_y is assumed to be non-singular, which is necessary to ensure that c can be maintained at c_s by manipulating the inputs using a controller with integral action.

Let $\Delta d = W_d \bar{d}$ and $\Delta e = W_e \bar{e}$, where the diagonal matrices W_d and W_e contain the expected magnitudes of disturbances and measurement errors, respectively. We consider that the allowable set for $\bar{d}$ and $\bar{e}$ is given as

$$\left\| \begin{bmatrix} \bar{d}^T & \bar{e}^T \end{bmatrix}^T \right\|_{\infty} \leq 1, \quad (8.8)$$

which allows the individual elements of $\bar{d}$ and $\bar{e}$ to lie within ± 1 . The local average loss ($L_{average}$) over the allowable set in (8.8) is given as [43]:

$$L_{average}(H) = \frac{1}{6} \left\| J_{uu}^{1/2} (HG_y)^{-1} HY \right\|_F^2, \quad (8.9)$$

where $\|\cdot\|_F$ denote the Frobenius norm and

$$Y = \begin{bmatrix} (G_y^d - G_y J_{uu}^{-1} J_{ud}) W_d & W_e \end{bmatrix}. \quad (8.10)$$

Here, $J_{uu} = \partial^2 J / \partial u^2$ and $J_{ud} = \partial^2 J / (\partial u \partial d)$ are partial derivatives of J evaluated at the nominal operating point. Note that $J_{uu}^{1/2}$ is guaranteed to exist as J_{uu} is positive definite. The expressions for $L_{average}$ for other allowable sets of d and e are given by Kariwala *et al.* [43], which differ from the expression in (8.9) by scalar constants. The expression for local worst-case loss is given by Halvorsen *et al.* [40]. We suggest the selection of CVs through minimization of average loss, as the worst case may not occur frequently in practice [43].

When individual measurements are used as CVs, the elements of H are limited to be 0 or 1 and $HH^T = I$. When combinations of measurements are used instead, the elements of H are allowed to take any value provided that the condition $\text{rank}(H) = n_u$ is satisfied. An

explicit expression to obtain optimal H , which minimizes local average loss in (8.9), is given as [34]

$$H = J_{uu}^{1/2} \left((G_y)^T (YY^T)^{-1} G_y \right)^{-1} (G_y)^T (YY^T)^{-1}, \quad (8.11)$$

where Y is defined in (8.10) and YY^T is assumed to have full rank. This assumption is easily satisfied in practice, as all measurements have error and thus the diagonal elements of W_e are non-zero.

Remark 8.1 *Alstad et al. [34] have shown that if H is an optimal combination matrix, then so is QH , where $Q \in \mathbb{R}^{n_u \times n_u}$ is any nonsingular matrix. Thus, by defining $Q = J_{uu}^{1/2} \left((G_y)^T (YY^T)^{-1} G_y \right)^{-1}$, the expression for optimal combination matrix, which minimizes average loss in (8.9), can be simplified as $H = (G_y)^T (YY^T)^{-1}$.*

The local method described earlier assumes that the set of active constraints does not change with disturbances limiting its application. In general, it may be optimal to keep different sets of variables at their limits for different disturbance scenarios [94]. The use of available approaches, *i.e.* split-range controllers [16], cascade controllers [18] or multi-parametric programming [94], however, leads to a control structure with increased complexity. In the next section, we propose an alternate approach to handle operation constraints, which maintains the simplicity of the control structure.

8.3 Local SOC with Constraints

In this section, we extend the available exact local method for CV selection to account for the presence of constraints.

8.3.1 Exact Local Method

We consider that the following constraints are imposed on the optimization problem in (8.1):

$$z = f_z(u, d) \leq b, \quad (8.12)$$

where $z, b \in \mathfrak{R}^{n_z}$. In general, z can consist of u and y , as well as states, which may not be measured online. Let us denote $z^* = f_z(u^*, d^*)$. Based on the linearized model, the constraint (8.12) can equivalently be expressed in terms of Δu and Δd as

$$\Delta z = G_z \Delta u + G_z^d \Delta d \leq b', \quad (8.13)$$

where G_z and G_z^d are $\partial f_z / \partial u$ and $\partial f_z / \partial d$, evaluated at the nominal operating point, respectively, and $b' = b - z^*$. Based on (8.5)-(8.7), maintaining $c = c_s$, *i.e.* $\Delta c = 0$, requires

$$\Delta u = -(HG_y)^{-1} H \begin{bmatrix} G_y^d W_d & W_e \end{bmatrix} \begin{bmatrix} \bar{d} \\ \bar{e} \end{bmatrix}, \quad (8.14)$$

which implies that

$$\Delta z = \left(-G_z (HG_y)^{-1} H \begin{bmatrix} G_y^d W_d & W_e \end{bmatrix} + G_z^d \begin{bmatrix} W_d & 0 \end{bmatrix} \right) \begin{bmatrix} \bar{d} \\ \bar{e} \end{bmatrix}. \quad (8.15)$$

Now, the CVs can be selected by minimizing the local loss expressed in (8.9), while ensuring that the constraints in (8.13) are satisfied over the allowable set of $\bar{d}$ and $\bar{e}$. By dropping the scalar term in (8.9), the SOC problem with constraints can be formulated as

$$\begin{aligned} \min_H & \left\| J_{uu}^{1/2} (HG_y)^{-1} H Y \right\|_F^2 \\ \text{s.t.,} & \left(-G_z (HG_y)^{-1} H \begin{bmatrix} G_y^d W_d & W_e \end{bmatrix} + G_z^d \begin{bmatrix} W_d & 0 \end{bmatrix} \right) \begin{bmatrix} \bar{d} \\ \bar{e} \end{bmatrix} \leq b', \\ & \left\| \begin{bmatrix} \bar{d}^T & \bar{e}^T \end{bmatrix}^T \right\|_\infty \leq 1. \end{aligned} \quad (8.16)$$

The optimization problem in (8.16) is nonlinear in H and thus is difficult to be solved directly. To overcome this difficulty, we perform a transformation to obtain an equivalent convex problem in the next proposition. This transformation was earlier adopted by Alstad *et al.* [34] to obtain an explicit solution for the unconstrained exact local method, but a formal proof was not provided.

Proposition 8.1 *The global optimal solution of the optimization problem in (8.16) can be obtained by solving*

$$\begin{aligned} \min_H & \|J_{uu}^{1/2} H Y\|_F^2 \\ \text{s.t., } & B' [\bar{d}^T \quad \bar{e}^T]^T \leq b', \\ & H G_y = I, \\ & \|[\bar{d}^T \quad \bar{e}^T]^T\|_\infty \leq 1, \end{aligned} \quad (8.17)$$

where Y is given in (8.10), which is independent of H , and

$$B' = -G_z H \begin{bmatrix} G_y^d W_d & W_n \end{bmatrix} + G_z^d \begin{bmatrix} W_d & 0 \end{bmatrix}. \quad (8.18)$$

Proof. For simplicity of notation, we refer to H in (8.17) as H' in the subsequent discussion. Let the constraint $(H G_y)^{-1} H G_y = I$ be added to (8.16), which does not affect the solution, and define the new variable H' as $H' = (H G_y)^{-1} H$. Then, the optimal solution of (8.16) can be obtained by solving the optimization problem in (8.17), provided there exists H^* that satisfies $(H^* G_y)^{-1} H^* = H'^*$ or equivalently

$$H^* (I - G_y H'^*) = 0. \quad (8.19)$$

Since the matrix $H'^* G_y$ with dimensions $n_u \times n_u$ is the same as the Identity matrix, the matrix $G_y H'^*$ with dimensions $n_y \times n_y$ has an eigenvalue of one with multiplicity n_u and an eigenvalue of zero with multiplicity $n_y - n_u$ [97]. Thus, the matrix $I - G_y H'^*$ has a null space of dimension n_u , which ensures the existence of H^* that satisfies (8.19). This shows that the solution to the optimization problem in (8.16) can be obtained by

solving (8.17). The global optimality of the solution follows from the fact that the optimization problem in (8.17) is convex. $\square$

From (8.8), we note that the allowable set of $\bar{d}$ and $\bar{e}$ defines a hypercube. Thus, the elements of $\Delta z = B'[\bar{d}^T \ \bar{e}^T]^T$ attain their largest values when the individual elements of $\bar{d}$ and $\bar{e}$ are either 1 or -1 [98-99]. Then, the optimization problem in (8.17) can be further simplified as

$$\begin{aligned} \min_H & \|J_{uu}^{1/2} H Y\|_F^2 \\ \text{s.t., } & \|B'_i\|_1 \leq b'_i, \ i = 1, 2, \dots, n_z, \\ & H G_y = I, \end{aligned} \quad (8.20)$$

where $\|\bullet\|_1$ denotes the vector norm computed as the sum of the absolute values of the elements of the vector. The inequality constraints in (8.20) can be expressed as linear constraints on H [99]. Thus, the optimization problem in (8.20) is convex, which can be solved easily to obtain the optimal combination matrix H^* , based on which the CVs can be selected as $c = Q H^* \tilde{y}$, where $Q \in \Re^{n_u \times n_u}$ is any nonsingular matrix.

Remark 8.2 *In general, a variable may be constrained to lie between its upper and lower bounds, e.g. $\underline{y}_i \leq y_i \leq \bar{y}_i$. For such constraints, we can define $z = [-y_i \ y_i]^T$ and $b = [-\underline{y}_i \ \bar{y}_i]^T$. For the optimization problem in (8.20) involving linearized model, these constraints are equivalent, if the upper and lower bounds are symmetric around y_i^* , i.e. $y_i^* - \underline{y}_i = \bar{y}_i - y_i^*$. On the other hand, only the lower bound is relevant for the optimization problem in (8.20) with the upper bound being redundant, if $y_i^* - \underline{y}_i \leq \bar{y}_i - y_i^*$; and vice versa.*

8.3.2 Measurement Subset Selection

As mentioned earlier, the use of combinations of fewer measurements as CVs, which give similar loss in comparison with combinations of all measurements, is preferable as it allows simpler implementation. For measurement subset selection, we note that a column of $(HG_y)^{-1}H$ is zero if and only if the corresponding column of H is zero. Thus, the transformation proposed in Proposition 8.1 still applies and the measurement subset consisting of n elements can be selected by including the following constraints in the optimization problem in (8.20):

$$\begin{aligned} \sum_{j=1}^{n_y} \sigma_{\tilde{y}_j} &= n, \quad \sigma_{\tilde{y}_j} \in \{0, 1\}, \\ -M\sigma_{\tilde{y}_j} &\leq H_{ij} \leq M\sigma_{\tilde{y}_j}, \quad i = 1, 2, \dots, n_u \end{aligned} \quad (8.21)$$

where $\sigma_{\tilde{y}_j} = 1$ if $\tilde{y}_j$ is included in the measurement set to be combined as CVs and 0 otherwise, and M is a large number satisfying $M \geq |H_{ij}|$ for $\forall i, j$. The constraints in (8.21) are motivated by previous work [52, 100], which are derived using the big-M method to convert the logical constraints into constraints involving binary variables. These constraints imply that

- (a) the number of nonzero columns of H is equal to n , and
- (b) if $\tilde{y}_j$ is not selected then all the elements of the j th column of H must be zero; otherwise, the j th column of H is unconstrained.

Now, the overall problem can be written as

$$\begin{aligned} \min_H & \|J_{uu}^{1/2}HY\|_F^2 \\ \text{s.t.}, & \|B'_i\|_1 \leq b'_i, \quad i = 1, 2, \dots, n_z, \\ & HG_y = I, \\ & \sum_{j=1}^{n_y} \sigma_{\tilde{y}_j} = n, \quad \sigma_{\tilde{y}_j} \in \{0, 1\}, \\ & -M\sigma_{\tilde{y}_j} \leq H_{ij} \leq M\sigma_{\tilde{y}_j}, \quad i = 1, 2, \dots, n_u. \end{aligned} \quad (8.22)$$

The optimization problem in (8.22) is a mixed integer cone program and can be solved efficiently using available software. In this paper, we use the *branch and bound* algorithm available in YALMIP [95], where Sedumi [101] is used for solving the cone program obtained upon relaxation of binary variables.

We note that the local methods are meant for pre-screening promising candidate CVs and further validation using the nonlinear model of the process is necessary for the final selection of CVs. This motivates determining a few ‘top’ solutions of the optimization problem in (8.22). In the following discussion, we present a simple approach to determine m solutions of H which give the least losses in increasing order. Let $\sigma = [\sigma_{\hat{y}_1} \quad \sigma_{\hat{y}_2} \quad \dots \quad \sigma_{\hat{y}_{n_y}}]^T$ and σ^l denote the l th best solution, where $l = 1, 2, \dots, m$. The l th best solution is obtained by solving (8.22) with the following additional constraints

$$(\sigma^p)^T \sigma^l \leq n-1, \quad p = 1, 2, \dots, l-1. \quad (8.23)$$

The constraint in (8.23) ensures that the l th solution is not the same as the $l-1$ solutions found earlier. Thus, by solving the optimization problem in (8.22) with the additional constraint in (8.23) with increasing l , the m solutions providing the least losses can be obtained sequentially. The final set of CVs can be selected from these m solutions through loss evaluation using the nonlinear model.

Remark 8.3 *Although an arbitrarily large M can be used for solving (8.22) in theory, numerical considerations require a ‘sufficiently small’ M to obtain a correct solution [95]. In this work, we iteratively solve the optimization problem until the absolute value of the largest element of H is at most 1% lower than the chosen M . Development of a more efficient algorithm, e.g. using customized branch and bound algorithm [47-49], to solve the measurement subset selection problem is an issue for future research.*

8.4 Case Study: Forced Circulation Evaporator

We consider forced-circulation evaporation process [43, 96] to demonstrate the usefulness of the proposed approach. The schematic of this process is shown in Figure 8.1. In this process, dilute solution is pumped upwards through the vertical heat exchanger, while steam flows in counter-current direction as the heating fluid to evaporate the solvent, thus increasing the concentration of the solution. A part of this concentrated solution is circulated back to the evaporator, while the rest is drawn as product.

Figure 8.1 Schematic of forced-circulation evaporator.

The operational objective of this process involves minimizing

$$J = 600F_{100} + 0.6F_{200} + 1.009(F_2 + F_3) + 0.2F_1 - 4800F_2 \quad (8.24)$$

which denotes negative profit. In (8.24), the first four terms are related to the costs of steam, water, pumping and raw material. The last term is related to the revenue obtained by selling the product. The following constraints need to be satisfied:

$$\begin{aligned}
& X_2 \geq 35.5 \\
40 \leq & P_2 \leq 80 \\
& P_{100} \leq 400 \\
0 \leq & F_{200} \leq 400 \\
0 \leq & F_1 \leq 20 \\
0 \leq & F_3 \leq 100
\end{aligned} \tag{8.25}$$

This process has eight degrees of freedom (DOF), among which three (X_1 , T_1 and T_{200}) are disturbances. The remaining five variables F_1 , F_2 , P_{100} , F_3 , and F_{200} are manipulated variables. The case where $X_1 = 5\%$, $T_1 = 40^\circ\text{C}$, and $T_{200} = 25^\circ\text{C}$ is taken as the nominal operating point. Solving the optimization problem in (8.24)-(8.25) for these nominal disturbances results in optimum negative profit of -582.233 \$/h. The corresponding nominally optimal values of different variables are shown in Table 8.1.

Table 8.1 Variables and optimal values

Var.	Description	Value	Var.	Description	Value
F_1	Feed flowrate	9.47 kg/min	L_2	Separator level	1.00 meter
F_2	Product flowrate	1.33 kg/min	P_2	Operating pressure	51.41 kPa
F_3	Circulating flowrate	24.72 kg/min	F_{100}	Steam flowrate	9.43 kg/min
F_4	Vapor flowrate	8.14 kg/min	T_{100}	Steam temperature	151.52 °C
F_5	Condensate flowrate	8.14 kg/min	P_{100}	Steam pressure	400.00 kPa
X_1	Feed composition	5.00 %	Q_{100}	Heat duty	345.29 kW
X_2	Product composition	35.50 %	F_{200}	C.W. flowrate	217.74 kg/min
T_1	Feed temperature	40.00 °C	T_{200}	Inlet C.W. temp.	25.00 °C
T_2	Product temperature	88.40 °C	T_{201}	Outlet C.W. temp.	45.55 °C
T_3	Vapor temperature	81.07 °C	Q_{200}	Condenser duty	313.21 kW

Degrees of Freedom (DOF) Analysis. The constraints on X_2 and P_{100} remain active over the entire set of allowable disturbances. In addition, separator level (L_2), which has no steady-state effect, needs to be stabilized at its nominal setpoint, which consumes one DOF. After control of active constraints and L_2 , two inputs (u) remain. Without loss of generality, they are taken as F_1 and F_{200} . For these inputs, we consider that 2 CVs are to be chosen as a subset or combinations of the following available measurements:

$$y = [P_2 \quad T_2 \quad T_3 \quad F_2 \quad F_{100} \quad T_{201} \quad F_5 \quad F_{200} \quad F_1]^T \quad (8.26)$$

Note that the pump *circulation* flow (F_3) is not included in y , as the linear model for this measurement results in large plant-model mismatch due to linearization [43].

Local Analysis. The allowable disturbance set corresponds to $\pm 5\%$ variation in X_1 and $\pm 20\%$ variation in T_1 and T_{200} around their nominal values. Based on these variations, we have $W_d = \text{diag}(0.25, 8, 5)$. The measurement errors for the pressure and flow measurements are taken to be $\pm 2.5\%$ and $\pm 2\%$, respectively, of the nominal operating values. For temperature measurements, this error is considered to be $\pm 1^\circ\text{C}$. Accordingly, we have $W_e = \text{diag}(1.29, 1, 1, 0.03, 0.19, 1, 0.16, 4.36, 0.19)$. The Hessian and gain matrices for this process are given in the reference [43].

For CV selection, the constraints on P_2 , F_{200} , F_1 and F_3 need to be considered. Based on the constraint limits in (8.25) and the nominal values shown in Table 8.1, we note that the lower bounds on P_2 , F_1 , F_3 and the upper bound on F_{200} is more restrictive than the corresponding upper bounds and lower bound, respectively. Thus, based on Remark 8.2, we define $z = [-P_2 \quad F_{200} \quad -F_1 \quad -F_3]^T$ and $b = [-40 \quad 400 \quad 0 \quad 0]^T$, which implies that $b' = b - z^* = [11.41 \quad 182.26 \quad 9.47 \quad 24.72]^T$.

First, the best individual measurements are selected by available local SOC and are found to be $c'_2 = [F_{100} \quad F_{200}]^T$ with average local loss being 19.50 \$/h. When linear combinations of all the 9 measurements are used, the average local loss decreases to 3.01

\$/h. A similar trend is observed for the proposed approach, *i.e.* the best individual measurements $c_2 = [F_{100} \ T_{201}]^T$ result in an average local loss of 22.16 \$/h, which reduces to 10.85 \$/h, when combinations of all the measurements are used as CVs. Results from both these approaches signify that controlling combinations of measurements can lead to substantial reduction in loss.

Combining fewer measurements as CVs, which gives similar loss as the loss obtained using combinations of all the available measurements, is practically desirable. The combinations of n out of 9 measurements ($n \leq 9$), which give the smallest average local loss for available exact local method were found using the branch and bound method [49]. A similar analysis is carried out for the proposed approach by solving the optimization problem in (8.22) for different values of n . The results are presented in Figure 8.2.

For both approaches, the use of combinations of three or four measurements as CVs offers a reasonable trade-off between simplicity of the control system and the operational loss. Five best candidates for the cases of $n = 3$ and $n = 4$ are obtained using the available and proposed approaches and the results are summarized in Table 8.2.

Figure 8.2 Average local losses of best CV candidates with n measurements obtained using available and proposed (explicit constraint handling) exact local methods.

Table 8.2 Average local and nonlinear losses for the self-optimizing CV candidates

CV candidates selected using available local SOC			CV candidates selected using explicit constraints handling		
Measurements	Average Losses [\$ /h]		Measurements	Average Losses [\$ /h]	
	Local	Nonlinear		Local	Nonlinear
F_2, F_{100}, F_{200}	3.91	3.97	P_2, F_2, F_{200}	16.41	15.13
F_2, F_5, F_{200}	5.96	4.04	T_2, F_2, F_{200}	16.58	15.80
F_2, F_{100}, T_{201}	6.74	7.83	$n = 3$ T_3, F_2, F_{200}	16.65	15.36
F_2, F_{200}, F_1	7.22	4.74	P_2, F_{100}, F_{200}	19.15	17.36
F_2, T_{201}, F_5	8.53	8.15	T_2, F_{100}, F_{200}	19.15	17.50
$F_2, F_{100}, F_5, F_{200}$	3.32	3.03	P_2, F_2, F_5, F_{200}	11.11	8.84
$F_2, F_{100}, F_{200}, F_1$	3.56	3.21	$P_2, F_2, F_{100}, F_{200}$	11.23	9.97
$P_2, F_2, F_{100}, F_{200}$	3.76	3.86	$n = 4$ T_2, F_2, F_5, F_{200}	11.38	9.30
$T_2, F_2, F_{100}, F_{200}$	3.76	3.87	$T_2, F_2, F_{100}, F_{200}$	11.46	10.39
$T_3, F_2, F_{100}, F_{200}$	3.77	3.83	T_3, F_2, F_5, F_{200}	11.49	9.27

Nonlinear Analysis. The losses for all the promising candidates identified using local analysis are computed based on the nonlinear model using 100 scenarios of randomly generated d and e . Note that cascade control is required for the implementation of CVs selected using available local SOC, otherwise P_2 violates the constraints in (8.25) for some disturbances and measurement error scenarios [18]. For implementation of the cascade control strategy, the lower and upper bounds on P_2 are revised to 41.29 and 78.71 kPa, respectively, to account for the measurement errors. The results of the analysis are presented in Table 8.2.

The nonlinear analysis shows that the following CV candidates

$$c'_3 = \begin{bmatrix} -49.44F_2 + 6.22F_{100} + 0.07F_{200} \\ -98.86F_2 + 16.16F_{100} - 0.02F_{200} \end{bmatrix} \quad (8.27)$$

$$c'_4 = \begin{bmatrix} -51.29F_2 + 4.65F_{100} + 2.34F_5 + 0.07F_{200} \\ -104.60F_2 + 11.30F_{100} + 7.25F_5 - 0.02F_{200} \end{bmatrix} \quad (8.28)$$

result in the lowest average losses among the candidates selected using available SOC (3.97 and 3.03 \$/h, respectively). On the other hand, the following candidates result in the lowest average losses among those selected using the proposed approach:

$$c_3 = \begin{bmatrix} 3.84P_2 - 318.66F_2 + 1.36F_{200} \\ 0.16P_2 - 6.10F_2 + 0.01F_{200} \end{bmatrix} \quad (8.29)$$

$$c_4 = \begin{bmatrix} 1.53P_2 - 730.12F_2 + 98.95F_5 + 1.14F_{200} \\ 0.10P_2 - 12.44F_2 + 1.88F_5 + 0.01F_{200} \end{bmatrix} \quad (8.30)$$

Table 8.2 shows that the CV candidates identified using the proposed approach give higher losses in comparison with those identified using the available exact local method and implemented using cascade controller. Nevertheless, the average losses with the use of c_3 and c_4 as CVs (15.13 and 8.84 \$/h, respectively) are relatively small in comparison to the nominal cost, *i.e.* 583.23 \$/h. Thus, the resulting implementation can still be considered to be economically acceptable. An advantage of using the CVs found using the proposed approach is that their implementation does not require additional controllers since all constrained variables remain within their bounds for all the disturbance scenarios. This can be confirmed from Figure 8.3, which shows that the variation of P_2 keeps within the admissible range of 40 kPa to 80 kPa for different CV alternatives. The results also indicate that the proposed approach leads to much smaller variation in P_2 , as due to its conservative design that ensures the variation be admissible even in the worst case.

Figure 8.3 Variation of P_2 with use of CVs obtained using available exact local method with cascade control and the proposed approach.

8.5 Conclusions

We have proposed an approach for systematic selection of CVs in the framework of SOC for processes with tight operation constraints. In this approach, linear combinations of measurements are selected as CVs such that maintaining the CVs at constant setpoints minimizes the local average loss while the constraints are satisfied over the allowable set of disturbances and implementation errors. In comparison with existing approaches, which involve the use of split-range controllers [16], cascade controllers [18], and parametric programming [94], the proposed approach is conservative, but allows for simpler implementation strategy. The use of the proposed approach is attractive, when the penalty (measured in terms of loss) of not tracking the optimal set of active constraints is not very high. The case-study of forced-circulation evaporator showed that the proposed approach can be used to obtain a good trade-off between the economic loss and the complexity of the control system.

Chapter 9

Selecting CVs as Optimal Measurement Combinations via Perturbation Control Approach

SOC has been used to select CVs as the optimal linear combinations of measurements by minimizing economic cost of a steady-state process. But it remains as an open problem to use SOC to select CVs for a dynamic process which does not enter a steady state at all or the transient cost of which must be counted. This chapter proposes the concept of ‘dynamic SOC’ (dSOC) to handle such a problem. The CVs are expressed as linear combinations of measurements and are selected for minimizing a cost defined for the whole operation interval. Given a set of candidate measurement combination matrices, a locally optimal selection of such a matrix is determined via perturbation control approach. Application of dSOC to a linear process is presented to illustrate the usefulness of the theoretical results.

9.1 Introduction

Recently the concept of SOC has been introduced to handle the problem of selecting CVs [13, 37]. SOC determines CVs by minimizing an economic cost defined for a steady-state process in the presence of disturbances and measurement noises (or implementation errors in general), where the CVs are assumed to be linear combinations

of measurements. The available SOC minimizes utility loss or cost increment due to disturbances and measurement noises based on local analysis. As the loss can be defined in different senses, the worst-case loss and the average loss have been investigated in the literature and their corresponding solutions of SOC have been reported [34, 40, 43-44, 46]. Since in practice it is preferred that fewer measurements be used, selecting CVs as combinations of a subset of available measurements have also been investigated [8-10]. Note that so far SOC minimizes costs defined for steady-state processes. For convenience, they are named as *static SOC* (sSOC) hereafter. Application of sSOC to practical control problems has been reported widely and proves to be useful [14].

SSOC is suitable when the operational cost of a process is determined by its steady state. This can be the case if a process operates at a steady state in most of time. However, there are cases in which a process does not enter a steady state at all. Typical examples are the batch processes which keep dynamic during the whole intervals of operation [33, 38]. There are also processes of which the costs during the transient operations count much and must be minimized in addition to the steady-state costs. For such dynamic processes, it is desirable to select CVs for minimizing costs over the whole operation intervals. Indeed this gives rise to a new problem, *dynamic SOC* (dSOC), in contrast to sSOC. So far, few studies have been carried out on dSOC but some initial and tentative ones as reported in [19-20]: The results, however, do not give any general formulation of dSOC; neither do they obtain a complete solution to such a problem. These motivate us to investigate dSOC systematically in this chapter.

We present a general formulation of dSOC for nonlinear processes and solve it for a solution via perturbation control approach (as well developed in control theory [102-103]). While it is too difficult to solve the general dSOC problem for a global optimal solution, we solve it for a local optimal solution by assuming an available set of candidate CVs (or

equivalently, MCMs). We find that, within the framework of dSOC, the optimal selection of CVs is essentially associated with an optimal control law, in sharp contrast to sSOC which is independent of the control law (due to the steady-state assumption). That is, the optimal selection of CVs by dSOC is dependent on the control law as adopted in a particular application. To be specific, in this work we assume a control law with *linear measurement feedback* (LMF), which computes instant control input as appropriate linear combinations of current measurements, endowing very simple implementation. The local optimal LMF control gain is solved and used to derive a solution for the dSOC problem. The results can be extended if other forms of control law are considered, such as state feedback and *linear CV feedback* which uses the current measurements of CVs as feedback signals.

The rest of the chapter is organized as follows. In Section 9.2, the dSOC problem is formulated and its special form is presented by assuming the MCM be restricted to a given set. In Section 9.3, derivation of a local optimal solution for the special dSOC problem is presented in detail, where two subsections are devoted to obtaining a local optimal LMF feedback gain and to selecting the best MCM, respectively. In Section 9.4, application of dSOC to a linear process is presented to illustrate the usefulness of the theoretical results. Finally, Section 9.5 concludes the chapter.

9.2 Problem Formulation

Consider a process described by

$$\dot{x} = f(x, u, w, t), \quad t \geq t_0, \quad (9.1)$$

$$y = h(x) + v, \quad (9.2)$$

where $x \in \mathfrak{R}^n$, $y \in \mathfrak{R}^m$ and $u \in \mathfrak{R}^k$ are the system state, measurement (or measured output, where the output may contain any measurable signals) and control input vectors,

respectively; $w \in \mathfrak{R}^l$ and $v \in \mathfrak{R}^m$ are the system disturbance and measurement noise vectors, respectively. Without loss of generality, we assume that $m \leq n$, i.e., the dimension of the measurements is smaller than that of the states.

Let the economic cost evaluating the process performance take the form of

$$J_0 = \phi_0(x(t_f), t_f) + \int_{t_0}^{t_f} F_0(x, u, t) dt, \quad (9.3)$$

where $[t_0 \ t_f]$ is the time horizon of interest. The cost function $\phi_0(x(t_f), t_f)$ depends on the terminal states and time and $F_0(x, u, t)$ on the intermediate states, control inputs and time. Conventional real-time optimization (RTO) [15] repeatedly solves the optimization problem

$$\begin{aligned} \min_{u(w, v, t)} J_0 \\ \text{s.t., Eq. (9.1),} \end{aligned} \quad (9.4)$$

where the control input $u(w, v, t)$ depend on instant values of the disturbances and noises. As the disturbances and noises change, changes follow in the solution of problem (9.4). RTO requires measurements of the disturbances and noises and also the computational cost is high. To simplify, the following problem may be solved instead

$$\begin{aligned} \min_{u(t)} E(J_0) \\ \text{s.t., Eq. (9.1),} \end{aligned} \quad (9.5)$$

where $E(\bullet)$ is the operator of expectation. Problem (9.5) removes the dependence of u on instant values of the disturbances and noises but requires statistic knowledge of them. Usually the optimal control law can be solved more efficiently as compared to RTO.

DSOC attempts to implement the optimal control in (9.5) in a *suboptimal* manner. Define the CV vector as

$$z = \Gamma y, \quad (9.6)$$

where $z \in \mathfrak{R}^{m'}$ ($m' \leq m$) and $\Gamma \in \mathfrak{R}^{k \times m}$ is a *constant* matrix to be determined. (The CVs

are also known as derived or performance outputs by assuming zero noises [103-104].) Conventional choices of the CV vector are special cases of (9.6), since any measurable or derived signals can be included in the measurements (y). The MCM (Γ) is determined for minimizing the cost when the CV vector is forced to track a reference in the presence of disturbances and measurement noises. In formal words, a dSOC problem can be formulated as an optimization problem defined in (9.5) subject to three additional constraints:

- (i) the CV vector has the same size as the control input, i.e., $m' = k$,
- (ii) for a given CV vector z , it is forced to track a reference $z_r(t)$ and the tracking error (or *tracking cost*) is minimized by optimal control, where $z_r(t)$ equals $\Gamma y_r(t)$ and $y_r(t)$ is a given *nominal optimal* path of $y(t)$,
- (iii) absolute values of the elements in each row of Γ sum up to 1, i.e., if γ_i is the i -th row of Γ then we have

$$\|\gamma_i\|_1 = 1, \forall i = 1, 2, \dots, k. \quad (9.7)$$

Constraint (i) is necessary for perfect tracking of a CV vector to a reference when the control inputs have dimensions of k . Constraints (ii) owes to a merit of dSOC. Note that the tracking error is minimized by optimal control regardless of the choice of Γ . Finally constraint (iii) imposes a normalization condition on the combination matrix to avoid a trivial (or zero) solution, making the dSOC problem be well-posed. The normalization loses no generality since only relative strengths of the measurement combinations matter in deriving the CVs.

The above formulation of dSOC means that the economic cost and the tracking cost are minimized simultaneously. We expect that smallness of the tracking cost would imply smallness of the economic cost. In a sense, we attempt to make the system achieve

‘near-optimal’ performance by maintaining small tracking cost, in spite of process disturbances and measurement noises. Let the tracking cost be evaluated by

$$J'_0 = \phi'_0(z(t_f), z_r(t_f), t_f) + \int_{t_0}^{t_f} F'_0(z, z_r, t) dt, \quad (9.8)$$

where $\phi'_0(z(t_f), z_r(t_f), t_f)$ and $F'_0(z, z_r, t)$ define the tracking costs at the terminal time and over the transient, respectively. The costs are analog to the economic cost defined in (9.3). Therefore dSOC solves the problem

$$\begin{aligned} & \left\{ \min_{u(t)} E(J_0), \min_{u(t), \Gamma} E(J'_0) \right\} \\ & \text{s.t., Eqs. (9.1)-(9.2) and (9.6)-(9.7),} \end{aligned} \quad (9.9)$$

where the first optimization involves a single decision variable, the control input $u(t)$, and the second optimization have two decision variables, $u(t)$ and the MCM Γ . Problem (9.9) is very difficult to solve in general. In the following, we consider a special case in which a practical solution can be obtained.

In industrial applications, it is usual that a couple of candidate CV vectors are known a priori and the job is to select one for best operation. Let us assume that a set of candidate Γ 's (or equivalently, CV vectors) are available, satisfying the constraint in (9.7). The optimal Γ is then selected from these available candidates, minimizing the economic cost. To solve the dSOC problem, the optimizations in (9.9) are solved for each candidate Γ , where it is essential to find an optimal control law for each given Γ . Let the set of candidate Γ 's be denoted by Ξ . The dSOC problem becomes

$$\min_{\Gamma \in \Xi} E(J_{0,opt}), \quad (9.10)$$

where for a given Γ , $E(J_{0,opt})$ is the cost achieved when $u(t) = u_{opt}(t)$ and

$$\begin{aligned} u_{opt}(t) &:= \arg \min_{u(t)} \{E(J_0), E(J'_0)\} \\ &\text{s.t., Eqs. (9.1)-(9.2) and (9.6).} \end{aligned} \quad (9.11)$$

Problem (9.11) has two objectives to be minimized at the same time, which in general leads to a set of Pareto optimal solutions. Regularization is usually adopted for a unique solution [105]. Regulate the objectives as $E(J_0 + \mu J'_0)$, where $\mu \geq 0$ is a scalar specified for a desired tradeoff between the two objectives. As μ and J'_0 are both specified, J'_0 may absorb μ . Hence a new objective J can be defined as the sum of J_0 and J'_0 . Therefore (9.11) becomes

$$\begin{aligned} u_{opt}(t) &:= \arg \min_{u(t)} E(J) \\ \text{s.t., Eqs. (9.1)-(9.2) and (9.6),} \end{aligned} \quad (9.12)$$

where

$$\begin{aligned} J &:= \phi(x(t_f), z(t_f), z_r(t_f), t_f) + \int_{t_0}^{t_f} F(x, u, z, z_r, t) dt, \\ \phi(x(t_f), z(t_f), z_r(t_f), t_f) &:= \phi_0(x(t_f), t_f) + \phi'_0(z(t_f), z_r(t_f), t_f), \\ F(x, u, z, z_r, t) &:= F_0(x, u, t) + F'_0(z, z_r, t). \end{aligned} \quad (9.13)$$

The optimizations in (9.10) and (9.12) describe the dSOC problem when a set of candidate MCMs are given. The solution of Γ to (9.10) will determine the CV vector as in (9.6). From the formulation, a key observation is that *an optimal MCM is essentially associated with an optimal control law*. Different forms of the control law (e.g., in the form of state or output feedback), which are imposed as additional constraints on (9.12), may lead to different solutions of MCM. This differs from sSOC of which the solution is independent of the control law [13, 34, 43]. In the next section, we present a local optimal solution to the dSOC problem by considering a particular form of the control law.

9.3 Local Optimal Solution

We solve the dSOC problem via perturbation control approach. Given a candidate MCM, the approach assumes a *nominal* optimal solution, and then linearizes the process and cost equations around the nominal optimal path, and consequently finds an optimal

control law minimizing the cost increment arising from perturbation. The local optimal solution of Γ is then obtained as the candidate Γ giving minimal cost increment when a corresponding optimal perturbation control is applied.

9.3.1 Optimal Perturbation Control Law

By adjoining the equation constraint in (9.1) to the cost function with a Lagrange multiplier, problem (9.12) is converted into

$$\min_{u(t)} E(\tilde{J}) := E\left(\phi(x(t_f), y_r(t_f), v(t_f), t_f, \Gamma) + \int_{t_0}^{t_f} (H - \lambda^T \dot{x}) dt\right), \quad (9.14)$$

where the Hamiltonian function

$$H := F(x, u, y_r, v, t, \Gamma) + \lambda^T f(x, u, w, t). \quad (9.15)$$

In (9.15), the scalar $\tilde{J}$ denotes the augmented cost, the vector $\lambda \in \mathfrak{R}^n$ a Lagrange multiplier, and y_r the nominal optimal path of y . The arguments z 's of the above functions have all been replaced by x 's, Γ 's and v 's, using (9.2) and (9.6). The function names are abused for convenience.

The perturbation control approach assumes control input of the form

$$u(t) = u^*(t) + \delta u(t), \quad (9.16)$$

where $u^*(t)$ is the optimal control for the system under nominal conditions and $\delta u(t)$ denotes the perturbation control to suppress the state deviation which is to be determined. Under nominal conditions, both the perturbation control $\delta u(t)$ and the tracking cost $E(J'_0)$ vanish, regardless of the choice of Γ . Note that dSOC depends on a form of the perturbation control law. *Dynamic measurement feedback*,² with the form

² The word 'output' has been widely used in literature to mean 'measured output' or 'measurement'. In this

$\delta u(s) = -K(s)\delta y(s)$ (in the frequency domain) as widely used in classic LQG control, could be an ideal choice. However, this kind of control law is seldom adopted in control practice, because of its complexity and cost for implementation [4]. As two simpler alternatives, we may consider the control law with LMF, i.e., $\delta u(t) = -K(t)\delta y(t)$, or *linear CV feedback*, i.e., $\delta u(t) = -K(t)\delta z(t)$. These two control laws use only current measurement deviations as the feedback signals, in contrast to dynamic feedback control that involves historical measurement deviations.

The LMF control law uses the measurements directly for control, which admits maximal utilization of the available information; by contrast, the linear CV-feedback control law uses CVs derived from the measurements for control which restricts utilization of the available information. As a consequence, the LMF control law would give lower cost theoretically; nevertheless, the linear CV-feedback control law admits easier implementation since perfect tracking may be obtained by applying simple control strategies like PID control. In the following, we do analysis and derive results based on the LMF perturbation control law. The analysis and results can easily be extended to the case with linear CV feedback, as will be briefly remarked at the end of this section.

Consider the LMF perturbation control,

$$\delta u(t) = -K(t)\delta y(t), \quad (9.17)$$

where $\delta y(t) := y(t) - y_r(t)$, and $K(t)$ is a time-varying feedback gain. (State feedback is a special case of (9.17) when $h(x) := x$ and v is constant in (9.2).) The gain $K(t)$ is determined for minimizing the cost increment due to perturbation, which is implicitly dependent on Γ . Since the minimal cost is always achieved for $\Gamma = 0$ if the constraints in (9.7) vanish, the constraints are imposed to make the dSOC problem be well-posed.

work, we use ‘output’ and ‘measurement’ to mean ‘true output’ and ‘measured output’, respectively.

Let the nominal initial state, disturbance and measurement noise be $x(t_0) = x^*(t_0)$, $w(t) = w^*(t)$ and $v(t) = v^*(t)$, respectively. Suppose that we have determined an optimal control law $u^*(t)$ that solves problem (9.14) (which is usually obtained by solving the equations arising from Pontryagin's Minimum Principle or Hamilton-Jacobi-Bellman formulation, or by directly solving the optimization problem using numerical methods [102-103]). This results in a nominal optimal path with $(x(t), \lambda(t)) = (x^*(t), \lambda^*(t))$, an economic cost $J_{0,opt}^*$, and an augmented cost $\tilde{J}^*$. Consider small perturbations from the nominal path produced by small changes in the initial states $\delta x(t_0)$, in the disturbances $dw(t)$ and the measurement noises $dv(t)$. We expect that such perturbations will give rise to perturbations $\delta x(t)$ and $\delta \lambda(t)$. (The relation between total and fixed variations of a variable, denoted by $dx(t)$ and $\delta x(t)$ respectively, is $dx(t)|_{t=t_*} = \delta x(t)|_{t=t_*} + \dot{x}(t)|_{t=t_*} dt_*$ where t_* is any value between t_0 and t_f [103].) Around the nominal optimal path expand the augmented cost $\tilde{J}$ in (9.14) to second order (since all first-order terms vanish about the optimal path) and all the constraints to first order. We have

$$\min_{K(t)} E(\tilde{J}) \approx J^* + \min_{K(t)} E\left(\lambda^{*T}(t_0)\delta x(t_0) + \delta^2 \tilde{J}\right), \quad (9.18)$$

where $\lambda^{*T}(t_0)\delta x(t_0)$ is the first-order cost increment due to changes in initial states [102]; and

$$\begin{aligned} \delta^2 \tilde{J} = & \frac{1}{2} \begin{bmatrix} \delta x^T(t_f) & dv^T(t_f) \end{bmatrix} \begin{bmatrix} \phi_{xx}(t_f) & \phi_{xv}(t_f) \\ \phi_{vx}(t_f) & \phi_{vv}(t_f) \end{bmatrix} \begin{bmatrix} \delta x(t_f) \\ dv(t_f) \end{bmatrix} \\ & + \frac{1}{2} \int_{t_0}^{t_f} \begin{bmatrix} \delta x^T & \delta u^T & dw^T & dv^T \end{bmatrix} \begin{bmatrix} H_{xx} & H_{xu} & H_{xw} & H_{xv} \\ H_{ux} & H_{uu} & H_{uw} & H_{uv} \\ H_{wx} & H_{wu} & H_{ww} & H_{wv} \\ H_{vx} & H_{vu} & H_{vw} & H_{vv} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta u \\ dw \\ dv \end{bmatrix} dt, \end{aligned} \quad (9.19)$$

subject to the constraint in (9.17) and

$$\delta \dot{x} = f_x \delta x + f_u \delta u + f_w dw, \quad t \geq t_0, \quad (9.20)$$

$$\text{given } \delta x(t_0), dw, \text{ and } dv. \quad (9.21)$$

In the above equations, the symbols $\bullet_*$ and $\bullet_{* \#}$ denote the derivatives $\partial \bullet / \partial *$ and $\partial^2 \bullet / \partial * \partial \#$ evaluated at the nominal path, respectively; and $J^* = \tilde{J}^*$ is used in (9.18), where J^* is the nominal optimal cost defined in (9.12). The vectors and matrices in the above equations may all be *time-varying*. Suppose that $E(\delta x(t_0)) = 0$. Given J^* , minimizing $E(\tilde{J})$ is thus locally equivalent to minimizing $E(\delta^2 \tilde{J})$.

Define the new symbols as

$$\begin{aligned} dn &:= [dw^T \quad dv^T]^T, \quad C := h_x, \\ A_x &:= f_x - f_u KC, \quad B_n := [f_w \quad -f_u K]. \end{aligned} \quad (9.22)$$

With the expression of δu in (9.17) and the symbols defined in (9.22), equations (9.19) and (9.20) are rewritten as

$$\begin{aligned} \delta^2 \tilde{J} &= \frac{1}{2} \begin{bmatrix} \delta x^T(t_f) & dv^T(t_f) \end{bmatrix} \begin{bmatrix} \phi_{xx}(t_f) & \phi_{xv}(t_f) \\ \phi_{vx}(t_f) & \phi_{vv}(t_f) \end{bmatrix} \begin{bmatrix} \delta x(t_f) \\ dv(t_f) \end{bmatrix} \\ &+ \frac{1}{2} \int_{t_0}^{t_f} \begin{bmatrix} \delta x^T & dn^T \end{bmatrix} \begin{bmatrix} \tilde{H}_{xx} & \tilde{H}_{xn} \\ \tilde{H}_{nx} & \tilde{H}_{nn} \end{bmatrix} \begin{bmatrix} \delta x \\ dn \end{bmatrix} dt, \end{aligned} \quad (9.23)$$

$$\delta \dot{x} = A_x \delta x + B_n dn, \quad t \geq t_0, \quad (9.24)$$

where

$$\begin{aligned} S(t_f) &= \phi_{xx}(t_f), \\ \tilde{H}_{xx} &= \begin{bmatrix} I & -C^T K^T \end{bmatrix} \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix} \begin{bmatrix} I \\ -KC \end{bmatrix}, \\ \tilde{H}_{nx}^T = \tilde{H}_{xn} &= \begin{bmatrix} I & -C^T K^T \end{bmatrix} \begin{bmatrix} H_{xw} & H_{xv} - H_{xu} K \\ H_{uw} & H_{uv} - H_{uu} K \end{bmatrix}, \\ \tilde{H}_{nn} &= \begin{bmatrix} H_{ww} & H_{wv} - H_{wu} K \\ H_{vw} - K^T H_{uw} & H_{vv} + K^T H_{uu} K - K^T H_{uv} - H_{vu} K \end{bmatrix}. \end{aligned} \quad (9.25)$$

Assume that $dv \sim N(0, W_v)$, $dw \sim N(0, W_w)$ and $\delta x(t_0) \sim N(0, P_0)$, which are mutually independent Gaussian white noises. Therefore we have $dn \sim N(0, W_n)$, where $W_n = \text{diag}\{W_w, W_v\}$. We proceed to derive equivalent expressions for the first part, named as $\delta^2 \tilde{J}_1$, and the second part, named as $\delta^2 \tilde{J}_2$, of $\delta^2 \tilde{J}$ in (9.23), respectively, and then combine them to get a more specific expression of $\delta^2 \tilde{J}$.

Firstly, a cross-correlation matrix has to be determined. Note that the solution of (9.24) is given by

$$\delta x(t) = \Phi(t, t_0) \delta x(t_0) + \int_{t_0}^t \Phi(t, \tau) B_n(\tau) dn(\tau) d\tau, \quad t \geq t_0, \quad (9.26)$$

where $\Phi(t, \tau)$ is the state transition matrix of the system (9.24). Based on (9.26) and the assumption that $\delta x(t_0)$ and $dn(t)$ are orthogonal, the cross-correlation matrix $\Omega_{nx}(t, t)$ is computed as

$$\begin{aligned} \Omega_{nx}(t, t) &= E\left(dn(t) \delta x^T(t)\right) \\ &= \int_{t_0}^t E\left(dn(t) dn^T(\tau)\right) B_n^T(\tau) \Phi^T(t, \tau) d\tau = \frac{1}{2} W_n B_n^T. \end{aligned} \quad (9.27)$$

The factor $1/2$ is due to the upper limit of t of the integral (which is 1 if the limit is larger than t).

Expand $\delta^2 \tilde{J}_1$ and take the expectation, yielding

$$\begin{aligned} E(\delta^2 \tilde{J}_1) &= \frac{1}{2} E\left(\delta x^T(t_f) \phi_{xx}(t_f) \delta x(t_f)\right) \\ &\quad + \frac{1}{2} \text{tr}\left(\phi_{vv}(t_f) W_v\right) + \text{tr}\left(\phi_{xv}(t_f) \Omega_{vx}(t_f, t_f)\right), \end{aligned} \quad (9.28)$$

where $\Omega_{vx}(t_f, t_f) := E(v(t_f) x^T(t_f))$. Similar to $\Omega_{nx}(t, t)$, the cross-correlation matrix $\Omega_{vx}(t, t)$ is obtained as

$$\begin{aligned}\Omega_{vx}(t, t) &= E\left(dv(t)\delta x^T(t)\right) \\ &= \int_{t_0}^t E\left(dv(t)dn^T(\tau)\right)B_n^T(\tau)\Phi^T(t, \tau)d\tau = -\frac{1}{2}W_v K^T f_u^T.\end{aligned}\quad (9.29)$$

Hence $\Omega_{vx}(t_f, t_f) = -\frac{1}{2}W_v K^T(t_f)f_u^T(t_f)$, which involves the control gain at the terminal time. This would make the optimization of $E(\delta^2 \tilde{J})$ be very complicated. To simplify, we assume that $\phi_{xv}(t_f) = 0$, i.e., there is no cross term of x and v in the function $\phi(t_f)$.

Consequently the last term in (9.28) vanishes and (9.28) simplifies into

$$E(\delta^2 \tilde{J}_1) = \frac{1}{2}E\left(\delta x^T(t_f)\phi_{xx}(t_f)\delta x(t_f)\right) + \frac{1}{2}\text{tr}\left(\phi_{vv}(t_f)W_v\right). \quad (9.30)$$

To remove the unknown $\delta x(t_f)$, we proceed to get an equivalent expression of $E(\delta^2 \tilde{J}_1)$. Let $S(t)$ be a symmetric matrix satisfying $S(t_f) = \phi_{xx}(t_f)$. Then we have the differential equation [106]:

$$\frac{d}{dt}E(\delta x^T S \delta x) = E(\delta x^T \dot{S} \delta x) + E\left(\delta x^T (SA_x + A_x^T S)\delta x\right) + \text{tr}(SB_n W_n B_n^T). \quad (9.31)$$

Note the identity

$$\int_{t_0}^{t_f} \frac{d}{dt}E(\delta x^T S \delta x)dt = E\left(\delta x^T(t_f)S(t_f)\delta x(t_f)\right) - E\left(\delta x^T(t_0)S(t_0)\delta x(t_0)\right). \quad (9.32)$$

From (9.31) and (9.32), we obtain

$$\begin{aligned}&\frac{1}{2}E\left(\delta x^T(t_f)S(t_f)\delta x(t_f)\right) \\ &= \frac{1}{2}\left(\text{tr}\left(S(t_0)P_0\right) + \text{tr}\int_{t_0}^{t_f}(\dot{S} + A_x^T S + SA_x)Pdt + \text{tr}\int_{t_0}^{t_f}SB_n W_n B_n^T dt\right),\end{aligned}\quad (9.33)$$

where $P(t) = E(\delta x \delta x^T)$ which defines the covariance of δx and satisfies $P(t_0) = P_0$.

Substitute this expression into (9.30), yielding the desired expression of $E(\delta^2 \tilde{J}_1)$,

$$E(\delta^2 \tilde{J}_1) = \frac{1}{2}\left(\text{tr}\left(\phi_{vv}(t_f)W_v\right) + \text{tr}\left(S(t_0)P_0\right) + \text{tr}\int_{t_0}^{t_f}(\dot{S} + A_x^T S + SA_x)Pdt + \text{tr}\int_{t_0}^{t_f}SB_n W_n B_n^T dt\right). \quad (9.34)$$

Next, we derive an equivalent form of $E(\delta^2 \tilde{J}_2)$, namely the mean of the integral part of $\delta^2 \tilde{J}$ as given in (9.23). Given $\Omega_{nx}(t, t)$ in (9.27), we expand $E(\delta^2 \tilde{J}_2)$ and obtain

$$E(\delta^2 \tilde{J}_2) = \frac{1}{2} \text{tr} \int_{t_0}^{t_f} (\tilde{H}_{xx} P + \tilde{H}_{xn} W_n B_n^T + \tilde{H}_{nn} W_n) dt, \quad (9.35)$$

where the ‘tr’ operation acts on the matrix resulted from the integration.

Adding up $E(\delta^2 \tilde{J}_1)$ in (9.34) and $E(\delta^2 \tilde{J}_2)$ in (9.35), we obtain an explicit expression of $E(\delta^2 \tilde{J})$. Consequently problem (9.18) is equivalent to

$$\begin{aligned} \min_{S(t), K(t)} E(\delta^2 J) &= \frac{1}{2} \left(\begin{aligned} &\text{tr}(\phi_{vv}(t_f) W_v) + \text{tr}(S(t_0) P_0) \\ &+ \text{tr} \int_{t_0}^{t_f} (\dot{S} + A_x^T S + S A_x + \tilde{H}_{xx}) P dt \\ &+ \text{tr} \int_{t_0}^{t_f} (B_n^T S B_n + B_n^T \tilde{H}_{xn} + \tilde{H}_{nn}) W_n dt \end{aligned} \right), \quad (9.36) \\ \text{s.t., } \dot{P} &= A_x P + P A_x^T + B_n W_n B_n^T, \quad t \geq t_0, \quad \text{given } P(t_0) = P_0. \end{aligned}$$

which has two decision variables, $S(t)$ and $K(t)$, to be determined. The differential equation constraint arises from the definition of $P(t)$ and the dynamic equation (9.24), which makes the above optimization difficult to solve. Fortunately, we find in Appendix C that *problem (9.36) can be solved equivalently without the constraint by taking $P(t)$ as an additional decision variable*. In other words, the equation constraint is a necessary condition for a minimum of the unconstrained problem and thus can be omitted when solving (9.36).

Based on variation theory, by requiring the increment of $E(\delta^2 J)$ be zero in the presence of variations in the decision variables, the necessary conditions for a minimum of (9.36) are obtained as

$$-\dot{S} = A_x^T S + S A_x + \tilde{H}_{xx}, \quad t \leq t_f, \quad (9.37)$$

$$\dot{P} = A_x P + P A_x^T + B_n W_n B_n^T, \quad t \geq t_0, \quad (9.38)$$

$$0 = \begin{pmatrix} 2H_{uu}KCPC^T - 2f_u^T SPC^T - 2H_{ux}PC^T + 2f_u^T Sf_u KW_v + 2H_{uu}KW_v - 2H_{uv}W_v \\ -H_{uw}W_w f_w^T C^T - f_u^T H_{xv}W_v + f_u^T H_{xu}KW_v + H_{ux}f_u KW_v \\ + f_u^T C^T K^T H_{uv}W_v + H_{uv}W_v K^T f_u^T C^T \\ -f_u^T C^T K^T H_{uu}KW_v - H_{uu}KW_v K^T f_u^T C^T - H_{uu}K C f_u KW_v \end{pmatrix}, \quad (9.39)$$

satisfying the boundary conditions

$$S(t_f) = \phi_{xx}(t_f), \quad P(t_0) = P_0. \quad (9.40)$$

Note that while the matrix $S(t)$ does not have an obvious physical meaning, the matrix $P(t)$ defines the mean covariance of the state $x(t)$. To conclude, the solution of (9.36) is solved from (9.37)-(9.40).

It is in general difficult to solve equations (9.37)-(9.40) due to the complex (9.39). The difficulty, however, degenerates significantly if $W_v = 0$ (i.e., the measurements are free of noises) which is approximately the case when the noises are filtered and minimized in applications. In the following, we restrict the derivations to this ideal case. When $W_v = 0$, the term $B_n W_n B_n^T$ in (9.38) degenerates to $f_w W_w f_w^T$; and from (9.39), K is solved as

$$K = H_{uu}^{-1}(f_u^T SP + H_{ux}P + \frac{1}{2}H_{uw}W_w f_w^T)C^T (CPC^T)^{-1}, \quad (9.41)$$

where CPC^T is assumed to be invertible.

Therefore we have to solve (9.41) together with (9.37)-(9.38) (which are differential Lyapunov equations) in order to get a solution of (9.36). Note that the three equations comprise a two-point boundary-value problem which in general remains difficult to solve.

In control practice, it is preferable to have a constant (or static) instead of time-varying feedback gain for simple implementation. An optimal static LMF gain can indeed be found for (9.36) as follows.

Let K be constant. With this additional constraint, following the steps of deriving (9.37)-(9.39) we can derive the new necessary conditions for a minimum of problem (9.36) as equations (9.37)-(9.38) in addition to

$$\int_{t_0}^{t_f} H_{uu} K C P C^T dt = \int_{t_0}^{t_f} (f_u^T S P + H_{ux} P + \frac{1}{2} H_{uw} W_w f_w^T) C^T dt. \quad (9.42)$$

The optimal static LMF gain K is solved from (9.37)-(9.38) and (9.42), which can be very difficult to solve. If the linearized process is time-invariant, (9.42) simplifies to

$$K = H_{uu}^{-1} \left(\int_{t_0}^{t_f} (f_u^T S P + H_{ux} P + \frac{1}{2} H_{uw} W_w f_w^T) C^T dt \right) \left(\int_{t_0}^{t_f} C P C^T dt \right)^{-1}. \quad (9.43)$$

However, it still requires solving a two-point boundary-value problem for a solution.

Given the linearized process being time-invariant, a case much easier to handle is when $t_f = \infty$, in which the equations (9.37)-(9.38) and (9.43) are dominated by the dynamics during the steady interval and consequently the optimal LMF gain can be solved from

$$0 = A_x^T S + S A_x + \tilde{H}_{xx}, \quad (9.44)$$

$$0 = A_x P + P A_x^T + B_n W_n B_n^T, \quad (9.45)$$

$$K = H_{uu}^{-1} (f_u^T S P + H_{ux} P + \frac{1}{2} H_{uw} W_w f_w^T) C^T (C P C^T)^{-1}, \quad (9.46)$$

which gives a constant K . These solution equations are in agreement with those derived for LQR optimal control with *static output feedback* when $W_w = 0$ and the initial states are uncertain [103]. The equations can be solved by an iterative algorithm sketched in Table 9.1, whose convergence is guaranteed if $W_w = 0$ under regular conditions [103, 107].

To conclude, the optimal time-varying perturbation control gain $K_{opt}(t)$ is solved from (9.37)-(9.38) and (9.40)-(9.41); and the optimal static perturbation control gain is solved from (9.37)-(9.38) and (9.42) (or (9.43) as a special case), or from (9.44)-(9.46) when $t_f = \infty$. The gain gives a local optimal solution to the problem defined in (9.12) by means of $u_{opt}(t) := u^*(t) - K_{opt}(t)y(t)$.

Table 9.1 Algorithm for solving a local optimal LMF gain when $W_v = 0$ and $t_f = \infty$ (The linearized process is supposed to be time-invariant.)

1. Initialize:

Set $i=0$ and $\delta^2 J_{opt}$ as a large number, e.g., 10^6

Determine a gain K_0 so that $f_x - f_u K_0 C$ is asymptotically stable, where $C = dh/dx$

2. i -th iteration:

Set $A_{x,i} = f_x - f_u K_i C$

Solve for S_i and P_i in

$$A_{x,i}^T S_i + S_i A_{x,i} + \tilde{H}_{xx} = 0,$$

$$A_{x,i} P_i + P_i A_{x,i}^T + B_n W_n B_n^T = 0,$$

Compute $\delta^2 J_i = \text{tr}(S_i P_0) + \text{tr} \int_{t_0}^{t_f} (B_n^T S_i B_n + B_n^T \tilde{H}_{xn} + \tilde{H}_{nn}) W_n$

If $\delta^2 J_i < \delta^2 J_{opt}$, then set $\delta^2 J_{opt} = \delta^2 J_i$ and $K_{opt} = K_i$

Evaluate the gain update direction

$$\Delta K = H_{uu}^{-1} \left(f_u^T S_i P_i + H_{ux} P_i + \frac{1}{2} H_{uw} W_w f_w^T \right) C^T (C P_i C^T)^{-1} - K_i$$

Update the gain by

$$K_{i+1} = K_i + \alpha \Delta K$$

where α is chosen such that $f_i - f_u K_{i+1} C$ is asymptotically stable and

$$\delta^2 J_{i+1} = \text{tr}(S_{i+1} P_0) + \text{tr} \int_{t_0}^{t_f} (B_n^T S_{i+1} B_n + B_n^T \tilde{H}_{xn} + \tilde{H}_{nn}) W_n \leq \delta^2 J_i.$$

If $\delta^2 J_{i+1}$ and $\delta^2 J_i$ are close enough to each other or i equals the maximal times of iteration, go to 3; otherwise, set $i = i + 1$ and go to 2

3. Terminate:

If i equals the maximal times of iteration, then the solution may not exist; otherwise, set

$$\delta^2 J_{opt} = 0.5 \delta^2 J_{opt}$$

Stop

Remark 9.1 Some special cases of the solution equations in (9.37)-(9.43) are discussed when the linearized process is time-invariant.

(i) If $W_w = 0$, $W_v = 0$ and $C = I$, it is easy to verify that equation (9.38) becomes redundant and (9.37) and (9.41) recover the solution equations of LQR optimal control with state feedback [103]. If further $t_f = \infty$, then the solution is a constant feedback gain

and can be solved from (9.44) and (9.46). The optimal static feedback gain solved from (9.43) and (9.37), when t_f is finite, is a new result, as far as we know.

(ii) Let the initial states of the process be given. If $W_w = 0$ and $W_v = 0$, the problem degenerates into an LQR optimal control problem with output feedback and the necessary conditions (9.37)-(9.40) determine an optimal feedback gain dependent on initial states of the process, satisfying the condition of $(KC - H_{uu}^{-1}(f_u^T S + H_{ux}))PC^T = 0$, where P is rank deficient. It would be very difficult, if not impossible, to solve an optimal K from the solution equations. As an easier case, we may assume there is an optimal gain such that $KC = H_{uu}^{-1}(f_u^T S + H_{ux})$, which removes the relevance of the initial state. Or, the relevance is eliminated if we assume P_0 is invertible, i.e., the initial states are uncertain. This implies that P be invertible and hence K can be expressed explicitly as $K = H_{uu}^{-1}(f_u^T S + H_{ux})PC^T(CPC^T)^{-1}$, making the equations be numerically solvable.

(iii) It would be N-P hard to determine whether there exist LMFs gains $K(t)$, for $t_0 \leq t \leq t_f$, stabilizing the linearized process, since a similar determination has been conjectured (with strong evidence) to be N-P hard for the simpler LQR problem with static output feedback [108-109]. Thus when applying dSOC, numerical experiments are required to check whether the optimal control law leads to a stable system or not.

Remark 9.2 If the measurements y is not only a function of x but also of u , i.e., $y = h(x, u) + v$ (compare with y given in (9.2)), the formulation of dSOC is similar. However, the dSOC problem becomes much difficult to solve in general. If $W_v = 0$, however, the problem is solvable which involves solving coupled equations similar to (9.37)-(9.38) and (9.41); the main difference is that the equation similar to (9.41) will have the variable $K(t)$ on both sides. The derivations are skipped for brevity.

9.3.2 Optimal Selection of Γ

Once the optimal perturbation control gain is solved for a given MCM Γ , the increment in the economic cost due to disturbance and noise perturbation can be estimated. Subsequently an optimal Γ is selected as the Γ resulting in minimal cost increment among the candidates. As follows, we first derive an explicit expression of the increment in economic cost.

Around the nominal optimal path, expand the economic cost $J_{0,opt}$ defined in (9.10) to second order, yielding

$$E(J_{0,opt}) \approx E(J_{0,opt}^*) + E(\delta J_{0,opt} + \delta^2 J_{0,opt}), \quad (9.47)$$

where

$$\delta J_{0,opt} = \phi_{0,x}(t_f) \delta x(t_f) + \int_{t_0}^{t_f} (F_{0,x} \delta x + F_{0,u} \delta u) dt, \quad (9.48)$$

$$\delta^2 J_{0,opt} = \frac{1}{2} \delta x^T(t_f) \phi_{0,xx}(t_f) \delta x(t_f) + \frac{1}{2} \int_{t_0}^{t_f} \begin{bmatrix} \delta x^T & \delta u^T \end{bmatrix} \begin{bmatrix} F_{0,xx} & F_{0,xu} \\ F_{0,ux} & F_{0,uu} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta u \end{bmatrix} dt, \quad (9.49)$$

which is subject to the state equation (9.20). In the equations, the symbols $\bullet_{0,*}$ and $\bullet_{0,*\#}$ denote respectively the derivatives $\partial \bullet_0 / \partial *$ and $\partial^2 \bullet_0 / \partial * \partial \#$ evaluated at the nominal trajectory, where $\bullet_0$ denotes the function. Given the nominal optimal cost $E(J_{0,opt}^*)$, we can estimate the economic cost increment as $E(J_{0,opt}) - E(J_{0,opt}^*)$. In order to select a MCM resulting in minimal cost increment, it is essential to obtain $E(\delta J_{0,opt} + \delta^2 J_{0,opt})$ for each candidate MCM.

With the explicit solution of $\delta x(t)$ in (9.26) and the assumption of zero means of variations in initial states, disturbances and noises, it is easy to deduce that $E(\delta x) = 0$ and $E(\delta u) = 0$. Consequently $E(\delta J_{0,opt}) = 0$ and the cost increment can be estimated as $E(\delta^2 J_{0,opt})$. Specifically, the expression of $E(\delta^2 J_{0,opt})$ in (9.47) can be rewritten as

$$\begin{aligned} \mathbb{E}(\delta^2 J_{0,opt}) &= \frac{1}{2} \mathbb{E} \left(\delta x^T(t_f) \phi_{0,xx}(t_f) \delta x(t_f) \right) \\ &\quad + \frac{1}{2} \mathbb{E} \left(\int_{t_0}^{t_f} \begin{bmatrix} \delta x^T & dv^T \end{bmatrix} \begin{bmatrix} \tilde{F}_{0,xx} & \tilde{F}_{0,xv} \\ \tilde{F}_{0,vx} & \tilde{F}_{0,vv} \end{bmatrix} \begin{bmatrix} \delta x \\ dv \end{bmatrix} dt \right), \end{aligned} \quad (9.50)$$

where

$$\begin{aligned} \tilde{F}_{0,xx} &:= \begin{bmatrix} I & -C^T K^T \end{bmatrix} \begin{bmatrix} F_{0,xx} & F_{0,xu} \\ F_{0,ux} & F_{0,uu} \end{bmatrix} \begin{bmatrix} I \\ -KC \end{bmatrix}, \\ \tilde{F}_{0,vx}^T = \tilde{F}_{0,xv} &:= \begin{bmatrix} I & -C^T K^T \end{bmatrix} \begin{bmatrix} -F_{0,xu} K \\ -F_{0,uu} K \end{bmatrix}, \\ \tilde{F}_{0,vv} &:= K^T F_{0,uu} K. \end{aligned} \quad (9.51)$$

We continue to find an explicit expression of $\mathbb{E}(\delta^2 J_{0,opt})$. The first part of $\mathbb{E}(\delta^2 J_{0,opt})$ is obtained as

$$\frac{1}{2} \mathbb{E} \left(\delta x^T(t_f) \phi_{0,xx}(t_f) \delta x(t_f) \right) = \frac{1}{2} \text{tr} \left(\phi_{0,xx}(t_f) P(t_f) \right), \quad (9.52)$$

where $P(t_f)$ is solved from (9.38).

In order to compute the integral part of $\mathbb{E}(\delta^2 J_{0,opt})$, the cross-correlation matrix $\Omega_{vx}(t, t)$ has to be determined. Specifically we have

$$\Omega_{vx}(t, t) = \mathbb{E} \left(dv(t) \delta x^T(t) \right) = \int_{t_0}^t \mathbb{E} \left(dv(t) dn^T(\tau) \right) B_n^T(\tau) \Phi^T(t, \tau) d\tau = -\frac{1}{2} W_v K^T f_u^T. \quad (9.53)$$

Consequently we obtain

$$\begin{aligned} &\frac{1}{2} \mathbb{E} \left(\int_{t_0}^{t_f} \begin{bmatrix} \delta x^T & dv^T \end{bmatrix} \begin{bmatrix} \tilde{F}_{0,xx} & \tilde{F}_{0,xv} \\ \tilde{F}_{0,vx} & \tilde{F}_{0,vv} \end{bmatrix} \begin{bmatrix} \delta x \\ dv \end{bmatrix} dt \right) \\ &= \frac{1}{2} \text{tr} \int_{t_0}^{t_f} \left(\tilde{F}_{0,xx} P - \tilde{F}_{0,xv} W_v K^T f_u^T + \tilde{F}_{0,vv} W_v \right) dt. \end{aligned} \quad (9.54)$$

where $P(t)$ is solved from (9.38). Adding up (9.54) and (9.52) for each side yields an estimate of the economic cost increment:

$$\mathbb{E}(\delta^2 J_{0,opt}) = \frac{1}{2} \text{tr} \left(\phi_{0,xx}(t_f) P(t_f) \right) + \frac{1}{2} \text{tr} \int_{t_0}^{t_f} \left(\tilde{F}_{0,xx} P - \tilde{F}_{0,xv} W_v K^T f_u^T + \tilde{F}_{0,vv} W_v \right) dt. \quad (9.55)$$

Note that if $W_v = 0$, the two terms with W_v vanish in (9.55) and the integrand contains a single term $\tilde{F}_{0,xx}P$. Since P is solved from (9.38), which is dependent on the disturbance variance matrix, the economic cost increment is still dependent on the variance of the disturbance. In addition, the cost increment remains to be estimated from (9.55) if the static LMF gain is solved from (9.37)-(9.38) and (9.42).

For each candidate MCM Γ , the increment in economic cost is computed from (9.55) under the optimal perturbation control derived in the last subsection. The optimal MCM to determine the CVs is then solved from (9.10), which gives minimal economic cost among all candidate MCMs.

Remark 9.3 (i) *If the perturbation control with linear CV feedback is adopted, i.e., $\delta u(t) = -K(t)\delta z(t)$, where $\delta z(t) := z(t) - z_r(t) = \Gamma\delta y(t)$ and $K(t)$ is the CV feedback gain, then the formulation of dSOC remains by replacing K with $K\Gamma$ in all places. The solution can similarly be obtained. Compared to measurement feedback in (9.17), CV feedback enforces a measurement feedback gain in a constrained form as $K\Gamma$ and hence leads to larger cost in general.*

(ii) *Given a candidate MCM, the perturbation control law with linear CV or measurement feedback is feasible if and only if it admits a feedback gain to stabilize the linearized process. It is in general too difficult to know the feasibility a priori, either analytically or numerically, as mentioned in Remark 9.1. Therefore we have to test it through numerical experiments: if the optimal perturbation control law leads to an unstable closed-loop system, then the control law is said to be infeasible; and feasible, otherwise. In this work, we make a convention that a given MCM (or CV vector) is deemed as invalid if the resulting optimal perturbation control is infeasible.*

9.4 Numerical Example

Let us consider a linear time-invariant process with quadratic cost. Let the functions and equations be of the form: $\dot{x} = Ax + B_u u + B_w w$, $y = Cx$, $z = \Gamma y$, $u = -K(y - y_r)$,

$J_0 = 0.5 \int_{t_0}^{t_f} (x^T Q x + u^T R u) dt$ and $J'_0 = 0.5 \int_{t_0}^{t_f} (z - z_r)^T M (z - z_r) dt$, where the matrices are

$$A = \begin{bmatrix} -3 & 3 & 3 \\ 0 & -2 & 2 \\ 0 & 0 & -0.5 \end{bmatrix}, B_u = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, B_w = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad (9.56)$$

$$C = I_3, Q = I_3, R = I_2, M = I_2 / \rho.$$

In (9.56), ρ is a positive scalar controlling the tradeoff between the economic cost J_0 and tracking cost J'_0 . Assume that $w \sim N(0, \sigma)$ and $x(t_0) \sim N(0, I_3)$ which are Gaussian white noises. Let the nominal disturbance and optimal (in an approximate sense) state trajectories both be constantly zero. Thus we have $y_r = 0$ and $z_r = 0$. The above system describes the perturbation system and $u(t)$ is the perturbation control input.

Consider three candidate MCMs:

$$\Gamma_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \Gamma_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \Gamma_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (9.57)$$

each of which leads to a CV vector with a size of 2×1 , the same size of the control input. For each candidate Γ , two sets of numerical studies are carried out: (a) σ is set to 1.0 and ρ varies from 1.0 to 10 at a step of 0.5, and (b) ρ is set to 1.0 and σ varies from 0 to 1.0 at a step of 0.1. These two sets of studies investigate the impacts of cost weighting and disturbance strength on the selection of CVs, respectively. The initial and terminal times are $t_0 = 0$ and $t_f = 20$, respectively. The time interval is sufficiently long (relative to the settling time of the closed-loop response) such that optimal static feedback gains are solved by taking t_f as infinity, approximately. So the feedback gains are solved from (9.44)-(9.46) and then applied to compute the costs.

As to the study (a), the numerical results are obtained and shown in Figure 9.1.(a). Two main observations are that: (i) the economic cost increment ($E(\delta^2 J_0)$) decreases as the weighting factor (ρ) increases, and (ii) the economic cost increment associated with the three candidate MCMs increases in the order of Γ_3 , Γ_2 , and Γ_1 . Observation (i) can be interpreted as follows: As ρ increases, the weighting ($1/\rho$) on the tracking cost becomes lighter and thus allows looser CV tracking performance; equivalently, this means heavier weighting on the economic cost and consequently leads to enhanced process performance with smaller economic cost increment arising from disturbance. Observation (i) confirms the theoretical tradeoff between minimizing the tracking errors of CVs and minimizing the economic cost of a process. Observation (ii) indicates that the economic cost increment associated with Γ_3 is the smallest over all values of ρ tested, as compared to the cost increments associated with Γ_1 and Γ_2 . This implies that, among the three candidate MCMs, Γ_3 is the best, and consequently the CVs should be determined as $\Gamma_3 y$.

As to the study (b), the results are shown in Figure 9.1.(b). The results indicate that the economic cost increment strictly increases as the disturbance covariance enlarges. When there is no disturbance (i.e., $\sigma = 0$), the economic cost increments associated with Γ_3 , Γ_1 and Γ_2 increase in order. This order, however, is soon changed as the strength of disturbance increases. Once disturbance appears (i.e., $\sigma > 0$), the cost increment associated with Γ_3 almost keeps smaller than those with Γ_2 and Γ_1 . The results again support the preceding conclusion that Γ_3 is the best MCM among the three candidates as used to determine the CVs.

When the perturbation control is changed with linear CV feedback, the two sets of studies (a) and (b) are carried out and the results are shown in Figure 9.2. The results turn out to be similar to those with LMF control, but the gaps of performance associated with

the three candidate MCMs become larger in each set of study. The combination matrix Γ_3 keeps superior to the other two candidates, giving smallest economic cost increments in all the cases tested. This strongly recommends selecting Γ_3 instead of Γ_1 and Γ_2 to determine the CVs, in agreement with the conclusion obtained under LMF control.

Figure 9.1 Economic cost increment ($E(\delta^2 J_0)$) as functions of the weighting factor (ρ) and the disturbance covariance (σ), under optimal LMF perturbation control.

Figure 9.2 Economic cost increment ($E(\delta^2 J_0)$) as functions of the weighting factor (ρ) and the disturbance covariance (σ), under optimal perturbation control with different CV feedbacks.

Figure 9.3 LMF control v.s. classic LQG control.

In addition, to validate the proposed LMF control, we compare the cost increments with those resulting from classic LQG control (which uses dynamic instead of LMF) for different values of the weighting factor (ρ) and the disturbance variance (σ). Since an infinite time horizon and zero measurement noise are assumed in the example, the LQG control with infinitesimal measurement noise should coincide with the LMF control. The numerical results are shown in Figure 9.3, indicating that the two controls do achieve the same performance (the slight gaps are due to numerical errors). This validates the proposed LMF control scheme in this special case.

The above costs computed by the analytical formulas have been confirmed by Monte Carlo simulations, which compute the average economic and total costs over a large number of scenarios of the step input response of the control system subject to various realizations of the disturbances. The results are omitted for brevity.

9.5 Conclusions

A theoretical formulation of dSOC was presented and a local solution of the optimal MCM was obtained by solving three coupled equations, provided that an LMF control law is applied and that a set of candidate MCMs are given. The application of dSOC to select CVs for a linear time-invariant process illustrated the usefulness of the theoretical results.

Since the solution equations of dSOC comprise a two-point boundary-value problem which is in general very difficult to solve. Future work is needed to develop efficient algorithms to solve these equations in a general case, and to test the theoretical results with nonlinear processes.

Chapter 10

Summary and Future Work

10.1 Summary

Some new results on PID controller tuning (Chapters 3-6) and SOC design (Chapters 7-9) have been obtained, which are briefly summarized as follows.

Chapter 3 gave an *almost* closed-form solution of the PI/PD/PID parameters satisfying specified GPMs for an IPTD process and derived explicit expressions for estimating the GPMs attained by a given PI/PD/PID controller. The results unify a large number of tuning rules into the same framework of tuning PI/PD/PID controllers based on GPM specifications; and the GPMs attained by available tuning rules were computed and documented for engineers as reference in the future design.

Chapter 4 derived simple PID tuning rules in analog to the SIMC rules based on results in Chapter 3. Compared to SIMC rules that use a first-order Taylor expansion of the time delay component of a process, the new rules adopt a second-order Taylor expansion and hence endows more accurate design to follow the performance specifications. Simulations showed that the new rules lead to improved disturbance rejection while achieving the same peak sensitivities compared to the SIMC counterparts.

Chapter 5 proposed systematic approaches to carrying out 2DOF-DS for designing PID and PID-C controllers, respectively, which lead to explicit PID and PID-C tuning rules for typical process models. Although the new rules have complicated forms, simulations

showed that they can achieve very good performance for a wide range of processes and are advantageous over recent rules in many cases.

Chapter 6 analytically derived a PI tuning rule with the CSR method. The rule requires only the measurements of the peak time, steady-state offset, and overshoot or rise time in a CSR experiment, needing no explicit model of a process at all. The tuning rule is simple to use and has been demonstrated to be very efficient for a wide range of processes. Meanwhile, the analysis provides a kind of analytical support to the PI tuning rule reported in [11] which is derived from extensive numerical experiments.

Chapter 7 reported some new results on the local solutions for SOC. More complete characterizations of the solutions were obtained for SOC to minimize worst-case and average losses, respectively. The results reveal that the available solution for SOC to minimize average loss is complete. This insight contributes to a clearer characterization of the relation between the solutions for SOC to minimize these two kinds of losses.

Chapter 8 dealt with SOC design of constrained processes. It is proposed to treat the problem as the available SOC subject to process constraints. The problem is convex and can be solved efficiently. Compared to existing approaches for the same problem, the proposed approach has a unique advantage of retaining the feature of simplicity of SOC for near-optimal operation.

Chapter 9 formulated the problem of dSOC and obtained a local solution for it by adopting perturbation control approach. It is found that the solution is essentially associated with an optimal perturbation control. By assuming that the perturbation control is in the form of LMF and that a set of candidate CVs are available, a way of selecting the optimal CVs that minimize the economic loss was presented. The application of dSOC to a linear process illustrated the usefulness of the theoretical results.

10.2 Future Work

10.2.1 On PID Controller Tuning

The PID tuning rules developed in Chapter 4 are equivalent to the SIMC rules if the processes are delay-dominated. This implies that the new rules would lead to the same performance as SIMC counterparts in these cases, which, however, can be far from being optimal. This can be seen from the derivation that PI control of an FOPTD process is basically reduced to P control of a pure TD process, which implies limited performance that can be resulted in. The observation is in agreement with the results of most recent studies on SIMC rules [110]. All in all, there is still space to improve the new rules for PID control of delay-dominated processes. Additionally, as mentioned in the conclusion part of Chapter 4, how to appropriately set the D parameter for PID control of a DIPTD process also requires further study.

The analysis in Chapter 6 indicates that a PI tuning rule with no process model (which is model-free in a sense) can be obtained by implicitly identifying the process parameters in terms of the CSR parameters, namely the peak time, the steady-state offset, the overshoot and/or the rise time. Once a good model-based PI tuning rule is obtained, a comparable CSR PI tuning rule is just in hand, according to the derivation in Chapter 6. Then a question naturally arises: what kind of model-based PI tuning rule will lead to a best CSR PI tuning rule? This question should be clarified in future studies.

Another basic question with the derived CSR tuning rules, including the one developed in Chapter 6 and the one reported in [11], is why the rules can be applicable to a wide range of processes while they are derived based on either an IPTD or an FOPTD process model. The ‘magic’ ability should have certain theoretical explanations, at least in the sense of certain approximations.

Additionally, research is demanded on tuning PID controllers for high-order processes. In practice, a process is usually of a high-order model in nature. If it can well be approximated by an integral or first/second-order model, then the PID tuning methods in the thesis and others in the literature may be applied. Otherwise, advanced tuning methods are required. So far there have been few such tuning methods for high-order processes.

Moreover, while the thesis and major literature have concentrated on tuning PID controllers based on frequency-domain analysis, the time response and its performance measures are ultimate goals of design and applications of a PID control system. Therefore research on PID controller tuning based on time-domain analysis is much desirable and needs more investigation.

10.2.2 On SOC Design

As mentioned in the conclusion part of Chapter 9, an efficient algorithm is demanded for solving the two-point boundary value problem consisting of two coupled differential equations and one nonlinear algebraic equation. And also, the theory of dSOC has to be tested with nonlinear processes to validate its value in practice. Some new questions and developments are possible once the work is finished.

On the other hand, it should be noted that all the studies on SOC in Chapters 7-9 assume no special structural constraints on the MCMs, where ‘special’ means that some of the CVs must be resulted from different sets of measurements. The special structural constraints, however, may occur in practice. For example, if the measurements are distributed and far from each other in space, each CV may be required to be expressed as linear combinations of the local measurements in order to reduce the implementation cost. This naturally results in a structural constraint on the MCM that certain elements of the matrix must be zero. SOC with structural constraints on MCM has aroused attention

recently [45, 54] and should be investigated further to obtain a general solution which can be solved efficiently.

In addition, future research is needed to relax the assumption that the CVs are linear combinations of measurements. Some progress has been made in this direction, referring to [111] which allows the CVs to be polynomials of measurements. The current results, however, assume zero measurement noise. The way to solve such an SOC problem with measurement noise is still in exploration. And in the most general case, we need to solve the SOC problem when CVs are selected as per the optimality conditions of the optimization problem without setpoint constraints. The solution to this exact problem or its approximation also requires future investigation.

Appendices

A Approximate Analytical Solutions of β for (3.11) and (3.34)

To solve (3.11) and (3.34) for approximate solutions, first consider approximating the following equation.

$$x = \tan y, \quad y \in (-\pi/2, \pi/2). \quad (\text{A.1})$$

Divide the domain of y into two parts:

$$\begin{aligned} \mathcal{D}_1 &:= (-\arctan x_b, \arctan x_b), \text{ and} \\ \mathcal{D}_2 &:= (-\pi/2, -\arctan x_b] \cup [\arctan x_b, \pi/2), \end{aligned} \quad (\text{A.2})$$

where $x_b \geq 1$ is a boundary value. Since (A.1) has odd solutions, it is sufficient to consider solving it in the domain consisting of $\mathcal{D}_1^r := [0, \arctan x_b)$ and $\mathcal{D}_2^r := [\arctan x_b, \pi/2)$.

In $\mathcal{D}_1^r$, approximate (A.1) by the Taylor expansion of $\tan y$ to the fifth order, giving

$$x = \tan y \approx y + y^3/3 + 2y^5/15, \quad (\text{A.3})$$

of which the relative approximation error is

$$e_1(y) := (y + y^3/3 + 2y^5/15) / \tan y - 1. \quad (\text{A.4})$$

In $\mathcal{D}_2^r$, first convert (A.1) into the arctangent form and then approximate it by

$$y = \arctan x = \pi/2 - \arctan z \approx \pi/2 - \lambda_b z, \quad (\text{A.5})$$

where $z := x^{-1}$ and $\lambda_b := \lambda(1/x_b)$ and $\lambda(\bullet)$ is a function defined as

$$\lambda(t) := (\arctan t)/t, \quad t \in (0, +\infty). \quad (\text{A.6})$$

The corresponding relative approximation error is

$$e_2(z) := \tan(\pi/2 - \lambda_b z) / z^{-1} - 1 = z / \tan(\lambda_b z) - 1. \quad (\text{A.7})$$

Note that *i*) to be consistent with $e_1(y)$, the tangents of both sides of (A.5) are taken to calculate $e_2(z)$; and *ii*) the Taylor expansion is not used in $\mathcal{D}_2^r$ since it is hard to attain high accuracy; and *iii*) in $\mathcal{D}_2^r$ it has $z \in (0, x_b^{-1}]$.

From (A.4) and (A.7), it can be easily proved that $e_1(y) < 0$, $de_1(y)/dy \leq 0$, $e_2(z) > 0$ and $de_2(z)/dz \leq 0$. Thus the maximum absolute values of $e_1(y)$ and $e_2(y)$ are respectively

$$\begin{cases} \|e_1(y)\|_\infty = -e_1(\arctan x_b), \text{ and,} \\ \|e_2(z)\|_\infty = \lim_{z \rightarrow 0} e_2(z) = 1/\lambda - 1. \end{cases} \quad (\text{A.8})$$

Here $\|e_1(y)\|_\infty$ and $\|e_2(z)\|_\infty$ are both functions of x_b , as shown in Figure A.1, where the intersection point is numerically obtained as $x_B := x_b \approx 1.848$. At this point, the maximum absolute values of the relative errors by the two different approximations equal each other at 9.10%, and $\lambda_B := \lambda = \lambda(1/x_B) \approx 0.917$.

Figure A.1 The maximal absolute values of the relative errors of the approximate solutions, as functions of the boundary point x_b .

For y being an explicit function of x , e.g., $y=2x$, by taking x_B and λ_B as the boundary parameters for the above two approximations, an approximate solution of (A.1) can be obtained by solving either (A.3) or (A.5) for x .

In addition, notice that in some cases where y is an explicit function of x , (A.3) may prevent an analytic solution of x . As a compromised solution, a lower order Taylor expansion of $\tan y$ may be adopted. Consider the third-order Taylor expansion case where (A.3) and (A.4) are replaced respectively by

$$x = \tan y \approx y + y^3/3, \text{ and} \quad (\text{A.9})$$

$$e_1(y) = (y + y^3/3) / \tan y - 1. \quad (\text{A.10})$$

Keep (A.5) unchanged. By deducting similarly as above, the approximation boundaries are obtained as $x_B \approx 1.500$ and $\lambda_B \approx 0.882$, at which the maximum absolute values of the relative errors by the two different approximations equal each other at 13.38%.

A.1 An Approximate Solution of (3.11)

In particular, let $x := \beta > 0$ and $y := \theta\beta > 0$ in (A.1). From (A.3) and (A.5), an approximate solution of (A.1) can be obtained as follows:

$$\begin{cases} \beta = \frac{1}{2\theta} \sqrt{-5 + \sqrt{\frac{120}{\theta} - 95}}, & 0 < \beta < \beta_B, \\ \beta = \frac{\pi}{4\theta} \left(1 + \sqrt{1 - \frac{16\lambda_B\theta}{\pi^2}} \right), & \beta \geq \beta_B, \end{cases} \quad (\text{A.11})$$

where $\lambda_B = 0.917$ and $\beta_B = 1.848$. Alternatively, by specifying the conditions of θ , the solution (A.11) can be re-expressed as

$$\begin{cases} \beta = \frac{\pi}{4\theta} \left(1 + \sqrt{1 - \frac{16\lambda_B\theta}{\pi^2}} \right), & \text{if } 0 < \theta \leq \theta_B, \\ \beta = \frac{1}{2\theta} \sqrt{-5 + \sqrt{\frac{120}{\theta} - 95}}, & \text{if } \theta_B < \theta < 1, \end{cases} \quad (\text{A.12})$$

where

$$\theta_B := \min \left\{ \frac{\pi^2}{16\lambda_B}, \frac{1}{\beta_B} \left(\frac{\pi}{2} - \frac{\lambda_B}{\beta_B} \right) \right\} \approx 0.582. \quad (\text{A.13})$$

Note that for (A.11), as the boundaries of the applying regions of θ do not coincide, for simplicity θ_B is taken as the one calculated from the second equation of (A.11). The validity of the approximate solution of (A.1) by (A.12) is demonstrated by the exemplary results shown in Figure 3.2.

A.2 An approximate solution of (3.34).

To solve (3.34), two different cases are considered separately as follows (The point $\beta = 1/\sqrt{k}$ is undefined in the equations and is therefore omitted.):

$$\arctan \frac{\beta}{1 - k\beta^2} = \theta\beta, \text{ if } 1 - k\beta^2 > 0; \quad (\text{A.14})$$

$$\arctan \frac{\beta}{1 - k\beta^2} = \theta\beta - \pi, \text{ if } 1 - k\beta^2 < 0. \quad (\text{A.15})$$

Let $x := \beta/(1 - k\beta^2)$ and $y := \theta\beta$ in (A.1). From (A.5) and (A.9) an approximate solution of (A.14) is derived as

$$\begin{cases} \beta = \sqrt{\frac{1}{2} \left[\frac{1}{k} - \frac{3}{\theta^2} + \sqrt{\left(\frac{1}{k} + \frac{3}{\theta^2} \right)^2 - \frac{12}{k\theta^3}} \right]}, & \text{if } 0 < \beta < \beta_B, \\ \beta = \frac{\pi}{4(\theta - \lambda_B k)} \left(1 + \sqrt{1 - \frac{16\lambda_B(\theta - \lambda_B k)}{\pi^2}} \right), & \text{if } \beta_B \leq \beta < 1/\sqrt{k}, \end{cases} \quad (\text{A.16})$$

where

$$\lambda_B := \lambda(1/x_B), \beta_B := (\sqrt{1+4kx_B^2} - 1)/(2kx_B), \quad (\text{A.17})$$

with $x_B := 1.5$ and $\lambda(\bullet)$ being defined in (A.6).

To solve (A.15), the approximation skills used in (3.22)-(3.23) are adopted. Specifically, by applying the skill used in (3.23), an approximate solution of (A.15) is obtained as

$$\beta = \frac{\pi}{4(\theta - \lambda'_B k)} \left(1 + \sqrt{1 - \frac{16\lambda'_B(\theta - \lambda'_B k)}{\pi^2}} \right), \quad \text{if } 1/\sqrt{k} < \beta \leq \beta'_B, \quad (\text{A.18})$$

where

$$\lambda'_B := \lambda(1/x'_B), \beta'_B := (\sqrt{1+4kx_B'^2} - 1)/(2kx'_B), \quad (\text{A.19})$$

with $x'_B := 1.0$ and $\lambda(\bullet)$ being defined in (A.6). And for the case where $\beta > \beta'_B$, by applying the skill used in (3.22) the following equation of β is obtained:

$$\beta^3 + a_2\beta^2 + a_1\beta + a_0 = 0, \quad (\text{A.20})$$

where

$$a_2 := -\pi/\theta, \quad a_1 := (\lambda'_B - \theta)/(k\theta), \quad a_0 := \pi/(k\theta). \quad (\text{A.21})$$

Equation (A.20) is a standard cubic equation with real coefficients, and its feasible solution (being real and positive) is obtained as

$$\beta = -a_2/3 + S + T, \quad (\text{A.22})$$

where

$$S := \sqrt[3]{R + \sqrt{D}}, \quad T := \sqrt[3]{R - \sqrt{D}}, \quad (\text{A.23})$$

with

$$\begin{aligned} D &:= Q^3 + R^2, \quad Q := (3a_1 - a_2^2)/9, \\ R &:= (9a_2a_1 - 27a_0 - 2a_2^3)/54. \end{aligned} \quad (\text{A.24})$$

Since D in (A.23) may be negative, leading to complex numbers in the calculations which should be avoided in applications, the solution (A.22) is expressed in an alternative way that

$$\beta = -a_2/3 + U, \quad (\text{A.25})$$

where

$$\begin{cases} U := \sqrt[3]{R + \sqrt{D}} + \sqrt[3]{R - \sqrt{D}}, & \text{if } D \geq 0; \\ U := 2\sqrt[6]{R^2 - D} \cos(\varphi/3), \\ \quad \text{with } \varphi := \arctan(\sqrt{-D}/R) + \mathcal{H}(R)\pi, & \text{if } D < 0. \end{cases} \quad (\text{A.26})$$

Here $\mathcal{H}(\bullet)$ is the function defined in (3.29), and D and R keep the same as those in (A.24).

With (A.16), (A.18) and (A.25), the approximate solution of (3.34) is thus obtained as follows

$$\beta = \begin{cases} \sqrt{\frac{1}{2} \left[\frac{1}{k} - \frac{3}{\theta^2} + \sqrt{\left(\frac{1}{k} + \frac{3}{\theta^2} \right)^2 - \frac{12}{k\theta^3}} \right]}, & \text{if } \beta < \beta_B; \\ \frac{\pi}{4(\theta - \lambda_B k)} \left(1 + \sqrt{1 - \frac{16\lambda_B(\theta - \lambda_B k)}{\pi^2}} \right), & \text{if } \beta_B \leq \beta < 1/\sqrt{k}; \\ \frac{\pi}{4(\theta - \lambda'_B k)} \left(1 + \sqrt{1 - \frac{16\lambda'_B(\theta - \lambda'_B k)}{\pi^2}} \right), & \text{if } 1/\sqrt{k} < \beta \leq \beta'_B; \\ -a_2/3 + U, & \text{if } \beta > \beta'_B, \end{cases} \quad (\text{A.27})$$

where the intermediate variables, λ_B and β_B , λ'_B and β'_B , a_2 and U , are defined in (A.17), (A.19), and $\{(A.21), (A.24), (A.26)\}$, respectively. Since it is hard to give the piecewise conditions of (A.27) in terms of θ like that in (A.12), the candidate solutions are calculated in a top-down sequence until a feasible β is obtained; if no feasible solution is achieved, (3.34) will be taken as having no solution, or a numerical solution to it has to be tried.

Additionally, another simpler yet less accurate approximate solution for (3.34) can be derived. The main idea is as follows. For the case of (A.14) and the case of (A.15) with $1/\sqrt{k} < \beta \leq \beta'_B$ (Here β'_B is of a different value from that in (A.27).), the approximate solutions remain the same as those in (A.16) and (A.18) respectively; and for the case of (A.15) with $\beta > \beta'_B$, first (A.15) is approximated by replacing “ $1 - k\beta^2$ ” with $-k\beta^2$ (requiring that $k\beta_B'^2 \gg 1$ — here $k\beta_B'^2 = 10$ is used, by selecting a proper boundary point x'_B). Then by applying the same skill as that in (3.22), a less accurate yet simpler approximate solution of (3.34) can be obtained. Specifically, it is as follows:

$$\beta = \begin{cases} \sqrt{\frac{1}{2} \left[\frac{1}{k} - \frac{3}{\theta^2} + \sqrt{\left(\frac{1}{k} + \frac{3}{\theta^2} \right)^2 - \frac{12}{k\theta^3}} \right]}, & \text{if } \beta < \beta_B; \\ \frac{\pi}{4(\theta - \lambda_B k)} \left(1 + \sqrt{1 - \frac{16\lambda_B(\theta - \lambda_B k)}{\pi^2}} \right), & \text{if } \beta_B \leq \beta < 1/\sqrt{k}; \\ \frac{\pi}{4(\theta - \lambda'_B k)} \left(1 + \sqrt{1 - \frac{16\lambda'_B(\theta - \lambda'_B k)}{\pi^2}} \right), & \text{if } 1/\sqrt{k} < \beta \leq \beta'_B; \\ \frac{\pi}{2\theta} \left(1 + \sqrt{1 - \frac{4\lambda''_B \theta}{k\pi^2}} \right), & \text{if } \beta > \beta'_B, \end{cases} \quad (\text{A.28})$$

where $\lambda_B := \lambda(1/x_B)$, $\lambda'_B := \lambda(1/x'_B)$, $\lambda''_B := \lambda(x'_B)$, $\beta_B := (\sqrt{1 + 4kx_B^2} - 1)/(2kx_B)$ and $\beta'_B := \sqrt{10/k}$, with $x_B := 1.5$, $x'_B := \beta'_B/(k\beta_B'^2 - 1)$ and $\lambda(\bullet)$ being defined in (A.6). As expected, the estimated β may not be accurate when $\beta > 1/\sqrt{k}$, but it is found to be able to achieve the final goal of estimating the gain margin A_m with satisfactory accuracy. The relative estimation errors are mostly within 7%. Exemplary results are shown in Figure A.2.

Figure A.2 Typical relative estimation errors of β and A_m , with β being estimated by (A.28).

B Selecting a Proper Damping Ratio ζ

To avoid the difficulty of tuning the PID parameters by ζ and k_1 , ζ may be set as a proper constant. Specifically, ζ is selected as 1.0 based on time-domain performance analysis of the approximate closed-loop system described in (4.7). In the analysis, $0 < \zeta \leq 1$ is assumed as is required for efficient response in engineering [77].

Consider the unit step input response of the closed-loop system in (4.7). The response is obtained as

$$\begin{aligned}
 Y(s) &= \frac{1}{s} \frac{\frac{-k_2 + 0.5}{(1/k_1 - 1)k_2 + 0.5} s^2 + \frac{k_2 - 1}{(1/k_1 - 1)k_2\theta + 0.5\theta} s + \frac{1}{(1/k_1 - 1)k_2\theta^2 + 0.5\theta^2}}{s^2 + \frac{k_2 - 1}{(1/k_1 - 1)k_2\theta + 0.5\theta} s + \frac{1}{(1/k_1 - 1)k_2\theta^2 + 0.5\theta^2}} \\
 &= \frac{a}{s} + \frac{b(s + \zeta\omega_n) + c\omega_n\sqrt{1 - \zeta^2}}{s^2 + 2\zeta\omega_n s + \omega_n^2},
 \end{aligned} \tag{B.1}$$

where the parameters ζ and ω_n are the same as those in (4.9), and a , b and c are given by

$$a = 1, \quad b = -\frac{k_2}{(1-k_1)k_2 + 0.5k_1}, \quad c = -\frac{b\zeta}{\sqrt{1-\zeta^2}}. \quad (\text{B.2})$$

Assume the initial states of the system and their derivatives are zero. By inverse Laplace transforms, (B.1) leads to the time-domain response as follows

$$y(t) = a + be^{-\sigma t} \cos \omega_d t + ce^{-\sigma t} \sin \omega_d t, \quad (\text{B.3})$$

where $\sigma := \zeta\omega_n$ and $\omega_d := \omega_n\sqrt{1-\zeta^2}$. From (B.3), the time-domain performance indices like the rise time t_r , the peak time t_p , and the overshoot M_p can all be calculated. Let the rise time be defined as the time for $y(t)$ reaching the steady-state value of one for the first time. This means

$$y(t_r) = 1 = a + be^{-\sigma t_r} \cos \omega_d t_r + ce^{-\sigma t_r} \sin \omega_d t_r. \quad (\text{B.4})$$

Equation (B.4) solves t_r as

$$t_r = \left(\tan^{-1}(\omega_d / \sigma) \right) / \omega_d. \quad (\text{B.5})$$

With $dy(t)/dt|_{t=t_p} = 0$, the peak time t_p is solved as

$$t_p = \begin{cases} \beta / \omega_d, & \text{if } \zeta \geq \sqrt{2}/2; \\ (\pi + \beta) / \omega_d, & \text{otherwise,} \end{cases} \quad (\text{B.6})$$

where $\beta := \tan^{-1} \left(2\zeta\sqrt{1-\zeta^2} / (2\zeta^2 - 1) \right)$. Consequently, the overshoot (which is defined as the maximum instantaneous amount by which the step response exceeds its final value and is expressed as a percentage of the final value) M_p is calculated as

$$M_p = \left(y(t_p) - 1 \right) \times 100\% = -be^{-\sigma t_p} \times 100\%. \quad (\text{B.7})$$

It is easy to see that t_r , t_p and M_p are all functions of ζ and ω_n . Since ω_n is a function of ζ and k_1 (refer to Eqs. (4.9) and (4.10)), it means that t_r , t_p and M_p are essentially functions of ζ and k_1 . Hence, the relations between these three performance indices and the two parameters ζ and k_1 can be observed by plotting out their relation numerically, as shown in Figure B.2 (For the case where $\zeta = 1$, limits are taken to obtain the index values.). Note that t_r/θ , t_p/θ and M_p are functions of ζ and k_1 independent of θ .

Figure B.2 indicates that both t_r/θ and t_p/θ are decreasing in k_1 and are increasing in ζ if $k_1 < 0.75$ (which is roughly the dividing point) while decreasing in ζ if $k_1 > 0.75$. The results also indicate that M_p is increasing in k_1 and decreasing in ζ at a smaller rate as ζ increases. Regarding all the three observed quantities, the impacts of ζ are less obvious as compared to those of k_1 . These observations mean that k_1 can control the system performance with a higher sensitivity than what ζ does. Therefore, it is suitable to set ζ as a constant while leaving k_1 as the only tuning parameter. Since t_r/θ and t_p/θ are not sensitive to changes of ζ , ζ can be set as 1.0 in order to achieve as small an overshoot as possible. Moreover, the results in Figure B.2 indicate that it can be sufficient to tune k_1 in the range of $[0.2, 0.6]$ (out of which the response is either too sluggish or too aggressive) and a k_1 around 0.5 can be a good choice for a satisfactory tradeoff between the performance indices.

Figure B.2 The achieved time-domain indices of system described in (4.7) as the tuning parameters ζ and k_1 change. The bold red curves correspond to $\zeta = 1.0$.

C Deriving the Necessary Conditions for a Minimum of (9.36)

The derivation extends that in pp. 133 of Chapter 3 of the book [103] dealing with variations of vectors to a new one dealing with variations of matrices. Define

$$\Pi = \text{tr}((A_x^T S + S A_x + \tilde{H}_{xx})P) + \text{tr}((B_n^T S B_n + B_n^T \tilde{H}_{xn} + \tilde{H}_{nn})W_n). \quad (\text{C.1})$$

Problem (9.36) can be rewritten as

$$\min_{S(t), P(t), K(t)} E(\delta^2 J) = \min_{S(t), P(t), K(t)} \left(\frac{1}{2} \text{tr}(S(t_0)P_0) + \frac{1}{2} \int_{t_0}^{t_f} (\Pi + \text{tr}(\dot{S}P)) dt \right). \quad (\text{C.2})$$

Using Leibniz's rule, the increment in $E(\delta^2 J)$ as a function of increments in S , P , K , and t is

$$\begin{aligned} 2dE(\delta^2 J) &= \text{tr}(P_0 dS) \Big|_{t=t_0} + (\Pi + \text{tr}(\dot{S}P)) dt \Big|_{t=t_f} - (\Pi + \text{tr}(\dot{S}P)) dt \Big|_{t=t_0} \\ &+ \int_{t_0}^{t_f} \left[\text{tr}(\Pi_S^T \delta S) + \text{tr}(\Pi_K^T \delta K) + \text{tr}(P \delta \dot{S}) + \text{tr}((\Pi_P^T + \dot{S}^T) \delta P) \right] dt. \end{aligned} \quad (\text{C.3})$$

To eliminate the variation in $\dot{S}$, integrate by parts to see that

$$\begin{aligned} \int_{t_0}^{t_f} \text{tr}(P \delta \dot{S}) dt &= \text{tr}(P \delta S) \Big|_{t=t_f} - \text{tr}(P \delta S) \Big|_{t=t_0} - \int_{t_0}^{t_f} \text{tr}(\dot{P} \delta S) dt \\ &= \text{tr}(P \delta S) \Big|_{t=t_f} - \text{tr}(P \delta S) \Big|_{t=t_0} - \text{tr}(P \dot{S}) dt \Big|_{t=t_f} + \text{tr}(P \dot{S}) dt \Big|_{t=t_0} - \int_{t_0}^{t_f} \text{tr}(\dot{P} \delta S) dt, \end{aligned} \quad (\text{C.4})$$

where the relation, $dS(t) = \delta S(t) + \dot{S}(t)dt$, has been used. Substitute this into (C.3), yielding

$$\begin{aligned} 2dE(\delta^2 J) &= \text{tr}(PdS)\Big|_{t=t_f} + \text{tr}((P_0 - P)dS)\Big|_{t=t_0} \\ &+ \left(\Pi + \text{tr}(\dot{S}P - P\dot{S})\right)dt\Big|_{t=t_f} - \left(\Pi + \text{tr}(\dot{S}P - P\dot{S})\right)dt\Big|_{t=t_0} \\ &+ \int_{t_0}^{t_f} \left[\text{tr}\left((\Pi_S - \dot{P})^T \delta S\right) + \text{tr}(\Pi_K^T \delta K) + \text{tr}\left((\Pi_P + \dot{S})^T \delta P\right) \right] dt. \end{aligned} \quad (\text{C.5})$$

The minimum of (9.36) is attained when $dE(\delta^2 J) = 0$ for all independent increments in its arguments. Setting to zero the coefficients of the independent increments δS , δK , and δP yields necessary conditions for a minimum as given in (9.37)-(9.39). Since $S(t_f)$, t_0 and t_f are given and fixed, $dS(t_f)$, dt_0 and dt_f are all zero. In (C.5), The three terms of increments dS , dt , dt evaluated at $t=t_f$, $t=t_0$, and $t=t_f$, respectively, are thus automatically equal to zero. Setting the coefficient of the second term in (C.5) to zero yields the boundary condition for a minimum as given in (9.40).

While it is straightforward to derive the explicit expressions of Π_S and Π_P , it is much involved to get the expression of Π_K . The details are given below.

Let $\Pi_1 = \text{tr}\left((A_x^T S + S A_x + \tilde{H}_{xx})P\right)$ and $\Pi_2 = \text{tr}\left((B_n^T S B_n + B_n^T \tilde{H}_{xn} + \tilde{H}_{nn})W_n\right)$. We have

$$\begin{aligned} \frac{\partial \Pi_1}{\partial K} &= \frac{\partial}{\partial K} \text{tr} \left(\left((f_x^T - C^T K^T f_u^T) S + S(f_x - f_u K C) \right. \right. \\ &\quad \left. \left. + \begin{bmatrix} I & -C^T K^T \end{bmatrix} \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix} \begin{bmatrix} I \\ -KC \end{bmatrix} \right) P \right) \\ &= \frac{\partial}{\partial K} \text{tr} \left(\left(f_x^T S - C^T K^T f_u^T S + S f_x - S f_u K C \right. \right. \\ &\quad \left. \left. + H_{xx} - C^T K^T H_{ux} - H_{xu} K C + C^T K^T H_{uu} K C \right) P \right) \\ &= -2f_u^T S P C^T - 2H_{ux} P C^T + 2H_{uu} K C P C^T, \end{aligned} \quad (\text{C.6})$$

and

$$\begin{aligned}
\frac{\partial \Pi_2}{\partial K} &= \frac{\partial}{\partial K} \text{tr} \left(\left(\begin{aligned} &\left[\begin{array}{c} f_w^T \\ -K^T f_u^T \end{array} \right] S \begin{bmatrix} f_w & -f_u K \end{bmatrix} \\ &+ \left[\begin{array}{c} f_w^T \\ -K^T f_u^T \end{array} \right] \begin{bmatrix} I & -C^T K^T \end{bmatrix} \begin{bmatrix} H_{xw} & H_{xv} - H_{xu} K \\ H_{uw} & H_{uv} - H_{uu} K \end{bmatrix} \\ &+ \begin{bmatrix} H_{ww} & H_{wv} - H_{wu} K \\ H_{vw} - K^T H_{uw} & H_{vv} + K^T H_{uu} K - K^T H_{uv} - H_{vu} K \end{bmatrix} \end{aligned} \right) W_n \right) \\
&= \frac{\partial}{\partial K} \text{tr} \left(\begin{aligned} &\left[\begin{array}{cc} f_w^T S f_w W_w & * \\ * & K^T f_u^T S f_u K W_v \end{array} \right] \\ &+ \left[\begin{array}{cc} (f_w^T H_{xw} - f_w^T C^T K^T H_{uw}) W_w & * \\ * & \begin{pmatrix} -K^T f_u^T (H_{xv} - H_{xu} K) \\ + K^T f_u^T C^T K^T (H_{uv} - H_{uu} K) \end{pmatrix} W_v \end{array} \right] \\ &+ \left[\begin{array}{cc} H_{ww} W_w & * \\ * & (H_{vv} + K^T H_{uu} K - K^T H_{uv} - H_{vu} K) W_v \end{array} \right] \end{aligned} \right) \\
&= 2f_u^T S f_u K W_v - H_{uw} W_w f_w^T C^T - f_u^T H_{xv} W_v + f_u^T H_{xu} K W_v + H_{ux} f_u K W_v \\
&\quad + f_u^T C^T K^T H_{uv} W_v + H_{uv} W_v K^T f_u^T C^T + 2H_{uu} K W_v - 2H_{uv} W_v \\
&\quad - \frac{\partial}{\partial K} \text{tr} (K^T f_u^T C^T K^T H_{uu} K W_v),
\end{aligned} \tag{C.7}$$

where the $*$'s denote terms of no interest. In particular, the last term in (C.7) can explicitly be derived based on the definition of the derivative of a trace of a matrix.

Consider a small perturbation, Δ , in K . The change caused in the trace is

$$\begin{aligned}
&\text{tr} \left((K + \Delta)^T f_u^T C^T (K + \Delta)^T H_{uu} (K + \Delta) W_v - K^T f_u^T C^T K^T H_{uu} K W_v \right) \\
&= \text{tr} \left(\begin{aligned} &\Delta^T f_u^T C^T K^T H_{uu} K W_v + K^T f_u^T C^T \Delta^T H_{uu} K W_v \\ &+ K^T f_u^T C^T K^T H_{uu} \Delta W_v \end{aligned} \right) + \text{tr} (O(\Delta^T \Delta)) \\
&= \text{tr} (\Delta^T f_u^T C^T K^T H_{uu} K W_v) + \text{tr} (K^T f_u^T C^T \Delta^T H_{uu} K W_v) \\
&\quad + \text{tr} (K^T f_u^T C^T K^T H_{uu} \Delta W_v) + \text{tr} (O(\Delta^T \Delta)),
\end{aligned} \tag{C.8}$$

where $O(\Delta^T \Delta)$ denotes the sum of all higher-order terms of Δ (not necessary in the form of ' $\Delta^T \Delta$ ') and is omitted when computing the change. Adding the term ' $\text{tr} (K^T f_u^T C^T K^T H_{uu} K W_v)$ ' to both sides of (C.8) results in an interpretation of the above equation as a Taylor expansion of $\text{tr} (K^T f_u^T C^T K^T H_{uu} K W_v)$ in its neighborhood. Thus the

derivative $\partial \text{tr}(K^T f_u^T C^T K^T H_{uu} K W_v) / \partial K$ can be computed as the sum of the derivatives of the first-three terms in (C.8) w.r.t. Δ , i.e.,

$$\begin{aligned} & \frac{\partial}{\partial K} \text{tr}(K^T f_u^T C^T K^T H_{uu} K W_v) \\ &= f_u^T C^T K^T H_{uu} K W_v + H_{uu} K W_v K^T f_u^T C^T + H_{uu} K C f_u K W_v. \end{aligned} \quad (\text{C.9})$$

Therefore, using (C.6) and (C.7) we obtain $\Pi_K = \partial \Pi_1 / \partial K + \partial \Pi_2 / \partial K$, which gives rise to the necessary condition of $\Pi_K = 0$ as given in (9.39).

If K is constant, then the necessary conditions of $\dot{P} = \Pi_s$ and $-\dot{S} = \Pi_p$ keep the same as those in (9.37)-(9.38) and the condition of $\Pi_K = 0$ has to be changed into

$\int_{t_0}^{t_f} \Pi_K dt = 0$, which can be seen from (C.5) by requiring both sides of the equation be equal to zero. This results in the necessary condition in (9.42), when $W_v = 0$.

Author's Publications

Journal Papers

1. W. Hu, L. M. Umar, G. Xiao, and V. Kariwala, Local self-optimizing control of constrained processes, *Journal of Process Control*, vol. 22, no. 2, pp. 488-493, 2012.
2. W. Hu, and G. Xiao, Self-clocking principle for congestion control in the Internet, *Automatica* (brief paper), vol. 48, no. 2, pp. 425-429, 2012.
3. W. Hu, G. Xiao, and X. Li, An analytical method for PID controller tuning with specified gain and phase margins for integral plus time delay processes, *ISA Transactions*, vol. 50, no. 2, pp. 268-276, 2011.
4. W. Hu, and G. Xiao, Analytical PI controller tuning using closed-loop setpoint response, *Industrial & Engineering Chemistry Research*, vol. 50, no. 4, pp. 2461-2466, 2011.
5. W. Hu, W.-J. Cai, and G. Xiao, Decentralized control system design for MIMO processes with integrators/differentiators, *Industrial & Engineering Chemistry Research*, vol. 49, no. 24, pp. 12521-12528, 2010.

Conference Papers

1. W. Hu, L. M. Umar, V. Kariwala, and G. Xiao, Local self-optimizing control with input and output constraints, in: *18th World Congress of the International Federation of Automatic Control (IFAC)*, Milano, Italy, Aug. 2011.
2. W. Hu, G. Xiao, and W.-J. Cai, PID controller design based on two-degrees-of-freedom direct synthesis, in: *23rd Chinese Control and Decision Conference (CCDC)*, Mianyang, China, May 2011.

3. W. Hu, W.-J. Cai, and G. Xiao, Relative gain array for MIMO processes containing integrators and/or differentiators, in: *11th International Conference on Automation, Robotics and Computer Vision (ICARCV)*, Singapore, Dec. 2010.
4. W. Hu, G. Xiao, and W.-J. Cai, Simple analytic formulas for PID tuning, in: *11th International Conference on Automation, Robotics and Computer Vision (ICARCV)*, Singapore, Dec. 2010.
5. W. Hu, and G. Xiao, Design of congestion control based on instantaneous queue sizes in the routers, in: *Proc. of IEEE Globecom*, Hawaii, USA, Nov. 2009.

Bibliography

- [1] A. O'Dwyer, *Handbook of PI and PID Controller Tuning Rules*, 3rd ed. London: Imperial College Press, 2009.
- [2] J.G. Ziegler and N.B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, vol. 64, no., pp. 759-768, 1942.
- [3] K.J. Åström and T. Hägglund, *Advanced PID Control*. Research Triangle Park, NC: ISA-The Instrumentation, Systems, and Automation Society, 2005.
- [4] M. Kano and M. Ogawa, "The state of the art in chemical process control in Japan: Good practice and questionnaire survey," *Journal of Process Control*, vol. 20, no., pp. 969-982, 2010.
- [5] A. Visioli and Q.C. Zhong, *Control of Integral Processes with Dead Time*. London: Springer, 2011.
- [6] S. Skogestad, "Simple analytic rules for model reduction and PID controller tuning," *Journal of Process Control*, vol. 13, no. 4, pp. 291-309, 2003.
- [7] D. Chen and D.E. Seborg, "PI/PID controller design based on direct synthesis and disturbance rejection," *Industrial and Engineering Chemistry Research*, vol. 41, no. 19, pp. 4807-4822, 2002.
- [8] A.S. Rao, V.S.R. Rao, and M. Chidambaram, "Direct synthesis-based controller design for integrating processes with time delay," *Journal of the Franklin Institute*, vol. 346, no. 1, pp. 38-56, 2009.
- [9] M. Lee, M. Shamsuzzoha, and T.N.L. Vu, "IMC-PID approach: An effective way to get an analytical design of robust PID controller," *International Conference on Control, Automation and Systems*, vol., pp. 2861-2866, Oct. 2008.
- [10] M. Shamsuzzoha and M. Lee, "Analytical design of enhanced PID filter controller for integrating and first order unstable processes with time delay," *Chemical Engineering Science*, vol. 63, no. 10, pp. 2717-2731, 2008.

- [11] M. Shamsuzzoha and S. Skogestad, "The setpoint overshoot method: A simple and fast closed-loop approach for PID tuning," *Journal of Process Control*, vol. 20, no., pp. 1220-1234, 2010.
- [12] C.C. Yu, *Autotuning of PID Controllers: A Relay Feedback Approach*. London: Springer, 2006.
- [13] S. Skogestad, "Plantwide control: The search for the self-optimizing control structure," *Journal of Process Control*, vol. 10, no. 5, pp. 487, 2000.
- [14] H. Manum, "Simple implementation of optimal control of process systems," Ph.D. dissertation, Norwegian University of Science and Technology, Trondheim, July 2010.
- [15] T.E. Marlin and A.N. Hrymak, "Real-time operations optimization of continuous processes," *AIChE Symposium Series*, vol. 93, no. 316, pp. 156-164, 1997.
- [16] V. Lersbamrungsuk, T. Srinophakun, S. Narasimhan, and S. Skogestad, "Control structure design for optimal operation of heat exchanger networks," *AIChE Journal*, vol. 54, no. 1, pp. 150-162, 2008.
- [17] E. Pistikopoulos, M. Georgiadis, and V. Dua, *Multi-parametric Programming: Theory, Algorithms and Applications*. Weinheim: Wiley-VCH, 2007.
- [18] Y. Cao, "Constrained self-optimizing control via differentiation," in *Proc. of Proc. of the 7th International Symposium on Advanced Control of Chemical Processes (ADCHEM)*, July 2004.
- [19] H. Dahl-Olsen, S. Narasimhan, and S. Skogestad, "Optimal output selection for control of batch processes," in *Proc. of American Control Conference*, June 2008.
- [20] H. Dahl-Olsen and S. Skogestad, "Near-optimal control of batch processes - by tracking of approximated sufficient conditions of optimality," in *Proc. of AIChE Annual Meeting*, Oral Presentation, Nov. 2009.
- [21] A. Visioli, *Practical PID Control*. London: Springer, 2006.
- [22] M.A. Johnson, M.H. Moradi, and J. Crowe, *PID Control: New Identification and Design Methods*. Berlin: Springer, 2005.

- [23] K.L. Chien, J.A. Hrones, and J.B. Reswick, "On the automatic control of generalized passive systems," *Trans. ASME*, vol. 74, no., pp. 175-185, 1952.
- [24] G.H. Cohen and G. Coon, "Theoretical consideration of retarded control," *Trans. ASME*, vol. 75, no. 1, pp. 827-834, 1953.
- [25] D.E. Rivera, M. Morarl, and S. Skogestad, "Internal model control. 4. PID controller design," *Industrial Engineering Chemistry Process Design and Development*, vol. 25, no. 1, pp. 252-265, 1986.
- [26] J.G. Truxal, *Automatic Feedback Control System Synthesis*. McGraw-Hill Education, 1955.
- [27] K.J. Åström and T. Hägglund, "Revisiting the Ziegler-Nichols step response method for PID control," *Journal of Process Control*, vol. 14, no. 6, pp. 635-650, 2004.
- [28] F.S. Taip and M.T. Tham, "Validity of several tuning methods for different PID algorithms," *International Journal of Engineering and Technology*, vol. 4, no. 1, pp. 22-30, 2007.
- [29] K.J. Åström and T. Hägglund, "Automatic tuning of simple regulators with specifications on phase and amplitude margins," *Automatica*, vol. 20, no. 5, pp. 645-651, 1984.
- [30] H. Hjalmarsson, S. Gunnarsson, and M. Gevers, "A convergent iterative restricted complexity control design scheme," in *Proc. of Conference on Decision and Control* 1994.
- [31] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin, "Iterative feedback tuning: Theory and applications," *IEEE Control Systems Magazine*, vol. 18, no. 4, pp. 26-41, 1998.
- [32] J. Crowe, M. Johnson, and M. Grimble, "PID parameter cycling to tune industrial controllers: a new model-free approach," in *Proc. of 13th IFAC Symposium on System Identification*, Aug. 2003.
- [33] B. Srinivasan, S. Palanki, and D. Bonvin, "Dynamic optimization of batch processes: I. Characterization of the nominal solution," *Computers & Chemical Engineering*, vol. 27, no. 1, pp. 1-26, 2003.

- [34] V. Alstad, S. Skogestad, and E. Hori, "Optimal measurement combinations as controlled variables," *Journal of Process Control*, vol. 19, no. 1, pp. 138-148, 2009.
- [35] K. Ariyur and M. Krstic, *Real-time Optimization by Extremum-seeking Control*. Hoboken, NJ: John Wiley & Sons, 2003.
- [36] M. Guay and T. Zhang, "Adaptive extremum seeking control of nonlinear dynamic systems with parametric uncertainties," *Automatica*, vol. 39, no. 7, pp. 1283-1293, 2003.
- [37] S. Skogestad, "Near-optimal operation by self-optimizing control: From process control to marathon running and business systems," *Computers & chemical engineering*, vol. 29, no. 1, pp. 127-137, 2004.
- [38] B. Srinivasan, D. Bonvin, E. Visser, and S. Palanki, "Dynamic optimization of batch processes: II. Role of measurements in handling uncertainty," *Computers & Chemical Engineering*, vol. 27, no. 1, pp. 27-44, 2003.
- [39] J. Kadam, W. Marquardt, B. Srinivasan, and D. Bonvin, "Optimal grade transition in industrial polymerization processes via NCO tracking," *AIChE Journal*, vol. 53, no. 3, pp. 627-639, 2007.
- [40] I. Halvorsen, S. Skogestad, J. Morud, and V. Alstad, "Optimal selection of controlled variables," *Industrial and Engineering Chemistry Research*, vol. 42, no. 14, pp. 3273-3284, 2003.
- [41] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, 1st ed. Chichester, UK: John Wiley & Sons, 1996.
- [42] E. Hori and S. Skogestad, "Selection of controlled variables: Maximum gain rule and combination of measurements," *Industrial & Engineering Chemistry Research*, vol. 47, no. 23, pp. 9465-9471, 2008.
- [43] V. Kariwala, Y. Cao, and S. Janardhanan, "Local self-optimizing control with average loss minimization," *Industrial and Engineering Chemistry Research*, vol. 47, no. 4, pp. 1150-1158, 2008.
- [44] V. Kariwala, "Optimal measurement combination for local self-optimizing control," *Industrial & Engineering Chemistry Research*, vol. 46, no. 11, pp. 3629-3634, 2007.

- [45] S. Heldt, "Dealing with structural constraints in self-optimizing control engineering," *Journal of Process Control*, vol. 20, no. 9, pp. 1049-1058, 2010.
- [46] V. Alstad and S. Skogestad, "Null space method for selecting optimal measurement combinations as controlled variables," *Industrial and Engineering Chemistry Research*, vol. 46, no. 3, pp. 846-853, 2007.
- [47] Y. Cao and V. Kariwala, "Bidirectional branch and bound for controlled variable selection: Part I. Principles and minimum singular value criterion," *Computers & Chemical Engineering*, vol. 32, no. 10, pp. 2306-2319, 2008.
- [48] V. Kariwala and Y. Cao, "Bidirectional branch and bound for controlled variable selection. Part II: Exact local method for self-optimizing control," *Computers & Chemical Engineering*, vol. 33, no. 8, pp. 1402-1412, 2009.
- [49] V. Kariwala and Y. Cao, "Bidirectional branch and bound for controlled variable selection. Part III: local average loss minimization," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 1, pp. 54-61, 2010.
- [50] H. Manum, S. Narasimhan, and S. Skogestad, "A new approach to explicit MPC using self-optimizing control," <http://www.nt.ntnu.no/users/skoge/publications/2007/>, 2007.
- [51] R. Yelchuru and S. Skogestad, "MIQP formulation for optimal controlled variable selection in self optimizing control," in *Proc. of PSE Asia*, July 2010.
- [52] R. Yelchuru, S. Skogestad, and H. Manum, "MIQP formulation for controlled variable selection in self-optimizing control," in *Proc. of IFAC International Symposium on Dynamics and Control of Process Systems*, July 2010.
- [53] R. Yelchuru and S. Skogestad, "Optimal controlled variable selection for individual process units in self-optimizing control with MIQP formulation," *American Control Conference*, vol., pp., June 2011.
- [54] R. Yelchuru and S. Skogestad, "Optimal controlled variable selection with structural constraints using MIQP formulations," *18th IFAC Congress*, vol., pp., Aug. 2011.
- [55] A. O'Dwyer, *Handbook of PI and PID Controller Tuning Rules*, 2nd ed. London: Imperial College Press, 2006.

- [56] M. Kano and M. Ogawa, "The state of art in advanced process control in Japan," *IFAC symposium ADCHEM*, vol., pp. 2009.
- [57] A. Ali and S. Majhi, "PID controller tuning for integrating processes," *ISA transactions*, vol. 49, no. 1, pp. 70-78, 2010.
- [58] A. Seshagiri Rao, V.S.R. Rao, and M. Chidambaram, "Direct synthesis-based controller design for integrating processes with time delay," *Journal of the Franklin Institute*, vol. 346, no. 1, pp. 38-56, 2009.
- [59] P. Garcia and P. Albertos, "A new dead-time compensator to control stable and integrating processes with long dead-time," *Automatica*, vol. 44, no. 4, pp. 1062-1071, 2008.
- [60] B. Wang, D. Rees, and Q.C. Zhong, "Control of integral processes with dead time. Part IV: various issues about PI controllers," *IEE Proc.-Control Theory and Appl.*, vol. 153, no. 3, pp. 302-306, 2006.
- [61] W. Luyben, "Tuning Proportional- Integral- Derivative Controllers for Integrator/Deadtime Processes," *Ind. Eng. Chem. Res.*, vol. 35, no. 10, pp. 3480-3483, 1996.
- [62] I. Kookos, A. Lygeros, and K. Arvanitis, "On-line PI controller tuning for integrator/dead time processes," *Eur J Control*, vol. 5, no. 1, pp. 19-31, 1999.
- [63] A. Visioli, "Optimal tuning of PID controllers for integral and unstable processes," in *Proc.* 2002.
- [64] Y. Wang and W. Cai, "Advanced Proportional- Integral- Derivative Tuning for Integrating and Unstable Processes with Gain and Phase Margin Specifications," *Ind. Eng. Chem. Res.*, vol. 41, no. 12, pp. 2910-2914, 2002.
- [65] M. Chidambaram and R. Padma Sree, "A simple method of tuning PID controllers for integrator/dead-time processes," *Computers & chemical engineering*, vol. 27, no. 2, pp. 211-215, 2003.
- [66] W. Hu, G. Xiao, and W.-j. Cai, "Simple analytic formulas for PID tuning," in *Proc. of 11th International Conference on Automation, Robotics and Computer Vision (ICARCV)*, Dec. 2010.

- [67] Q.G. Wang, "Handbook of PI and PID Controller Tuning Rules, Aidan O'Dwyer, Imperial College Press, London, 375pp, ISBN 1-86094-342-X, 2003," *Automatica*, vol. 41, no. 2, pp. 355-356, 2005.
- [68] C.H. Lee, "A survey of PID controller design based on gain and phase margins (invited paper)," *International Journal of Computational Cognition*, vol. 2, no. 3, pp. 63-100, 2004.
- [69] K.K. Tan, Q.-G. Wang, C.C. Hang, and T. Hägglund, *Advances in PID Control*. London: Springer, 1999.
- [70] W.K. Ho, C.C. Hang, and L.S. Cao, "Tuning of PID controllers based on gain and phase margin specifications," *Automatica*, vol. 31, no. 3, pp. 497-502, 1995.
- [71] H.W. Fung, Q.G. Wang, and T.H. Lee, "PI tuning in terms of gain and phase margins," *Automatica*, vol. 34, no. 9, pp. 1145-1149, 1998.
- [72] Q.G. Wang, H.W. Fung, and Y. Zhang, "PID tuning with exact gain and phase margins," *ISA transactions*, vol. 38, no. 3, pp. 243-249, 1999.
- [73] W. Hu, G. Xiao, and X. Li, "PI/PD/PID tuning rules for IPTD (integral plus time delay) processes and their realized GPM (gain and phase margins)," *Supplemental Material*, http://www3.ntu.edu.sg/home/EGXXiao/GPM_Tables.pdf, 2010.
- [74] I.L. Chien and P.S. Fruehauf, "Consider IMC tuning to improve controller performance," *Chem. Eng. Prog.*, vol. 86, no. 10, pp. 33-41, 1990.
- [75] D.E. Rivera, M. Morarl, and S. Skogestad, "Internal model control. 4. PID controller design," *Industrial Engineering Chemistry Process Design and Development*, vol. 25, no. 1, pp. 252-265, 1986.
- [76] W. Hu, G. Xiao, and X. Li, "An analytical method for PID controller tuning with specified gain and phase margins for integral plus time delay processes," *ISA Transactions*, doi:10.1016/j.isatra.2011.01.001, 2011.
- [77] K. Ogata, *Modern Control Engineering*, 4th ed. Inc. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
- [78] D. Seborg, T. Edgar, and D. Mellichamp, *Process Dynamics and Control*, 2nd ed. New York, USA: John Wiley, 2004.

- [79] G. Szita and C.K. Sanathanan, "Robust design for disturbance rejection in time delay systems," *Journal of the Franklin Institute*, vol. 334, no. 4, pp. 611-629, 1997.
- [80] G.C. Goodwin, S.F. Graebe, and M.E. Salgado, *Control System Design*. Upper Saddle River, NJ: Prentice Hall, 2001.
- [81] M. Shamsuzzoha and M. Lee, "Design of advanced PID controller for enhanced disturbance rejection of second-order processes with time delay," *AIChE Journal*, vol. 54, no. 6, pp. 1526-1536, 2008.
- [82] T. Liu, W. Zhang, and D. Gu, "Analytical design of two-degree-of-freedom control scheme for open-loop unstable processes with time delay," *Journal of Process Control*, vol. 15, no. 5, pp. 559-572, 2005.
- [83] Y. Lee, J. Lee, and S. Park, "PID controller tuning for integrating and unstable processes with time delay," *Chemical Engineering Science*, vol. 55, no. 17, pp. 3481-3493, 2000.
- [84] Y. Lee, S. Park, M. Lee, and C. Brosilow, "PID controller tuning for desired closed-loop responses for SI/SO systems," *AIChE Journal*, vol. 44, no. 1, pp. 106-115, 1998.
- [85] M. Lee, M. Shamsuzzoha, and T.N.L. Vu, "IMC-PID approach: An effective way to get an analytical design of robust PID controller," in *Proc. of International Conference on Control, Automation and Systems*, Oct. 2008.
- [86] M. Shamsuzzoha and M. Lee, "Design of advanced PID controller for enhanced disturbance rejection of second-order processes with time delay," *AIChE Journal*, vol. 54, no. 6, pp. 1526, 2008.
- [87] M. Shamsuzzoha and M. Lee, "An Enhanced Performance PID Filter Controller for First Order Time Delay Processes," *Journal of Chemical Engineering of Japan*, vol. 40, no. 6, pp. 501, 2007.
- [88] M. Morari and E. Zafiriou, *Robust Process Control*. Prentice Hall, 1989.
- [89] M. Shamsuzzoha and M. Lee, "IMC-PID controller design for improved disturbance rejection of time-delayed processes," *Industrial and Engineering Chemistry Research*, vol. 46, no. 7, pp. 2077-2091, 2007.

- [90] C. Xiang, Q.G. Wang, X. Lu, L.A. Nguyen, and T.H. Lee, "Stabilization of second-order unstable delay processes by simple controllers," *Journal of Process Control*, vol. 17, no. 8, pp. 675-682, 2007.
- [91] A. Leva and F. Donida, "Normalised expression and evaluation of PI tuning rules," in *Proc. of IFAC*, July 2008.
- [92] M. Yuwana and D. Seborg, "A new method for on-line controller tuning," *AIChE Journal*, vol. 28, no. 3, pp. 434-440, 1982.
- [93] O. Axelsson, *Iterative Solution Methods*. London, UK: Cambridge University Press, 1996.
- [94] H. Manum, "Simple implementation of optimal control of process systems," Ph.D. dissertation, Norwegian University of Science and Technology, Trondheim, Jul. 2010.
- [95] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. of CACSD Conference*, Sep. 2004.
- [96] R.B. Newell and P.L. Lee, *Applied Process Control : A Case Study*. Brookvale, Australia: Prentice Hall, 1989.
- [97] C.R. Johnson and E.A. Schreiner, "The relationship between AB and BA," *The American Mathematical Monthly*, vol. 103, no. 7, pp. 578-582, 1996.
- [98] M.A. Dahleh and I.J. Diaz-Bobillo, *Control of Uncertain Systems: A Linear Programming Approach*. Upper Saddle River, NJ: Prentice-Hall, 1995.
- [99] V. Kariwala and S. Skogestad, "L1/Q approach for efficient computation of disturbance rejection measures for feedback control," *Journal of Process Control*, vol. 17, no. 6, pp. 501-508, 2007.
- [100] G. Sierksma, *Linear and Integer Programming: Theory and Practice*, 2nd ed. New York: Marcel Dekker, 2002.
- [101] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 1, pp. 625-653, 1999.

- [102] A.E. Bryson and Y.C. Ho, *Applied Optimal control: Optimization, Estimation, and Control*. Bristol, PA: Hemisphere, 1975.
- [103] F.L. Lewis and V.L. Syrmos, *Optimal Control*, 2nd ed. New York: Wiley-Interscience, 1995.
- [104] J.P. Hespanha, *Linear Systems Theory*. New Jersey, USA: Princeton University Press, 2009.
- [105] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.
- [106] F.L. Lewis, L. Xie, and D. Popa, *Optimal and Robust Estimation: With an Introduction to Stochastic Control Theory*, 2nd ed. Boca Raton: CRC, 2007.
- [107] D. Moerder and A. Calise, "Convergence of a numerical algorithm for calculating optimal output feedback gains," *IEEE Transactions on Automatic Control*, vol. 30, no. 9, pp. 900-903, 1985.
- [108] V. Blondel and J.N. Tsitsiklis, "NP-hardness of some linear control design problems," *SIAM Journal of Control and Optimization*, vol. 35, no. 6, pp. 2118-2127, 1997.
- [109] M. Fu, "Pole placement via static output feedback is NP-hard," *IEEE Transactions on Automatic Control*, vol. 49, no. 5, pp. 855-857, 2004.
- [110] C. Grimholt, "Verification and improvements of the SIMC method for PI control," <http://www.nt.ntnu.no/users/skoge/diplom/prosjekt10/grimholt>, Technical report. 5th year project. Department of Chemical Engineering, Norwegian University of Science and Technology, 2010.
- [111] J. Jäschke and S. Skogestad, "Optimal controlled variables for polynomial systems," *Journal of Process Control*, vol. 22, no. 1, pp. 167-179, 2012.