Optimal tuning of PID controllers

And the verification of the SIMC rules

Chriss Grimholt

Norwegian University of Science and Technology

Abstract

The simple PI/PID controller is the most used controller in the process industry. However, finding good settings for this simple controller is not trivial. One simple way of finding settings for a PID controller is the popular SIMC tuning rule. This thesis deals with the following main topics,

- Finding optimal parameters for fixed-order controllers.
- Validating that the SIMC tuning rules are close-to optimal for first-order with delay processes.
- Validating that the SIMC tuning rules are close-to optimal for double integrating with delay processes.
- Showing that a well tuned PID controller is a better choice than a Smith predictor.

A central part of this thesis is defining and quantifying an optimal PID controller. We optimize performance (using integrated absolute error for input and output disturbances as the cost function) subject to constraints on robustness (using the sensitivity peaks M_s and M_T). We provide analytical gradients for the cost function and the constraints, which give increased accuracy and better convergence properties than numerical gradients.

Using this, we derive optimal PI- and PID-settings for first-order plus delay processes. The optimal PID IAE-performance is compared

with the IAE-performance for controllers that have been tuned using the SIMC-rules, where the SIMC tuning parameter τ_c is adjusted such that the two controllers have the same robustness. The original SIMC-rules give a PI-controller for a first-order plus delay process, and we find that this controller is close to the optimal PI-controller. The only exception is for delay-dominant processes where the SIMC-rule gives a pure integrating controller. We propose and study a very simple modification to the original SIMC-rule, which is to add a derivative time $\tau_d = \theta/3$, where θ is the delay. This gives performance close to the IAE-optimal PID-controller also for delay-dominant processes. We call this the "improved" SIMC-rule, but we put improved in quotes, because this controller requires more input usage.

We also investigate optimal PID control of a double integrating with delay process and compare with the SIMC rules. What makes the double integrating process special is that derivative action is actually necessary for stabilization. Surprisingly, the SIMC PID controller is almost identical to the optimal PID controller. We also propose a generalized SIMC rule which includes second-order processes with large time constants.

For processes with large time delays, the common belief is that PI and PID controllers give poor performance, and that a Smith predictor or a similar dead-time compensator can significantly improve performance. In the last chapter, we argue that this is a myth. For a given robustness level, we find that the performance improvement with the Smith predictor is small even for a pure time delay process. For other first-order with delay processes a PID controller is generally better for a given robustness level. In addition, the Smith predictor is much more sensitive to time delay errors than PI and PID controllers.

Acknowledgments

First of all, I want to express my sincere gratitude to my supervisor Professor Sigurd Skogestad. His passion for process control has been inspiring, and it has been a pleasure and a privilege to work with him. I would never have been able to complete this thesis without his guidance, his encouragement, his incredible patience, and *just* the right amount of pushing.

A sincere thanks goes to my co-supervisor Johannes Jäschke for his positiveness and support. I also want to give a sincere thanks to Professor Heinz A. Preisig for his valuable insight and all the interesting discussions.

I want to give many heartfelt thanks to my office mates Vinicius de Oliveira, Vladimiros Minasidis and Sebastian Roll for all the happy times trough the years.

I want to thank Esmaeil Jahanshahi and Kristian Soltesz for showing such interest in my work and extending on it.

A special thanks goes to Axel Lødemel Holene and Martin S. Foss, and Rannei Solbak Simonsen which helped me validate some of the work presented in this thesis as part of their master studies at NTNU.

Finally, I would like to thank my family and friends for their love, encouragement and endless support.

Contents

Chapter One

Introduction

1.1 Motivation

In the process industry, almost all feedback loops are controlled using a P, PI or PID controller (later PID will be used to refer to all subset controllers). It is also a known fact that most of these PID controllers are poorly tuned, and many have never been tuned at all. This agrees well with my own industrial experience with tuning PID controllers. For example, I commonly observe loops with self-induced oscillations because of excessive integral action. Another common problem is oscillating cascade loops where the outer loop is tuned much faster than the inner loop.

Many of these loops have most certainly been tuned by someone with little or no training in controller tuning. However, even with a background in process control tuning is not straight-forward. There are hundreds, if not thousands, of published tuning rules. Which ones are good? Which ones are bad? Do we need all of them?

When performing tuning in practice, there is usually no time to identify a detailed model nor conduct simulations to find the most

appropriate tuning. Usually, one uses a simple engineering identification method like the open-loop step test. Based on this basic model, one need to find controller settings that gives the desired process behavior. The desired behavior is an engineering decision, but usually depends on the size of disturbances, the limits on variation, and the interaction with other controller loops.

Ziegler and Nichols (ZN) were probably the first to systematically looked at how to optimally tune PID controllers. This work was groundbreaking, and the ZN-rule is still taught as a method for tuning PID controllers to this date. At the time of publication (1940s), optimal behavior was thought to be a controller response with a quarter decay ratio. This results in PID controllers that are quite aggressive and have little robustness. In addition, the ZN-rule has no tuning constant to adjust the speed of the controller, and more importantly the desired process behavior.

To find good controller settings, I have mostly used the SIMC rules. This applies to this thesis and to my subsequent years at ABB. From a simple first or second order model, the SIMC rule gives a PI or a PID controller, respectively. The rule has a tuning constant, the closed-loop response time τ_c, which can be used to speed up or slow down the controlled response. By adjusting this tuning constant, we can tune the process to give the desired behavior. However, how good are actually the SIMC rules? This is one of the main questions we try to explore and answer in this thesis.

So how do we go about analyzing how good a tuning rule is? Most publication on tuning rules focus on performance, and most commonly only setpoint performance. A typical conclusion is "*our tuning rule gives a faster response and less deviation from setpoint than the other tuning rules we compared with*". These works usually ignore the central trade-off perspective between performance and

robustness. That is, that a fast controller will usually have worse robustness and higher input usage than a slow controller. What is most important, robustness or performance? Again, that is an engineering decision.

1.2 The SIMC procedure

The SIMC procedure (Skogestad, 2003) is a two step approach,

1. Obtain a first- or second-order plus delay model.
2. Derive model-based controller settings.

Where PI-settings result if we start from a first-order model, and PID-settings result from a second-order model.

This thesis does not deal with the topic of obtaining a model for tuning (step 1), and assumes that a correct first or second order model is available. However, a short summary on obtaining models are given below.

The first step in the SIMC design procedure is to obtain a first-order plus delay (FOPD) on the form

$$G_1(s) = \frac{k}{\tau_1 s + 1} e^{-\theta s} = \frac{k'}{s + 1/\tau_1} e^{-\theta s} \tag{1.1}$$

or a second-order plus delay (SOPD) model on the form

$$G_2(s) = \frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)} e^{-\theta s} \tag{1.2}$$

If a higher order model is available, an approximate first- or second order model can be obtain by using the half-rule detailed in Skogestad (2003). That is, the largest neglected denominator time constant is distributed evenly to the delay (θ) and the smallest retained time constant (τ_1 or τ_2). The rest of the neglected denominator time

constants are added to the delay. Also, if the model has inverse response in the numerator (positive zeros), these are also added to the delay. That is, given the higher order model,

$$G_i(s) = \frac{\prod_j(-T_{j0}^{\text{inv}}s + 1)}{\prod_i(\tau_{i0}s + 1)}e^{-\theta_0 s} \tag{1.3}$$

To obtain a FOPD model,

$$\tau_1 = \tau_{10} + \frac{\tau_{20}}{2}; \quad \theta = \theta_0 + \frac{\tau_{20}}{2} + \sum_{i\geq 3}\tau_{i0} + \sum_j T_{j0}^{\text{inv}} + \frac{h}{2} \tag{1.4}$$

and to obtain a SOPD model

$$\tau_1 = \tau_{10}; \quad \tau_2 = \tau_{20} + \frac{\tau_{30}}{2}; \quad \theta = \theta_0 + \frac{\tau_{30}}{2} + \sum_{i\geq 4}\tau_{i0} + \sum_j T_{j0}^{\text{inv}} + \frac{h}{2} \tag{1.5}$$

where h is the sampling time. If the model has positive numerator time constants (negative zeros), there are several rules for how to approximate them in Skogestad (2003).

In practice, the model parameters for a first-order model are commonly obtained from a step response experiment. This is done with the controller in manual and performing a step on the controller output, then observing the response in the measurement. If the response is fairly first order in shape, the approximate FOPD can easily be calculated. The time delay θ is the time from the step on the controller output to the first response is observed in the measurement. The process gain k is the change in stationary value of the measurement before and after the step divided by size of the step. The time constant τ_1 is the time from the first response in the measurement to the measurement has reached 63% of its new stationary value.

For processes with a large time constant τ_1, one has to wait a long time for the process to settle and reach its new stationary

value. Fortunately, it is generally not necessary to run the experiment for longer than about 10 times the delay. At this time, one may simply stop the experiment and either extend the response "by hand" towards settling, or approximate it as an integrating process.

$$\frac{ke^{-\theta s}}{\tau_1 s + 1} = \frac{k' e^{-\theta s}}{s + 1/\tau_1} \approx \frac{k' e^{-\theta s}}{s} \tag{1.6}$$

where $k' = k/\tau_1$ is the slope of the integrating response.

The reason is that for lag-dominant processes, that is for $\tau_1 > 8\theta$ approximately, the individual values of the time constant τ_1 and the gain k are not very important for controller design. Rather, their ratio k' determines the PI-settings, as is clear from the SIMC tuning rules presented below.

In some cases, open-loop responses may be difficult to obtain, and using closed-loop data may be more effective. The most famous closed-loop experiment is the Ziegler-Nichols where the system is brought to sustained oscillations by use of a P-controller. One disadvantage with the method is that the system is brought to its instability limit. An alternative is to use relay feedback, which induce the same type of oscillations, but without the need to bring the system to its instability limit. However, for these methods only two pieces of information is used, the ultimate gain K_u and the ultimate period P_u, so the method cannot possibly work on a wide range of first-order plus delay processes, which we know are described by three parameters (k, τ_1, θ). Asymmetric relay feedback induces an improved excitation which results in an asymmetric oscillation, and better models can be obtained (e.g. a FOPD model) (Berner, 2015). A different closed-loop approach which results in a FOPD model is to perform a setpoint step with a P-controller as described by Shamsuzzoha and Skogestad (2010).

For a serial (cascade) form PID controller

$$K_{\text{PID}}(s) = \frac{k_c(\tau_i s + 1)(\tau_d s + 1)}{\tau_i s}, \tag{1.7}$$

the SIMC PID-rules for a second-order plus delay process are (Skogestad, 2003)

$$k_c = \frac{1}{k}\frac{\tau_1}{(\tau_c + \theta)}, \quad \tau_i = \min\left\{\tau_1,\, 4(\tau_c + \theta)\right\}, \quad \tau_d = \tau_2 \tag{1.8}$$

where k_c, τ_i and τ_d are the controller gain, integral time and derivative time, respectively. Note that for a first-order process ($\tau_2 = 0$) we get a PI-controller (with $\tau_d = \tau_2 = 0$). The SIMC tuning parameter τ_c (sometimes called λ) corresponds to the desired closed-loop time constant.

The tuning parameter can be used to speed up or slow down the controller, and thus to reduce or increase the robustness. The effect of changing τ_c/θ on four common robustness measures (M_s, GM, PM and DM) is shown in Figure 1.1. The robustness margins are somewhat worse for cases with large time constants were we use $\tau_i = 4(\tau_c + \theta)$ rather than $\tau_i = \tau_1$. Note that this figure is not included in the original SIMC-paper (Skogestad, 2003).

For the case where one wants "tight" (fast) control, Skogestad (2003) recommended to select the tuning constant as $\tau_c = \theta$ to maintain good robustness with

Sensitivity peak (M_s) = 1.59 – 1.7,
Gain margin (GM) = 3.14 – 2.96,
Phase margin (PM) = 61.4° – 46.9°,
Delay margin (DM) = 2.14θ – 1.59θ.

Here the first value (best robustness) is for processes with small time constants τ_1 where we use $\tau_i = \tau_1$. The second value is for an integrating process where we use $\tau_i = 4(\tau_c + \theta)$.

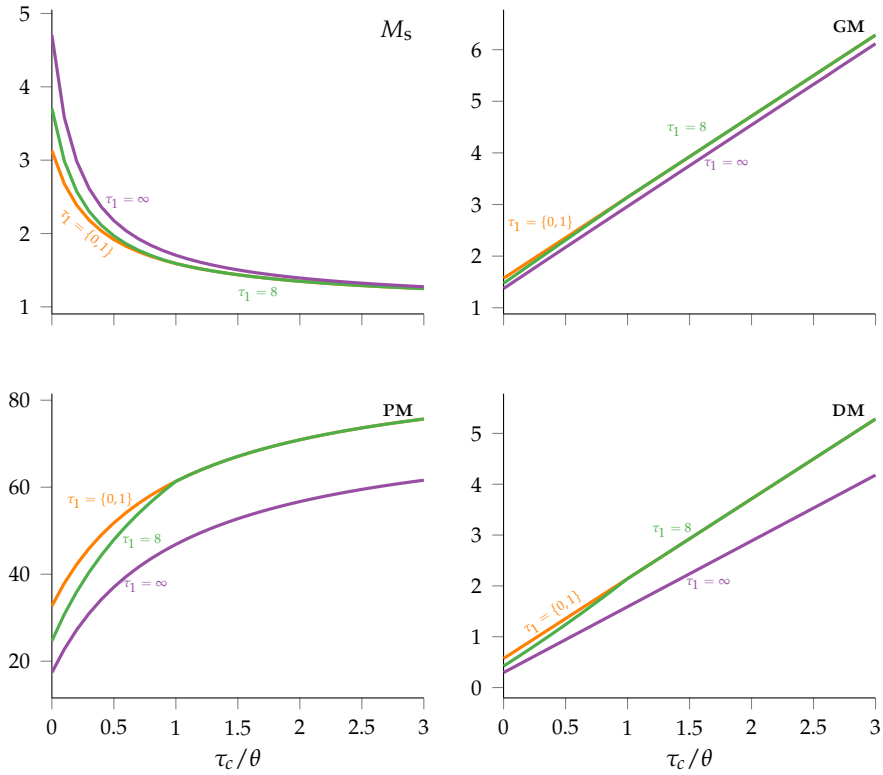

Figure 1.1: SIMC PID-rule for first-order plus delay process: Effect of tuning the parameter τ_c/θ on robustness parameters: M_S, GM, PM and DM

Note that the SIMC-tunings are for a serial (cascade) PID-controller. For the more common parallel (ideal) PID implementation

$$K_{\text{PID}}^{\text{parallel}}(s) = \tilde{k}_c \left(1 + \frac{1}{\tilde{\tau}_i s} + \tilde{\tau}_d s\right), \tag{1.9}$$

one must compute the factor $f = 1 + \tau_d/\tau_i$, and use the following settings

$$\tilde{k}_c = k_c f, \quad \tilde{\tau}_i = \tau_i f, \quad \text{and} \quad \tilde{\tau}_d = \tau_d / f. \tag{1.10}$$

1.3 Organization of thesis

This thesis consist of one introductory chapter, and four main chapters. These four main chapters are based on journal and conference papers, and are centered around the theme of optimal PI and PID tuning. Two papers related to the thesis where I am a co-author, but *not* the first author, are included in Appendix B and C for completeness and to show some applications. The precise structure of the thesis is as follows,

Chapter 2 Provides a summary of the previous work related to finding optimal PID parameters, and introduces the optimization problem solved in this thesis. The cost function related to controller performance (IAE) and the constraints related to controller robustness (M_S and M_T) are explained. The exact gradients for the cost function and constraints are then presented. This is followed by a small case study for finding optimal PID parameters that compares optimizing with approximated gradients (finite forward differentiation) and the presented exact gradients. This chapter also briefly presents gradients for other controllers and constraint parametrization.

Chapter 3 The optimal trade-off between controller performance and robustness (Pareto front) are found using the optimization problem presented in Chapter 2 for PI and PID controllers. This is done for four processes ranging from pure time delay to integrating process, and the benefits of using PID control is explored. The optimal PI and PID parameters for a range of first order processes are analyzed, followed by proposed improvements to the SIMC rules, namely *i*SIMC-PI and *i*SIMC.

Similar performance-robustness curves are generated for the SIMC rules by evaluating the rule for different values of the rules tuning constant on the same four processes. Optimality of SIMC is then analyzed by comparing the optimal performance-robustness trade-off with the corresponding trade-off for the SIMC rules. The same comparison is also done for the the proposed improved rules *i*SIMC-PI and *i*SIMC. In addition, the classical ZN rules have also been compared.

This chapter also briefly compares the parallel and serial PID implementation, discuss the implication of measurement filtering on PID control, and performance balance between input and output disturbances.

Chapter 4 This chapter extends the analysis done in Chapter 3 to finding optimal PID controllers for double integrating processes. The optimal trade-off is compared with the resulting SIMC. According to the SIMC rules, a second order process with large time constants, should be approximated as a double integrating process. However, this chapter also presents a generalized SIMC rule for second order process that also covers this double integrating case.

Chapter 5 This chapter extends the work done in Chapter 3 to analyzing Smith predictor (SP). The optimal SP is compared with optimal PI and PID controllers. The chapter also discuss some of the common SP robustness issues.

Appendix A This appendix gives the derivation of the exact gradients presented in Chapter 2.

Appendix B This paper shows how the work can be extended to simultaneous find optimal PID parameters and measurement filter.

Appendix C This paper show how the thesis work can be extended to find and analyses proportional-integral-derivative-filter (PIDF) controller when applied to riser slugging.

Appendix D This paper is about optimization of oil field production under gas coning conditions using a soft sensor to estimate the optimal conditions. Because it does not concern the main theme of the thesis, namely PID tuning, it has been placed in the appendix.

1.4 Contribution of this thesis

The main contribution of this thesis is summaries as follows,

- Gives the exact gradients to efficiently solve a PID tuning optimization problem with an IAE cost function and M_S & M_T robustness constraints.
- Finds the optimal performance-robustness trade-off for PI, PID and SP.
- Quantifying the benefits of using PID compared to PI.

- Concluding that even for processes with large time delays a PID controller should be used instead of a SP.
- Quantifying the benefits of using the parallel (can have complex zeros) versus the serial (limited to two real zeros) PID controller.
- Validating the the SIMC rules has a close-to optimal performance-robustness trade-off for first order and integrating processes.
- Proposing the improved SIMC rules, namely *i*SIMC-PI and *i*SIMC that have close-to optimal trade-off also for small time constant and close-to pure time delay processes.
- Proposing a generalized second order SIMC rule that also covers a second order process with large time constant. That is, close-to double integrating.

1.5 Publications resulting from the thesis work

Chapter 2 Published as:

- Chriss Grimholt and Sigurd Skogestad. Optimization of fixed order controllers using exact gradients. *Journal of Process Control*, 71:130–138, 2018a.

This is an extension of the following conference publication

- Chriss Grimholt and Sigurd Skogestad. Improved optimization-based design of PID controllers using exact gradients. In *12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering*, volume 37, pages 1751–1757, 2015a

Chapter 3 Published as:

- Chriss Grimholt and Sigurd Skogestad. Optimal PI and PID control of first-order plus delay processes and evaluation of

the original and improved SIMC rules. *Journal of Process Control*, 70, 2018b

This is an extension of the following publications:

- Chriss Grimholt and Sigurd Skogestad. Optimal PI-control and verification of the SIMC tuning rule. In *IFAC conference on Advances in PID control (PID'12)*. The International Federation of Automatic Control, March 2012a
- Chriss Grimholt and Sigurd Skogestad. The SIMC method for smooth pid controller tuning. In Ramon Vilanova and Antonio Visioli, editors, *PID control in the third Millennium – Lessons Learned and new approaches*. Springer, 2012b
-

Chapter 4 Published as:

- Chriss Grimholt and Sigurd Skogestad. Optimal PID control of double integrating processes. In *11th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems*, pages 127–132, NTNU, Trondheim, Norway, 6 2016

Chapter 5 Published as:

- Chriss Grimholt and Sigurd. Skogestad. Should we forget the smith predictor? In *3rd IFAC conference on Advances in PID control, Ghent, Belgium, 9-11 May 2018.*, 2018c

Appendix B Published as:

- Kristian Soltesz, Chriss Grimholt, and Sigurd Skogestad. Simultaneous design of pid controller and measurement filter by optimization. *IET Control Theory & Applications*, 11(3):341–348, 2017b

Appendix C Published as:

- E. Jahanshahi, V. De Oliveira, C. Grimholt, and Skogestad S. A comparison between internal model control, optimal PIDF and robust controllers for unstable flow in risers. In *19th World Congress*. The International Federation of Automatic Control, 2014

Appendix D Published as:

- Chriss Grimholt and Sigurd Skogestad. Optimization of oil field production under gas coning conditions using the optimal closed-loop estimator. In *2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*, pages 39–44, May 2015b

Chapter Two

Optimization of fixed-order controllers using exact gradients

Finding good controller settings that satisfy complex design criteria is not trivial. This is also the case for simple fixed-order controllers including the three parameter PID controller. To be rigorous, we formulate the design problem into an optimization problem. However, the algorithm may fail to converge to the optimal solution because of inaccuracies in the estimation of the gradients needed for this optimization. In this chapter we derive exact gradients for the problem of optimizing performance (IAE for input and output disturbances) with constraints on robustness (M_S, M_T). Exact gradients, give increased accuracy and better convergence properties than numerical gradients, including forward finite differences. The approach may be easily extended to other control objectives and other fixed-order controllers, including Smith Predictor and PIDF.

2.1 Introduction

The background for this chapter was our efforts to find optimal PID controllers for first-order plus time delay (FOPTD) processes. The PID controller may be expressed as

$$K_{\text{PID}}(s;p) = k_p + k_i/s + k_d s \tag{2.1}$$

where k_p, k_i, and k_d are the proportional, integral, and derivative gain, respectivly. The objective was to minimize for the simple feedback system in Figure 2.1, the IAE (time domain)

$$\min_K \quad \text{IAE} = \int_0^\infty \left|e(t)\right| \mathrm{d}t, \tag{2.2}$$

for given disturbances subject to achieving a given frequency domain robustness M_{ST}

$$M_{\text{ST}} = \max\{M_{\text{S}}, M_{\text{T}}\} \leq M^{ub}. \tag{2.3}$$

where M_{S} and M_{T} are the peaks of the sensitivity and complementary sensitivity functions, respectively. In general, this is a non-convex optimization problem which we initially solved using standard optimization software in MATLAB. We used gradient-free optimization like the Nelder-Mead Simplex method (Nelder and Mead, 1965), similar to the work of Garpinger and Hägglund (2008), and used SIMC settings (Skogestad, 2003) as initial values for the PID parameters. However, the gradient-free method was slow and unreliable for our purpose of generating trade-off curves between IAE and M_{ST}, which involves repeated optimizations with changing M_{ST}-values (Grimholt and Skogestad, 2013).

We achieved significant speedup by switching to gradient-based methods (`fmincon` in MATLAB), where the gradients of the cost func-

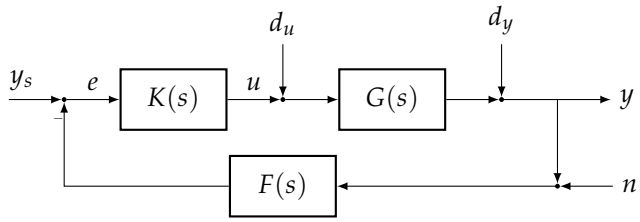

Figure 2.1: Block diagram of the closed loop system. In this chapter we assume $y_s = 0$, $n = 0$, and $F(s) = 1$. $K(s)$ is the feedback controller, $G(s)$ is the process, and $F(s)$ is the measurement filter (part of the controller). with controller $K(s)$ and plant $G(s)$.

tion (J) and the constraints (M_{ST}) with respect to the controller parameters were found numerically using finite differences. Our experience with this approach were fairly good, but quite frequently, for example for small values of M_{ST}, it did not converge to a solution. Surprisingly, this also occurred even though the initial guess was close to the optimum. It turned out that the main problem was not the non-convexity of the problem or the possibility for local minima, but rather inaccuracies in the estimation of the gradients when using finite-differences.

This led us to derive exact (analytical) expressions for the gradients of IAE and $M_{ST} \leq M^{ub}$. This approach is based on the chain rule, that is, first we derive the gradient of IAE with respect to the control error $e(t)$, then the gradient of $e(t)$ with respect to the controller $K(s)$, and then the gradient of $K(s)$ with respect to the controllers parameters p. Some of the gradients are derived in the Laplace domain, but they are evaluated in the time domain, based on a delayed state space realization.

The derivation of these gradients is the main part of the chapter

(Section 2.4). Our experience with exact gradients has been very good, and we achieve convergence for a much wider range of conditions in terms of constraints (M_{ST}) and models, see Section 2.4. In addition, the approach can easily be extended to other fixed-order controller (e.g., PIDF controller, Smith Predictor), and to other process models. This is discussed in Section 2.5. Note that the gradient of IAE is not calculated using direct sensitivity results (Biegler, 2010), but calculated through Laplace transforms. This greatly simplifies the derivation and the numerical results are good.

A preliminary version of this results was presented at the PSE2015-ESCAPE25 conference (Grimholt and Skogestad, 2015a). The main contribution compared to the conference paper is to provide a derivation of the gradients and focus on general fixed-order controllers instead of just PID.

2.2 Previous related work on optimal PID tuning

The simple three-parameter PID controller is the most common controller in the process industry. However, finding good tuning parameter by trial and error is generally difficult and also time consuming. Alternatively, parameters may be found for by use of tuning rules (e.g. Ziegler and Nichols (1942), and the SIMC-rules of Skogestad (2003)). Nevertheless, when the complexity of the design increases, it is beneficial to switch to optimization-based design. This approach can handle complex process models, non-standard controller parameterizations, and special requirements on controller performance and robustness.

The first systematic contribution on PID tuning is the famous ZN tuning rule published in 1942 under the title *Optimum Setting for Automatic-Controllers*. However, the "optimum" PID settings were

manly derived from visual inspection of the closed-loop response, aiming at disturbance attenuation with a quarter decay ratio, which results in a quite oscillatory and aggressive response for most process control application.

Hall (1943) proposed finding optimal controller settings by minimizing the integrated squared error ($\text{ISE} = \int e^2 \, \mathrm{d}t$). The ISE criteria is generally selected because it has nice analytical properties. By minimizing ISE, Hazebroek and Van der Waerden (1950) analyzed the ZN tuning rule, and proposed an improved rule. The authors noted that minimizing the ISE criterion could give rise to large fluctuations in the manipulated variable, and that the allowable parameters must be restricted for these processes. The analytical treatment of the ISE-optimization was further developed in the influential book by Newton et al. (1957).

The first appearance of a "modern" optimization formulation, which includes a performance vs. robustness trade-off, similar to the one used in this chapter, is found in Balchen (1958). Balchen minimizes the integrated absolute error ($\text{IAE} = \int |e| \, \mathrm{d}t$) while limiting the peak of the sensitivity function (M_s). The constraint on M_s may be adjusted to ensure a given relative dampening or stability margin. This is similar to the formulation used much later in Kristiansson and Lennartson (2006). In addition, Balchen mentions the direct link between minimizing integrated error ($\text{IE} = \int e \, \mathrm{d}t$) and maximizing the integral gain (k_i) for a unit step input disturbance, as given by the relationship $\text{IE} = 1/k_i$. Though the paper of Balchen is very interesting, it seems to have been largely overlooked by the scientific community.

Schei (1994) followed up this idea and derived PI settings by maximizing the integral gain k_i subject to a given bound on the sensitivity function M_s. Åström et al. (1998) formulated this optimization

problem as a set of algebraic equations which could be efficiently solved. In Panagopoulos et al. (2002) the formulation was extended to PID control, and a fast concave-convex optimization algorithm is presented in Hast et al. (2013). However, quantifying performance in terms of IE can lead to oscillatory response, especially for PID controllers, because it does not penalize oscillations. Shinskey (1990) argued that the IAE is a better measure of performance, and IAE is now widely adopted in PID design (Ingimundarson and Hägglund, 2002; Åström and Hägglund, 2006; Skogestad, 2003; Huba, 2013; Garpinger et al., 2014; Alfaro et al., 2010; Alcántara et al., 2013).

2.3 Problem Formulation

Feedback system

We consider the linear feedback system in Figure 2.1, with disturbances entering at both the plant input (d_u) and plant output (d_y). Because the response to setpoints can always be improved by using a two degree-of-freedom (DOF) controller, we do not consider setpoints (y_s) in the initial design of the feedback controller. That is, we assume $y_s = 0$. It is worth noticing that, from a feedback point of view, a disturbance entering at the plant output (d_y) is equivalent to a setpoint change. Measurement noise (n) enters the system at the measured output (y). This system can be represented by four transfer functions, nicknamed "the gang of four",

$$S(s) = \frac{1}{1 + G(s)\,K(s)}, \qquad T(s) = 1 - S(s),$$

$$GS(s) = G(s)\,S(s), \qquad KS(s) = K(s)\,S(s).$$

Their effect on the control error and plant input is,

$$-e = y - y_s = S(s)\, d_y + GS(s)\, d_u - T(s)\, n, \tag{2.4}$$

$$-u = KS(s)\, d_y + T(s)\, d_u + KS(s)\, n. \tag{2.5}$$

Although we could use any fixed-order controller, we consider in this chapter mostly the parallel (linear in the parameter) PID controller,

$$K_{\text{PID}}(s; p) = k_p + k_i/s + k_d s, \tag{2.6}$$

$$p = \begin{pmatrix} k_p & k_i & k_d \end{pmatrix}^T, \tag{2.7}$$

where k_p, k_i, and k_d is the proportional, integral, and derivative gain, respectively. The controller can equivalently be written in the form

$$K_{\text{PID}}(s; p) = k_c \left(1 + \frac{1}{\tau_i s} + \tau_d s\right), \tag{2.8}$$

where $k_c = k_p$, $\tau_i = k_c/k_i$, and $\tau_d = k_d/k_c$ is the proportional gain, integral time, and derivative time, respectively. Note that for $\tau_i < 4\tau_d$, the parallel PID controllers has complex zeros. We have observed that this can result in several peaks or plateaux for the magnitude of sensitivity function in the frequency domain $|S(j\omega)|$. As shown later, this becomes important when adding robustness specifications on the frequency behavior.

Performance

We quantify controller performance in terms of the integrated absolute error (IAE),

$$\text{IAE}\,(p) = \int_0^\infty |e(t; p)|\ \mathrm{d}t, \tag{2.9}$$

when the system is subject to step disturbances. We include both input (d_u) and output (d_y) disturbances and choose the weighted cost function

$$J(p) = 0.5 \left(\varphi_{dy} \, \text{IAE}_{dy}(p) + \varphi_{du} \, \text{IAE}_{du}(p) \right) \tag{2.10}$$

where φ_{dy} and φ_{du} are normalization factors. It is necessary to normalize the resulting IAE_{du} and IAE_{dy}, to be able to compare the two terms in the cost function (2.10), and it is ultimately up to the user to decide which normalization method is most appropriate. Numerically the exact value of these variables are not very important, and in this chapter we have, similar to previous work (Grimholt and Skogestad, 2012a, 2013), selected the normalisation factors to be the inverse of the optimal IAE values for reference controllers (e.g. PI, PID) tuned for a step change on the input (IAE°_{du}) and output (IAE°_{dy}), respectively.

$$\varphi_{du} = \frac{1}{\text{IAE}^{\circ}_{du}} \quad \text{and} \quad \varphi_{dy} = \frac{1}{\text{IAE}^{\circ}_{dy}}.$$

This normalization is similar to the one used in Shinskey (1990). To ensure robust reference controllers, they are required to have $M_S = M_T = 1.59$ [*]. Note that two different reference controllers are used to obtain the IAE° values, whereas a single controller $K(s; p)$ is used to find $\text{IAE}_{dy}(p)$ and $\text{IAE}_{du}(p)$, when minimizing the cost function $J(p)$ in (2.10).

Robustness

In this chapter, we have chosen to quantify robustness in terms of the largest sensitivity peak, $M_{ST} = \max\{M_S, M_T\}$ (Garpinger and

[*] For those that are curious about the origin of this specific value $M_S = 1.59$, it is the resulting M_S value for a SIMC tuned PI controller with $\tau_c = \theta$ on FOPTD process with $\tau \leq 8\theta$.

Hägglund, 2008), where

$$M_S = \max_\omega |S(j\omega)| = \|S(j\omega)\|_\infty,$$
$$M_T = \max_\omega |T(j\omega)| = \|T(j\omega)\|_\infty,$$

and $\|\cdot\|_\infty$ is the H_∞ norm (maximum peak as a function of frequency).

For stable process, M_S is usually larger than M_T. In the Nyquist plot, M_S is the inverse of the closest distance to the critical point $(-1, 0)$ for the loop transfer function $L(s) = G(s)\,K(s)$. For robustness, a small M_S-value is desired, and generally M_S should not exceed 2. A reasonable M_S-value is about 1.6, and notice that $M_S < 1.6$ guarantees the following good gain and phase margins: GM> 2.67 and PM$> 36.4^\circ$ (Rivera et al., 1986).

From our experience, using the sensitivity peak as a single constraint $\big|S(j\omega)\big| \leq M^{ub}$ for all ω can lead to poor convergence. This is because the optimal controller can have several peaks of equal magnitude at different frequencies (see Figure 2.3), and the optimizer may jump between peaks during iterations. Each peak has a different gradient with respect to the PID parameters. To avoid this problem, instead of using a single constraint, we use multiple constraints obtained by griding the frequency response,

$$\big|S(j\omega)\big| \leq M^{ub} \qquad \text{for all } \omega \text{ in } \Omega, \tag{2.11}$$

where $\Omega = [\omega_1, \omega_2, \ldots, \omega_n]$ is a finite set of frequency points. This gives one inequality constraint for each grid frequency. In addition to handling multiple peaks, this approximation also improves convergence for *infeasible* initial controllers because more information is supplied to the optimizer. On the downside, the approximation results in a somewhat reduced accuracy and an increased computational load; However, we found that the benefit of improved convergence makes up for this.

Summary of the optimization problem

In summary, the optimization problem can be stated as follows,

$$\underset{p}{\text{minimize}} \quad J(p) = 0.5 \left(\varphi_{dy} \, \text{IAE}_{dy}(p) + \varphi_{du} \, \text{IAE}_{du}(p) \right) \tag{2.12}$$

$$\text{subject to} \quad c_{\text{S}}(p) = \left| S(j\omega;\, p) \right| - M_{\text{S}}^{ub} \leq 0 \quad \text{for all } \omega \text{ in } \Omega \tag{2.13}$$

$$c_{\text{T}}(p) = \left| T(j\omega;\, p) \right| - M_{\text{T}}^{ub} \leq 0 \quad \text{for all } \omega \text{ in } \Omega, \tag{2.14}$$

where M_{S}^{ub} and M_{T}^{ub} are the upper bound on $\left|S(j\omega)\right|$ and $\left|T(j\omega)\right|$, respectively. Typically, we select $M^{ub} = M_{\text{S}}^{ub} = M_{\text{T}}^{ub}$. From experience, selecting Ω as 10^4 logarithmically spaced frequency points with a frequency range from $0.01/\theta$ to $100/\theta$, where θ is the effective time delay of the process. If there is a trade-off between performance and robustness, at least one robustness constraints in (2.13) or (2.14) will be active.

A simple pseudo code for the cost function is shown in Algorithm 1 in Appendix A.1. It is important that the cost function also returns the error responses, such that they can be reused for the gradient. The pseudo code for the constraint function is shown in Algorithm 2 in Appendix A.1 The intention behind the pseudo codes is to give an overview of the steps involved in the calculations.

2.4 Gradients

The gradient of a function $f(p)$ with respects to a parameter vector p is defined as

$$\nabla_p f(p) = \begin{pmatrix} \frac{\partial f}{\partial p_1} & \frac{\partial f}{\partial p_2} & \cdots & \frac{\partial f}{\partial p_{n_p}} \end{pmatrix}^T, \tag{2.15}$$

where n_p is the number of parameters. In this chapter, p_i refers to parameter i, and the partial derivative $\frac{\partial f}{\partial p_i}$ is called the sensitivity of f. For simplicity we will use short hand notation $\nabla \equiv \nabla_p$.

Forward finite differences

The sensitivities can be approximated by forward finite differences (FFD)

$$\frac{\partial f}{\partial p_i} \approx \frac{f(p_i + \Delta p_i) - f(p_i)}{\Delta p_i}, \tag{2.16}$$

which require $(1 + n_p)$ perturbations. Because we consider both input and output disturbances, this results in at total of $2(1 + n_p)$ time response simulations.

One of the dominating factors affecting the accuracy of the finite difference approximation of the sensitivities is the accuracy of the time response simulation. This can easily be seen in the following example. The computed IAE-value ($\widetilde{\text{IAE}}$) we get from integrating (2.9) can be written into the true IAE and the integration error δ.

$$\widetilde{\text{IAE}} = \text{IAE} \pm \delta \tag{2.17}$$

Using forward finite differences (FFD) (2.16), and assuming for simplicity only one parameter p, the computed gradient becomes

$$\nabla\widetilde{\text{IAE}} = \frac{\text{IAE}_1 \pm \delta_1 - \text{IAE}_2 \pm \delta_2}{\Delta p} + O(\Delta p) = \frac{\Delta \text{IAE}}{\Delta p} + \frac{\Delta \delta}{\Delta p} + O(\Delta p) \tag{2.18}$$

where Δp is the perturbation in parameter, and $O(\Delta p)$ is the truncation error. The worst-case gradient error becomes

$$E_{\text{FFD}} = \frac{2\delta}{\Delta p} + O(\Delta p). \tag{2.19}$$

As the perturbation size $\Delta p \to 0$, the truncation error $O(\Delta p) \to 0$. However, the simulation error is magnified as $2\delta/\Delta p \to \infty$. On the other hand, the impact of simulation error can be reduced by increasing the perturbation size Δp, but then truncation error might became an issue. It this chapter we have chosen to use MATLAB's default perturbation size of $\sqrt{\epsilon}$, where ϵ is the machine precession. In order to derive simple expressions which are less prone to errors, we will instead use the gradient expressions derived below.

Cost function gradient

The gradient of the cost function $J(p)$ can be expressed as

$$\nabla J(p) = 0.5 \left(\varphi_{dy} \, \nabla \text{IAE}_{dy}(p) + \varphi_{du} \, \nabla \text{IAE}_{du}(p) \right) \tag{2.20}$$

The IAE sensitivities are difficult to evaluate analytically, but they can be found in a fairly straightforward manner by expressing them such that the integrals can be evaluated numerically.

By taking advantage of the fixed structure of the problem, we develop general expressions for the gradient. When the parameter sensitivity of the controller $K(s)$ is found, evaluating the gradient is a simple process of combining and evaluating already defined transfer functions. This enables the user to quickly find the gradients for a linear system for any fixed-order controller $K(s)$.

From the definition of the IAE in (2.9) and assuming that $|e(t)|$ is smooth, that is that $|e(t)|$ and $\text{sign}\{e(t)\} \nabla e(t)$ are continuous, the sensitivity of the IAE can be expressed as (see Appendix A.2 for

details)

$$\nabla \text{IAE}_{dy}(p) = \int_0^{t_f} \text{sign}\left\{e_{dy}(t)\right\} \nabla e_{dy}(t)\, \mathrm{d}t, \tag{2.21}$$

$$\nabla \text{IAE}_{du}(p) = \int_0^{t_f} \text{sign}\left\{e_{dy}(t)\right\} \nabla e_{dy}(t)\, \mathrm{d}t. \tag{2.22}$$

Introducing the Laplace transform, we see from (2.4) that

$$e_{dy}(s) = S(s)\, d_y \quad \text{for output disturbances and} \tag{2.23}$$

$$e_{du}(s) = GS(s)\, d_u \quad \text{for input disturbances.} \tag{2.24}$$

By using the chain rule (Åström and Hägglund, 2006), we can write the error sensitivities as a function of the parameter sensitivity of the controller $K(s)$ (See Appendix A.2)

$$\nabla e_{dy}(s) = -GS(s)\; S(s)\; \nabla K(s)\, d_y \tag{2.25}$$

$$\text{and} \quad \nabla e_{du}(s) = -GS(s)\; GS(s)\; \nabla K(s)\, d_u, \tag{2.26}$$

For the PID controller defined in (5.2), the controller sensitivities $\nabla K(s)$ with respects to the parameters $p = \begin{pmatrix} k_p & k_i & k_d \end{pmatrix}^T$ are.

$$\nabla K_{\text{PID}}(s) = \begin{pmatrix} \frac{\partial K_{\text{PID}}}{\partial k_p} & \frac{\partial K_{\text{PID}}}{\partial k_i} & \frac{\partial K_{\text{PID}}}{\partial k_d} \end{pmatrix}^T = \begin{pmatrix} 1 & 1/s & s \end{pmatrix}^T \tag{2.27}$$

To obtain $\nabla \text{IAE}_{dy}(p)$ and $\nabla \text{IAE}_{du}(p)$ in (2.21) and (2.22), we must obtain the inverse transforms of $\nabla e_{dy}(s)$ and $\nabla e_{du}(s)$ in (2.25) and (2.26). However, for processes $G(s)$ with time delay (which is the case for our problem), the sensitivity $S(s)$ will have internal delays (delays in the denominator), and there are no analytical solution to the inverse Laplace transforms of $\nabla e_{du}(s)$ and $\nabla e_{dy}(s)$, and they must be evaluated numerically. For example, to evaluate the gradient of IAE_{du} in (2.22) for PID control when considering a unit step disturbance ($d_u = 1/s$), we first obtain the time response of $\nabla e_{dy}(t)$

by performing an impulse response simulation of a state space realization of the following system,

$$\nabla e_{du}(s) = -GS(s)\ GS(s) \begin{pmatrix} 1/s & 1/s^2 & 1 \end{pmatrix}^T. \tag{2.28}$$

Typical numerical results for ∇e_{du} are shown in the lower plot of Figure 2.4. The gradient of the IAE is then calculated by evaluating the IAE integral (2.22) using numerical integration techniques like the trapezoidal method.

In many cases, $|e(t)|$ and sign $\{e(t)\}\ \nabla e(t)$ are not continuous on the whole time range. For example, $e(t)$ will have a discrete jump for step setpoint changes. For our assumptions to be valid in such cases, the integration must be split up into subintervals which are continuous. Violating this assumption will result in an inaccuracy in the calculation of the gradient at the time step of the discontinuity. However, by using very small integration steps in the time response simulations, this inaccuracy is negligible. Therefore, the integration has not been split up into subintervals for the case study in Section 2.4.

A simple pseudo code for the gradient calculation is shown in Algorithm 3 in Appendix A.1. Notice that the error responses from the cost function are reused (shown as inputs).

Because the gradient is evaluated by time domain simulations, the method is limited to processes that gives proper gradient transfer functions. If the gradient transfer functions are not proper, a small filter can be added to the process, controller or the gradient transfer function to make it proper. However, this will introduce small inaccuracies.

To obtain the gradient of the cost function (2.20), $2n_p$ simulations are needed for evaluating (2.25) and (2.26), and 2 simulations are needed to evaluate the error (2.23) and (2.24) , resulting in a total

of $2(1+n_p)$ simulations. This is the same number of simulations needed for the one-sided forward finite differences approximation in (2.16), but the accuracy is much better.

Constraint gradients

The gradient of the robustness constraints, $c_{\text{S}}(p)$ and $c_{\text{T}}(p)$, can expressed as (See Appendix A.3)

$$\nabla c_{\text{S}}(j\omega;p) = \nabla|S(j\omega)| = \frac{1}{|S(j\omega)|}\Re\left\{S^*(j\omega)\nabla S(j\omega)\right\} \qquad \text{for all } \omega \text{ in } \Omega \tag{2.29}$$

$$\nabla c_{\text{T}}(j\omega;p) = \nabla|T(j\omega)| = \frac{1}{|T(j\omega)|}\Re\left\{T^*(j\omega)\nabla T(j\omega)\right\} \qquad \text{for all } \omega \text{ in } \Omega \tag{2.30}$$

The asterisk $(*)$ is used to indicate the complex conjugate. By using the chain rule, we can rewrite the constraint gradient as an explicit function of $\nabla K(j\omega)$ (as we did with the cost function gradient),

$$\nabla S(j\omega) = -GS(j\omega)\; S(j\omega)\; \nabla K(j\omega) \tag{2.31}$$

$$\nabla T(j\omega) = \nabla\left(1 - S(j\omega)\right) = -\nabla S(j\omega) \tag{2.32}$$

The gradients of the constraints is then evaluated at each frequency point ω in Ω. The constraint gradients are algebraic functions of p, and can also be found using automatic differentiation. However, due to the fixed structure of the gradients they are more easily obtained by multiplication of the known transfer function and $\nabla K(j\omega)$. A pseudo code for the gradient calculation is show in Algorithm 4 in Appendix A.1.

Table 2.1: Comparison of optimal solutions when using different combinations of gradients for the problem given in (2.33)–(2.36).

Gradient type		Cost function	Optimal parameters			number of
Cost-function	Constraints	$J(p^\star)$	k_p	k_i	k_d	iterations
exact	exact	**2.0598**	0.5227	0.5327	0.2172	13
fin.dif.	exact	2.1400	0.5204	0.4852	0.1812	16
exact	fin.dif.	**2.0598**	0.5227	0.5327	0.2172	13
fin.dif.	fin.dif.	2.9274	0.3018	0.3644	0.2312	11

Case study

The exact gradients were implemented and computed for PID control of the design problem given in (2.12)–(2.14), with a FOPTD process

$$G(s) = \frac{e^{-s}}{s+1}. \tag{2.33}$$

To make the system proper, we also added a fist-order filter to the controller

$$K(s) = K_{\text{PID}}(s)\frac{1}{\tau_f s + 1}, \quad \text{with} \quad \tau_f = 0.001. \tag{2.34}$$

To ensure robust controllers, the sensitivity peaks are required to be small with

$$M_{\text{S}}^{ub} = M_{\text{T}}^{ub} = 1.3. \tag{2.35}$$

This problem has the following normalization factors

$$\text{IAE}_{dy}^{\circ} = 1.56, \quad \text{IAE}_{du}^{\circ} = 1.42, \tag{2.36}$$

and the optimization algorithm was started with controller parameters

$$p_0 = \begin{pmatrix} 0.2 & 0.02 & 0.3 \end{pmatrix}^T. \tag{2.37}$$

This case study was selected because it is a problem that looks simple, but can be surprisingly hard to solve. One reason for this is that the optimal solution has two equal M_s peaks (Figures 2.2 and 2.3), and can therefore exhibit cyclic behavior between the iterations when using a single $M_{sT} \leq M^{ub}$ constraint. Actually, if it was not for the filter, the optimal problem solution would have an infinite number of equal peaks.

The error response (obtained with a simulation length of 25 time units) and and cost function sensitivity was found by fixed step integration (with number of steps $n_{\text{steps}} = 10^4$). The problem was solved using MATLAB's `fmincon` with the active set algorithm. The optimal error response with sensitivities are shown in Figure 2.4. For comparison, the problem was also solved with approximated gradients using forward finite differences with MATLAB's default perturbation size of $\sqrt{\epsilon}$ (which is supposed to give a good balance between truncation error and round-off error).

As seen in Table 2.1, the exact gradients performed better than the approximated finite differences. The biggest improvement comes from the using the exact cost function gradients. The large difference in controller parameters between the two approaches indicates that the optimum is relatively flat, as the small numerical inaccuracies in the simulations and small truncation errors in the finite difference approximation gives gradients that falsely satisfy the tolerances of the algorithm.

As mentioned previously, one dominating factor affecting the accuracy of the finite difference approximation, is the accuracy of the time reprocess simulation. To illustrate this, the same design problem was solved with with different numbers of time steps during the integration. Even with $n_{\text{steps}} = 10^5$, the forward finite differences failed to converge to the optimum. On the other hand,

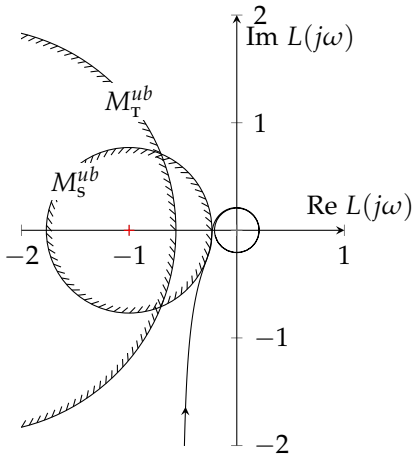

Figure 2.2: Nyquist plot of $L(j\omega)$ for the optimal controller for the problem given in (2.33)–(2.36).

the exact gradient could still converge to the local optimum with $n_{\text{steps}} = 500$.

The exact gradient converged for most initial guesses that gave a stable closed-loop system. On the other hand, forward finite differences failed to find the optimum even when starting very close to the minimum, for example

$$p_0 = \begin{pmatrix} 1.001p_1^\star & p_2^\star & p_3^\star \end{pmatrix}^T.$$

With central finite differences, the accuracy was may be increased, but this requires $2(1 + 2n_p)$ step simulations.

2.5 Extensions to other fixed order controllers

As stated previously, the method is easily extended to other linear fixed-order controllers, as it only requires changing the expression

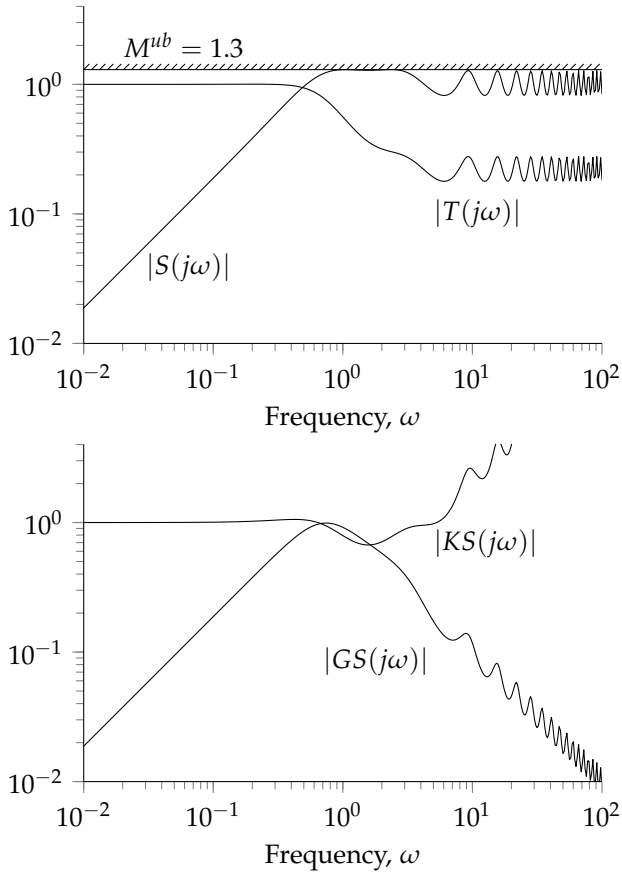

Figure 2.3: Frequency response of the "the gang of four" ($S(j\omega)$, $T(j\omega)$, $GS(j\omega)$, and $KS(j\omega)$) for the optimal controller for the problem given in (2.33)–(2.36).

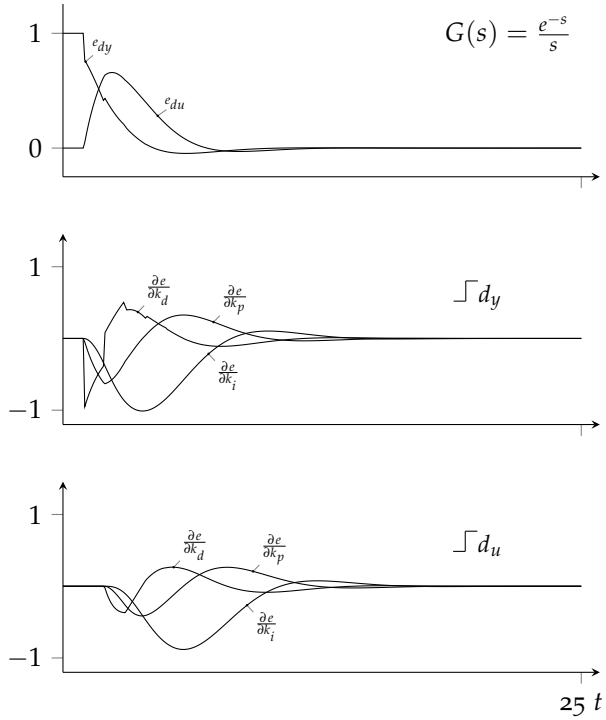

Figure 2.4: Optimal error response and corresponding error sensitivities for the problem given in (2.33)–(2.36).

for the parameter sensitivity $\nabla K(s)$ in (2.27). Here we will give the $\nabla K(s)$ for three other controllers: the serial PID controller, the PIDF controller, and the Smith predictor.

Serial PID controller

For the serial PID controller,

$$K_{\text{PID}}^{\text{serial}}(s) = k_c \frac{(\tau_i s + 1)(\tau_d s + 1)}{\tau_i s}, \tag{2.38}$$

the controller sensitivities are (elements in $\nabla K_{\text{PID}}^{\text{serial}}(s)$)

$$\frac{\partial K_{\text{PID}}^{\text{serial}}(s)}{\partial k_c} = \frac{(\tau_i s + 1)(\tau_d s + 1)}{\tau_i s}, \tag{2.39}$$

$$\frac{\partial K_{\text{PID}}^{\text{serial}}(s)}{\partial \tau_i} = -k_c \frac{(\tau_d s + 1)}{{\tau_i}^2 s}, \tag{2.40}$$

$$\frac{\partial K_{\text{PID}}^{\text{serial}}(s)}{\partial \tau_d} = k_c \frac{(\tau_i s + 1)}{\tau_i}. \tag{2.41}$$

Note the serial controller sensitivities must be updated during iterations, whereas they are constant for the parallel PID controller $K_{\text{PID}}(s)$, see (2.27)

PIDF controller

For a PIDF controller, here defined as

$$K_{\text{PIDF}}(s) = K_{\text{PID}}(s)F(s), \tag{2.42}$$

where the parameters in the filter $F(s)$ are extra degrees of freedom, the derivative can be expressed in terms of product rule,

$$\nabla K_{\text{PIDF}}(s) = F(s)\nabla K_{\text{PID}}(s) + K_{\text{PID}}(s)\nabla F(s), \tag{2.43}$$

For example, with a first-order filter

$$F(s) = \frac{1}{\tau_f s + 1} \quad \text{we get} \quad \nabla F(s) = \begin{pmatrix} 0 & 0 & 0 & -\frac{s}{(\tau_f s+1)^2} \end{pmatrix}^T. \tag{2.44}$$

Note that this PIDF controller with four tuning parameters is the most general second-order controller $K(s)$, given that the controller must have integral action.

Smith predictor

The Smith predictor (SP) uses an internal model of the process with time delay, $G(s)$. Let $G_\circ(s)$ be the process model without delay and assume that a PID controller is used for the delay process, $K_{\text{PID}}(s) = k_p + k_i/s + k_d s$. The Smith predictor becomes,

$$K_{\text{SP}}(s) = \frac{K_{\text{PID}}(s)}{1 + K_{\text{PID}}(s)(G_\circ(s) - G(s))}, \tag{2.45}$$

and the gradients are

$$\nabla K_{\text{SP}}(s) = \frac{\partial K_{\text{SP}}(s)}{\partial K_{\text{PID}}(s)} \nabla K_{\text{PID}} = \frac{\nabla K_{\text{PID}}}{\left[1 + K_{\text{PID}}(s)(G_\circ(s) - G(s))\right]^2}. \tag{2.46}$$

Where $\nabla K_{\text{PID}} = \begin{pmatrix} 1 & 1/s & s \end{pmatrix}^T$ is the gradient of the internal PID controller.

2.6 Discussion

Input usage and Noise filtering

We have not included a bound on input usage, e.g. by considering the maximum peak of $\left|KS(j\omega)\right|$ which is $\left\|KS\right\|_\infty$. For simple cases, input usage can be reduced by simply making the process more robust by lowering M_{ST} (Grimholt and Skogestad, 2012a). That is, reducing the controller gain will make the controller more robust *and* reduce input usage. However, for unstable processes this will not hold, and increasing the controller gain can actually make the controller *more* robust and increase input usage. For such processes, an additional constraint should be added to limit input usage to a desired levels.

For the parallel PID controller in (2.8), the gain and thus noise amplification goes to infinity at high frequencies. To avoid excessive input movements, a measurement filter may be added as we did in (2.34). We assume in this chapter that noise amplification is addressed separately after the initial controller design. Nevertheless, our design formulation could easily be extended to handle noise filtering more explicitly by using an appropriate constraint, e.g. $\|KS\|_\infty$. This is treated in a separate paper by Soltesz et al. (2017a). If the measurement filter actually enhances *noise-less* performance, the note that we are no longer talking about a PID controller, but a PIDF controller with four adjustable parameters.

Circle constraint

Circle constraints (Åström and Hägglund, 2006) provides an alternative to constraining $S(j\omega)$ and $T(j\omega)$. The idea is to ensure that the loop function $L(j\omega) = G(s)K(s)$ is outside given robustness circles in the Nyquist plot. For a given circle with center C and radius R, the robustness criteria can be expressed as

$$\left|C - L(j\omega)\right|^2 \geq R^2 \qquad \text{for all } \omega \text{ in } \Omega. \tag{2.47}$$

For $S(s)$, the centre and radius will be

$$C = -1 \quad \text{and} \quad R = 1/M^{ub}. \tag{2.48}$$

For $T(s)$, the centre and radius will be

$$C = -\frac{(M^{ub})^2}{(M^{ub})^2 - 1} \quad \text{and} \quad R = \frac{M^{ub}}{(M^{ub})^2 - 1}. \tag{2.49}$$

Written in standard form, the constraint becomes,

$$c_{\text{L}}(p) = R^2 - \left|C - L(j\omega)\right|^2 \leq 0 \qquad \text{for all } \omega \text{ in } \Omega, \tag{2.50}$$

with center C and radius R for the corresponding M^{ub} circle, respectively. The corresponding gradient is

$$\nabla c_{\text{L}}(j\omega) = 2\Re\left\{\left(C - L(j\omega)\right)^* \nabla L(j\omega)\right\} \qquad \text{for all } \omega \text{ in } \Omega. \quad (2.51)$$

The main computational cost is the evaluation of the transfer functions. It may seen that the circle constraint (2.51) has an advantage, because you only need to evaluate $L(j\omega)$, whereas, both $S(j\omega)$ and $T(j\omega)$ must be evaluated for the constraints (2.29) and (2.30) . However, by using the relation $S + T = 1$ we can rewrite e.g. $T(j\omega)$ in terms of $S(j\omega)$ by $T = 1 - S$. Thus, we only need to evaluate $S(j\omega)$ to calculate both c_{S} and c_{T}. Because the mathematical operations are relatively cheap, the two gradients types are almost equivalent.

Direct sensitivity calculations

Our method is closely related to the more general direct sensitivity method, as defined in Biegler (2010), used for optimal control problems, which we applied in Jahanshahi et al. (2014). However, processes with time delay we cannot use conventional ordinary differential equations (ODE) sensitivity results as the resulting feedback system has internal delay. To apply the direct sensitivity method to our problem, we set up the delayed differential equations (DDE) symbolically, found the parameter sensitivities of the states, and integrated using a integrator for delayed differential equations (e.g. `dde23`). Although the direct sensitivity approach works well on this problem, a lot of work went into setting up the DDE equations and sensitivities, and integrating them in MATLAB. By taking advantage the fixed structure of the problem and the control system toolbox in MATLAB, it is less work to use the approach presented in this chapter.

2.7 Conclusion

In this chapter we have successfully applied the exact gradients for a typical performance (IAE) with constrained robustness M_{ST} optimization problem. Compared to gradients approximated by forward finite difference, the exact gradients improved the convergence to the true optimal. By taking advantage of the fixed structure of the problem, the exact gradients were presented in such a way that they can easily be implemented and extended to other fixed-order controllers. When the parameter sensitivity of the controller $K(s)$ is found, evaluating the gradient is just a simple process of combining and evaluating already defined transfer functions. This enables the user to quickly find the gradients for a linear system for any fixed order controller. The MATLAB code for the optimization problem is available at the home page of Sigurd Skogestad.

Chapter Three

Optimal PI and PID control of first-order plus delay processes and evaluation of the original and improved SIMC rules

The first-order plus delay process model with parameters k (gain), τ (time constant) and θ (delay) is the most used representation of process dynamics. This chapter has three objectives. First, we derive optimal PI- and PID-settings for this process. Optimality is here defined as the minimum Integrated Absolute Error (IAE) to disturbances for a given robustness level. The robustness level, which is here defined as the sensitivity peak (M_s), may be regarded as a tuning parameter. Second, we compare the optimal IAE-performance with the simple SIMC-rules, where the SIMC tuning parameter τ_c

is adjusted to get a given robustness. The "original" SIMC-rules give a PI-controller for a first-order with delay process, and we find that the SIMC PI-controller is close to the optimal PI-controller for most values of the process parameters (k, τ, θ). The only exception is for delay-dominant processes where the SIMC-rule gives a pure integrating controller. The third objective of this chapter is to propose and study a very simple modification to the original SIMC-rule, which is to add a derivative time $\tau_d = \theta/3$ (for the serial PID-form). This gives performance close to the IAE-optimal PID also for delay-dominant processes. We call this the "improved" SIMC-rule, but we put "improved" in quotes, because this controller requires more input usage, so in practice the original SIMC-rule, which gives a PI-controller, may be preferred.

3.1 Introduction

The PID controller is by far the most common controller in industrial practice. However, although it has only three parameters, it is not easy to tune unless one uses a systematic approach. The first PID tuning rules were introduced by Ziegler and Nichols (1942). Although some other empirical rules were suggested, the Ziegler-Nichols (ZN) rules remained for about 50 years as the best and most commonly used rules. However, there are at least three problems with the ZN-rules:

1. The ZN-settings are rather aggressive for most processes with oscillations and overshoots.
2. The ZN-rule contains no adjustable tuning parameter to adjust the robustness and make it less aggressive.
3. For a pure time delay process, the ZN-PID settings give instability and the ZN-PI settings give very poor performance (also see

discussion section).

For many years there was almost no academic interest in revisiting the PID controller to obtain better tuning rules. Dahlin (1968) considered discrete-time controllers and introduced the idea of specifying the desired closed-loop response and from this backing out the controller parameters. Typically, a first-order response is specified with closed-loop time constant τ_c (called λ by Dahlin). Importantly, τ_c (or λ) is a single tuning parameter which the engineer can use to specify how aggressive the controller should be. For first or second-order plus delay processes, the resulting controller can be approximated by a PID controller. This idea is also the basis of the Internal Model Control (IMC) PID-controller of Rivera et al. (1986) which results in similar PID tuning rules. The IMC PI-tuning rules, also known as lambda tuning, became widely used in the pulp and paper industry around 1990 (Bialkowski, 1996).

However, the Dahlin and IMC rules set the controller integral time equal to the dominant process time constant (τ_i=τ) and this means that integral action is effectively turned off for "slow" or "integrating" processes with a large value of τ. This may be acceptable for setpoint tracking, but not for load disturbances, that is, for disturbances entering at the plant input. This led Skogestad (2003) to suggest the SIMC rule where τ_i is reduced for processes with large time constants. However, to avoid slow oscillations it should not be reduced too much, and this led to the SIMC-rule $\tau_i = \min\{\tau,\ 4(\tau_c + \theta)\}$, where θ is the effective time delay of the process.

Since about 2000, partly inspired by the work of Åström (e.g., Åström and Hägglund (1988); Åström et al. (1992)), there has been a surge in academic papers on PID control as can be seen by the Handbook on PID rules by O'Dwyer (2006) which lists hundreds of

tuning rules.

In particular, the very simple SIMC PID tuning rules (Skogestad, 2003) have found widespread industrial acceptance. However, there has also been suggestions to improve the SIMC rules (Haugen, 2010; Lee et al., 2014). One question then naturally arises: Is there any point in searching for better PID rules for first-order plus delay processes, or are the SIMC rules good enough? To answer this question, we want in this chapter to answer the following three more detailed questions:

- What are the optimal PI and PID settings for a first-order with delay process?
- How close are the simple SIMC rules to these optimal settings?
- Can the SIMC rules be improved in a simple manner?

For our analysis we consider the stable first-order plus time delay processes

$$G(s) = \frac{ke^{-\theta s}}{(\tau s + 1)}, \tag{3.1}$$

where k is the process gain, τ is the process time constant, and θ is the process time delay. We mainly consider the serial (cascade) form PID controller,

$$K_{\text{PID}}(s) = \frac{k_c(\tau_i s + 1)(\tau_d s + 1)}{\tau_i s}, \tag{3.2}$$

where k_c, τ_i and τ_d are the controller gain, integral time and derivative time. The main reason for choosing this form is that the SIMC PID-rules become simpler. For the more common parallel (ideal) PID implementation

$$K_{\text{PID}}^{\text{PARALLEL}}(s) = \tilde{k}_c \left(1 + \frac{1}{\tilde{\tau}_i s} + \tilde{\tau}_d s\right), \tag{3.3}$$

one must compute the factor $f = 1 + \tau_d/\tau_i$, and use the following settings

$$\tilde{k}_c = k_c f, \quad \tilde{\tau}_i = \tau_i f, \quad \text{and} \quad \tilde{\tau}_d = \tau_d / f. \tag{3.4}$$

For PI-control, $f = 1$ and the two forms are identical. In addition, a filter F is added, at least when there is derivative action, so the overall controller is

$$K(s) = K_{\text{PID}}(s)\, F(s). \tag{3.5}$$

Normally, we use is a first-order filter,

$$F = \frac{1}{\tau_f s + 1}. \tag{3.6}$$

Note that τ_f is not considered a tuning parameter in this chapter, but rather set at a fixed small value, depending on the case. The filter is generally needed when we have derivative action, and we may write $\tau_f = \tau_d/\alpha$ where α often is in the range from 5 to 10. For other notation, see Figure 3.1.

The main trade-off in controller design is between the benefits of high controller gains (performance) and the disadvantages of high controller gain (robustness and input usage) e.g., (Boyd and Barratt, 1991; Kristiansson and Lennartson, 2006). In this chapter, we focus on the trade-off between IAE-performance and M_s-robustness. More precisely, we use the integrated absolute error (IAE) for combined input and output disturbances as the performance measure and obtain optimal PI and PID settings for various robustness levels, where robustness is measured in terms of the peak sensitivity (M_s) The resulting Pareto-optimal trade-off between performance and robustness is subsequently used to evaluate the SIMC PI and PID rules.

The chapter is structured as follows. In Section 2 we define the measures used to quantify the performance/robustness trade-off.

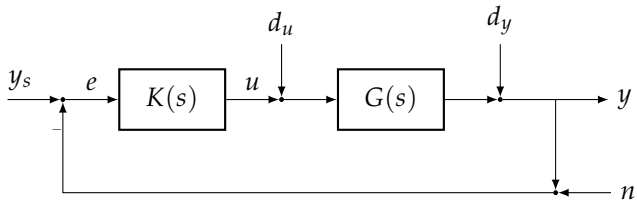

Figure 3.1: Block diagram of one degree-of-freedom feedback control system. We may treat a setpoint change (y_s) as a special case of an output disturbance (d_y).

Based on this, optimal PI and PID controllers are presented in Section 3. In Section 4 we present the SIMC rules and propose "improved" rules, referred to as *i*SIMC and *i*SIMC-PI. In Section 5, the various SIMC and the improved rules are *i*SIMC evaluated. In Section 6, we discuss input usage and some other issues.

Preliminary versions of some of the results were presented in Grimholt and Skogestad (2012a) and Grimholt and Skogestad (2013).

3.2 Quantifying the optimal controller

The first authors to use the terms "optimal settings" for PID-control where Ziegler and Nichols (1942) in their classical paper. Generally, it is difficult to define "optimality" of a controller, as there are many important aspects to take into consideration, including set-point response, disturbance rejection, robustness, input usage, and noise sensitivity. Often a control loop is evaluated solely on the basis of its response to a setpoint change, but in process control, disturbance rejection is usually the major concern. Another important aspect is robustness, which often is completely omitted. Åström and Häg-

glund (2006) emphasize the need of including all the behaviors of the control loop.

Performance

In this chapter, we quantify performance in terms of the IAE,

$$\text{IAE} = \int_0^\infty \left| y(t) - y_s(t) \right| \mathrm{d}t. \tag{3.7}$$

To balance the servo/regulatory trade-off we choose a weighted average of IAE for a step input disturbance d_u (load disturbance) and a step output disturbance d_y:

$$J(p) = 0.5 \left(\frac{\text{IAE}_{dy}(p)}{\text{IAE}_{dy}^{\circ}} + \frac{\text{IAE}_{du}(p)}{\text{IAE}_{du}^{\circ}} \right) \tag{3.8}$$

where IAE_{dy}° and IAE_{du}° are weighting factors, and p is the controller parameters. Note that we do not consider setpoint responses, but instead output disturbances. For the system in Figure 3.1, the closed-loop responses in the error $e = y_s - y$ to an output disturbance d_y and to a setpoint change y_s are identical, except for the sign. The difference is that since the setpoint is known we could further enhance the setpoint performance using a two-degrees-of freedom controller (which is not considered in this chapter), whereas the unmeasured output disturbance can only be handled by the feedback controller K (which is the focus of this chapter). Of course, we may consider other disturbance dynamics, but step disturbances at the plant input and output give are believed to be representative for most cases.

The two weighting factors IAE° for input and output disturbances, respectively, are selected as the optimal IAE values when using PI control (as recommended by Boyd and Barratt (1991)). To

ensure robust reference PI controllers, they are required to have M_S = 1.59, and the resulting weighting factors are given for four processes in Table 3.1. For those that are curious about the origin of this specific value M_S = 1.59, it is the resulting M_S value for a SIMC tuned PI controller with $\tau_c = \theta$ on FOPTD process with $\tau \leq 8\theta$. Note that two different reference PI controllers are used to obtain the weighting factors.

Robustness

Robustness may be defined in many ways, for example, using the classical gain and phase margins, which are related to robustness with respect to the model parameters k and θ, respectively. However, as a single robustness measure, we in this chapter quantify robustness in terms of M_{ST}, defined as the largest value of M_S and M_T (Garpinger and Hägglund, 2008),

$$M_{ST} = \max\{M_S,\ M_T\}. \tag{3.9}$$

Table 3.1: Reference PI-controllers and resulting weighting factors for four processes

	Output disturbance			Input disturbance		
Process	k_c	τ_i	IAE°_{dy}	k_c	τ_i	IAE°_{du}
e^{-s}	0.20	0.32	1.61	0.20	0.32	1.61
$e^{-s}/(s+1)$	0.55	1.14	2.07	0.52	1.05	2.02
$e^{-s}/(8s+1)$	4.00	8.00	2.17	3.33	3.67	1.13
e^{-s}/s	0.50	∞	2.17	0.40	5.78	15.10

IAE_{dy} and IAE_{du} are for a unit step disturbance on output (y) and input (u), respectively.

where M_S and M_T are the largest peaks of the sensitivity $S(s)$ and complimentary sensitivity $T(s)$ functions, respectively. Mathematically,

$$M_S = \max_{\omega} \left| S(j\omega) \right| = \left\| S(j\omega) \right\|_\infty ,$$
$$M_T = \max_{\omega} \left| T(j\omega) \right| = \left\| T(j\omega) \right\|_\infty ,$$

where $\|\cdot\|_\infty$ is the H_∞ norm (maximum peak as a function of frequency), and the sensitivity transfer functions are defined as

$$S(s) = {}^{1}/_{\left(1+G(s)\,K(s)\right)} \quad \text{and} \quad T(s) = 1 - S(s). \tag{3.10}$$

For most stable processes, $M_S \geq M_T$. In the frequency domain (Nyquist plot), M_S is the inverse of the closest distance between the critical point -1 and the loop transfer function $G(s)\,K(s)$. For robustness, small M_S and M_T values are desired, and generally M_S should not exceed 2. For a given M_S we are guaranteed the following GM and PM,(Rivera et al., 1986).

$$\text{GM} \geq \frac{M_S}{M_S - 1} \quad \text{and} \quad \text{PM} \geq 2 \arcsin\left(\frac{1}{2M_S}\right) \geq \frac{1}{M_S}. \tag{3.11}$$

For example, $M_S = 1.6$ guarantees GM ≥ 2.67 and PM $\geq 36.4^\circ = 0.64$ rad.

Optimal controller

For a given process and given robustness level (M^{ub}), the IAE-optimal controller is found by solving the the following optimization problem:

$$\min_{p} \quad J(p) = 0.5 \left(\frac{\text{IAE}_{dy}(p)}{\text{IAE}^{\circ}_{dy}} + \frac{\text{IAE}_{du}(p)}{\text{IAE}^{\circ}_{du}} \right) \tag{3.12}$$

$$\text{subject to:} \quad M_{\text{S}}(p) \leq M^{ub} \tag{3.13}$$

$$M_{\text{T}}(p) \leq M^{ub} \tag{3.14}$$

where in this chapter the parameter vector p is for a PI or PID controller. For more details on how to solve the optimization problem, see Grimholt and Skogestad (2018a). The problem is solved repeatedly for different values of M^{ub}. One of the constraints in (3.13) or (3.14) will be active if there is a trade-off between robustness and performance. This is the case for values of M^{ub} less than about 2 to 3. Usually the M_{S}-bound is active, except for integrating processes with a small M^{ub} (less than about 1.3), where the M_{T}-bound is active (see Figure 3.4, later).

In retrospect, looking at the results of this chapter, we would have obtained similar optimal PI- and PID-settings for the process (3.1) if we only considered input disturbances for performance and only used M_{S} for robustness.

3.3 Optimal PI and PID control

Trade-off between robustness and performance and comparison of PI and PID control

In this section, we present IAE-optimal (J) settings for PI and PID controllers as a function of the robustness level (M_{ST}). However, before presenting the optimal settings, we show in Figure 3.2 the Pareto optimal IAE-performance (J) as a function of robustness (M_{ST}) for optimal PI and PID controllers for three processes.

Note that the curves in Figure 3.2 stop when M_{ST} is between 2 and 3. This is because performance (J) actually gets worse and the curve for J bends upwards (Grimholt and Skogestad, 2012a) when M_{ST} increases beyond this value. Thus, there is no trade-off and the region with M_{ST} larger than about 2 should be avoided.

We see from Figure 3.2 that for a pure time delay process there is no advantage in adding derivative action; and it is optimal to use simple PI control. As the time constant τ increases, the benefit of using derivative action also increases. For an integrating process, derivative action improves IAE-performance by about 40%, compared to optimal PI control. This is emphasized again in Figure 3.3, where performance is shown as a function of the normalized time constant for robust controllers with $M_{ST} = 1.4$.

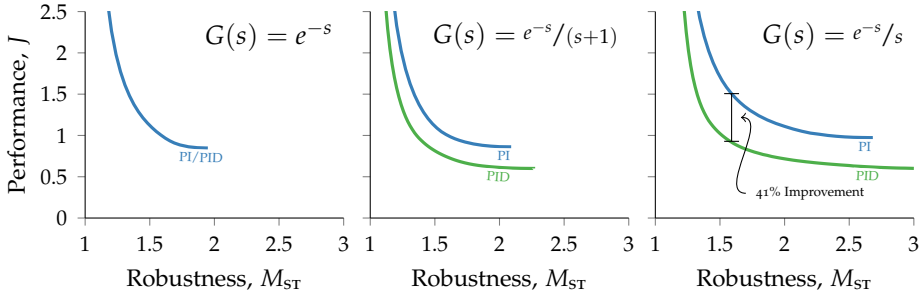

Figure 3.2: Pareto-optimal IAE-performance (J) for PI and PID control.

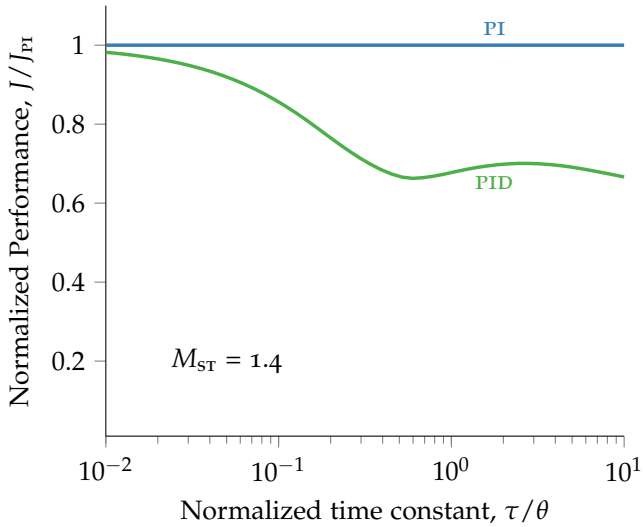

Figure 3.3: PID control: Normalized IAE-optimal performance for $M_{ST} = 1.4$ as a function of the normalized time constant τ/θ.

Optimal PI control

The IAE-optimal PI settings are shown graphically in Figure 3.4 as a function of τ/θ for different robustness levels (M_{ST}) and are also given for $M_S = 1.59$ for four processes in Table 3.2. For PI control we observe three main regions (Figure 3.4) in terms of optimal integral time τ_i:

Delay dominant:	$\tau/\theta < 0.4$	$\tau_i \approx \theta/3$
Balanced:	$0.4 < \tau/\theta < 4$	$\tau_i \approx \tau$
Lag dominant:[a]	$4 < \tau/\theta$	$\tau_i \approx k\theta$

[a] Where k depends on M_{ST} ($k \approx 6$ for $M_{ST} = 1.59$).

These regions match well the classification of first-order plus time delay processes in Garpinger et al. (2014).

In contrast with the IMC rules (Rivera et al., 1986) and the SIMC rules (Skogestad, 2003), the optimal controller does not converge to a pure integral controller ($K_I(s) = k_i/s$, corresponding to $\tau_i \to 0$) as $\tau/\theta \to 0$ (Figure 3.4). Rather, for a pure time delay processes, the integral time is approximately $\theta/3$, which we will use in the proposed *i*SIMC-PID and *i*SIMC-PI rules (see below). The optimal integral time of about $\theta/3$ is almost independent of the robustness level (M_{ST}-values) . For balanced processes, the integral time is similar to the time constant ($\tau_i \approx \tau$, see dashed line), and also almost independent of the robustness level. This value agrees with the IMC and SIMC rules.

For lag-dominant processes (with $\tau > 4\theta$), the integral time for $M_{ST} = 1.59$ approaches $\tau_i = 6.22\theta$ for $\tau/\theta = \infty$ (integrating process) (Figure 3.4, lower right). This is somewhat smaller than the value $\tau_i = 8\theta$ obtained from the SIMC rule. Also the normalized controller

gain, $k_c k\theta/\tau$ approaches a constant value as τ goes to infinity (Figure 3.4, upper right). For example, for $M_{ST} = 1.59$, the optimal value is $k_c k\theta/\tau = 0.414$ for $\tau/\theta = 50$, and 0.409 for $\tau/\theta = \infty$ (integrating process). This is close to the value $k_c k\theta/\tau = 0.5$ obtained with the SIMC-rule with $M_{ST} = 1.59$.

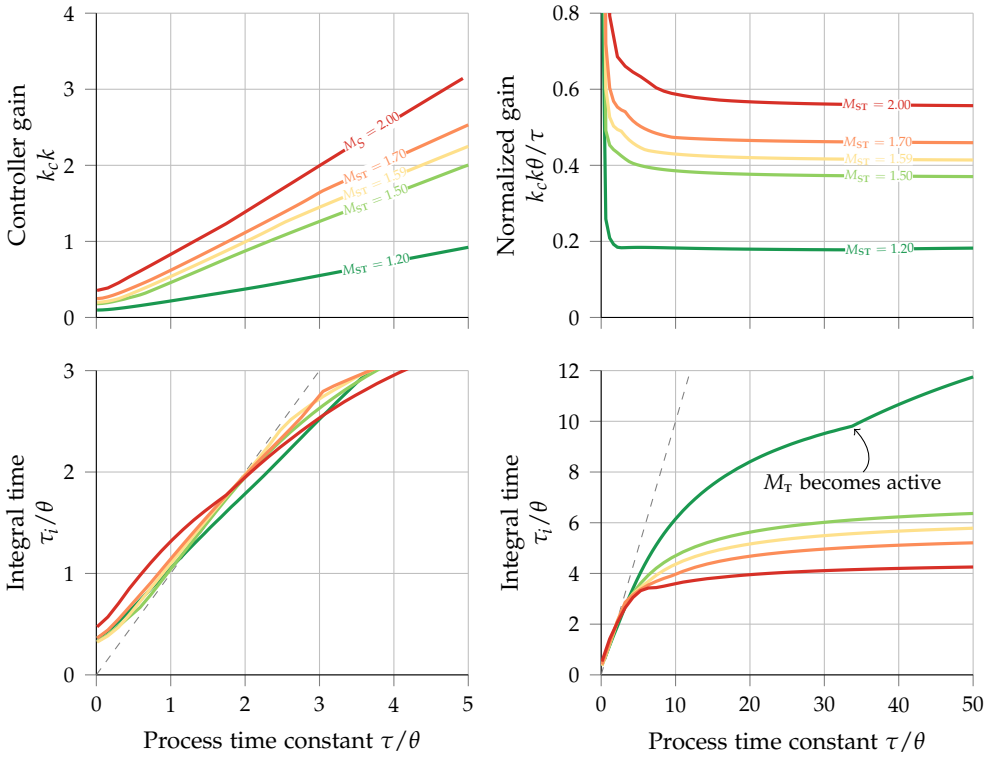

Figure 3.4: PI control: IAE-optimal settings as a function of τ/θ for five values of M_{ST}.

Table 3.2: PI control: Comparison of optimal, SIMC, and *i*SIMC-PI controllers with M_{ST} = 1.59.

	Optimal PI			SIMC				*i*SIMC-PI				
Process	k_c	τ_i	J	k_c	τ_i	τ_c	J	k_c	τ_i	τ_c	J	M_{ST}
e^{-s}	0.20	0.32	1.00	0[a]	0	1.00	1.35	0.21	0.33	0.61	1.00	1.59
$\frac{e^{-s}}{(s+1)}$	0.54	1.10	1.01	0.50	1.00	1.00	1.03	0.61	1.33	1.20	1.08	1.59
$\frac{e^{-s}}{(8s+1)}$	3.47	4.04	1.23	4.00	8.00	1.00	1.38	4.01	8.31	1.08	1.41	1.59
$\frac{e^{-s}}{s}$	0.41	6.22	1.50	0.45	8.97	1.24	1.63	0.45	8.97	1.24	1.63	1.59

[a] This is an I controller with integral gain $k_i = k_c/\tau_i$=0.5.

Optimal PID control

The IAE-optimal PID settings are shown graphically in Figure 3.5 and are also given for $M_s = 1.59$ for four processes in Table 3.3. The optimal PID settings can be divided into the same regions as for PI control. Note that for a pure time delay process, it is optimal with PI control, that is, it is optimal to have $\tau_d = 0$ and $\tau_i = \theta/3$. Actually, if we allow for having the derivative time larger than the integral time, then we could interpret it differently, and say that for a pure time delay process, the optimal controller is a integral-derivative (ID)-controller with $\tau_i = 0$ and $\tau_d = \theta/3$. We will see that this latter interpretation is consistent with the proposed improved SIMC PID-rule, whereas the first interpretation is consistent with the improved SIMC PI-rule.

For PID-control, the balanced region ($0.2 < \tau/\theta 4$) can be divided in two. In the lower part (τ/θ<1.25), the optimal derivative and integral time are the same, $\tau_i = \tau_d$, and increase with τ/θ. In the upper part, τ_i increases with τ/θ, whereas τ_d remains approximately constant. Note that the region with $\tau_i = \tau_d$ agrees with the recommendation of Ziegler and Nichols (1942)*. However, we see from Figure 3.5 that $\tau_i = \tau_d$ is optimal only for a fairly small range of first-order plus time delay processes with τ/θ between about 0.2 and 1.25.

From Figure 3.5 we see that the integral time (τ_i) is smaller than the process time constant (τ) for all processes with $\tau/\theta > 4$, whereas we found that $\tau_i \approx \tau$ was optimal in the balanced region for PI control. For given values of M_{sT}, the optimal PID controller gain is slightly larger than the optimal PI controller gain, and the integral

* Ziegler and Nichols (1942) recommend the integral time to be 4 times the derivative time for the parallel (ideal) PID controller, which for the serial (cascade) PID form corresponds to τ_i=τ_d, see (3.4).

action is also larger (with a lower value of τ_i).

For lag-dominant processes ($\tau/\theta > 4$), the normalized controller gain $k_c k\theta/\tau$ approaches a constant value as $\tau \to \infty$. For example, for $M_{ST} = 1.59$ we have $k_c k\theta/\tau \to 0.54$. The same can be observed for the integral and derivative times which for $M_{ST} = 1.59$ approach $\tau_i/\theta = 3.24$ and $\tau_d/\theta = 0.48$, respectively (Figure 3.5, bottom right). For increasing M_{ST} values (less robustness), the optimal controller

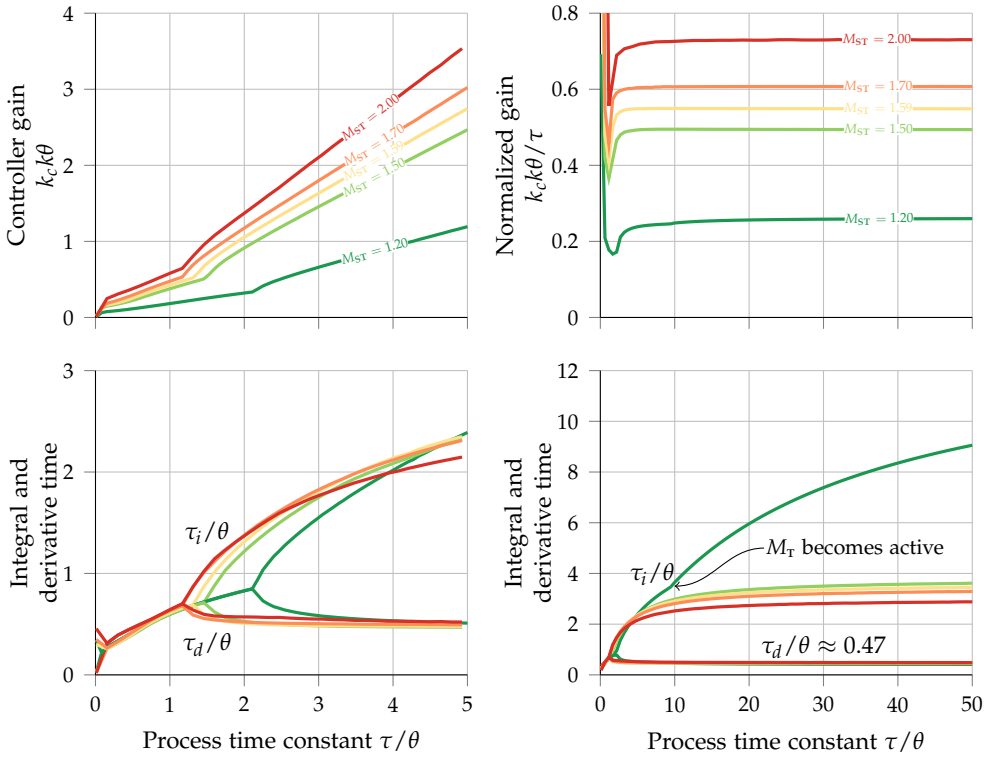

Figure 3.5: PID control: optimal settings as a function of τ/θ for five values of M_{ST}.

gain increases and the optimal integral time decreases. Interestingly, for all lag-dominant processes the optimal derivative time is $\tau_d \approx 0.47\theta$ almost independent of the M_{ST}-value.

Table 3.3: PID control: Comparison of optimal and SIMC controllers with $M_{ST} = 1.59$.

	Optimal PID				*i*SIMC						
Process	k_c	τ_i	τ_d	J	k_c	τ_i	τ_d	τ_c	J	M_{ST}	Performance (J) loss
e^{-s}	0.20	0.32	0	1.00	0 [a]	0	0.33	0.61	1.00	1.59	0%
$\frac{e^{-s}}{(s+1)}$	0.42	0.61	0.61	0.74	0.62	1.00	0.33	0.61	0.79	1.59	6%
$\frac{e^{-s}}{(8s+1)}$	4.34	2.63	0.49	0.81	4.92	6.50	0.33	0.63	1.00	1.59	23%
$\frac{e^{-s}}{s}$	0.53	3.18	0.51	0.89	0.59	6.81	0.33	0.70	1.09	1.59	22%

[a] This is an ID controller with integral gain $k_i = k_c/\tau_i = 0.62$. The ID controller can be rewritten as a PI controller ($\tau_d = 0$) with $k_c = 0.62 \times 0.33 = 0.21$ and $\tau_i = 0.33$.

Parallel vs. serial PID controller

The above optimization was for the serial PID controller in (3.2). A more general PID controller is the parallel, or ideal, PID controller in (3.3), which allows for complex zeros. The parallel PID controller is better only for processes with τ/θ between 0.4 and 1.2, which is the region where $\tau_i = \tau_d$ (two identical real zeros) for the serial PID controller. Furthermore, the improvement with the parallel (ideal) PID form is very minor as illustrated in Figure 3.6, which compares the IAE performance for a "balanced" process with $\tau/\theta = 1$. Therefore, the serial PID implementation in (3.2) is sufficient for first-order plus time delay processes.

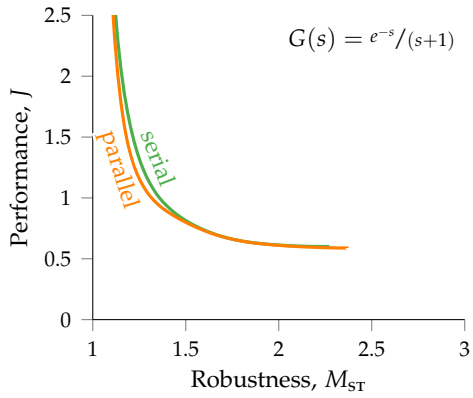

Figure 3.6: PID control: Comparison of IAE-optimal performance (J) for serial PID control (3.2) and parallel PID control (3.3) for a process with $\tau/\theta = 1$.

3.4 The original and improved SIMC rules

Original SIMC rule

We consider the first-order with delay process in (3.1). The original SIMC PID tunings for this process give a PI controller (Skogestad, 2003):

SIMC

$$k_c = \frac{1}{k}\frac{\tau}{(\tau_c + \theta)}, \quad \tau_i = \min\left\{\tau,\ 4(\tau_c + \theta)\right\}. \tag{3.15}$$

Here the closed-loop time constant τ_c is an adjustable tuning parameter which is used to get the desired trade-off between output performance, robustness and input usage. For "tight control" (good performance) with acceptable robustness (M_s about 1.6 to 1.7), Skogestad (2003) recommends selecting $\tau_c = \theta$. However, in many cases "smooth control" is desired and we should use a larger value for τ_c.

"Improved" SIMC rule with derivative action (*i*SIMC)

In this chapter, we propose the "improved" SIMC PID-rule for a first-order with delay process. Since an important feature of the SIMC rules is simplicity, we keep the same expressions for k_c and τ_i as in the original PI-rule in (3.15), but derivative action is added to improve the performance for a time delay,

***i*SIMC**

$$\tau_d = \theta/3 \tag{3.16}$$

Note that the *i*SIMC tunings are for the serial PID implementation in (3.2). For the more common parallel (ideal) PID implementation in

(3.3), one must compute the factor $f = 1 + \tau_d/\tau_i$ and use the values in (3.4).

As seen from Figure 3.5, the value $\tau_d = \theta/3$ is close to the optimal for a pure time delay process (with $\tau = 0$). For larger values of process time constant τ, the optimal value of τ_d is closer to $\theta/2$. However, we chose to use the smaller value $\tau_d = \theta/3$ in order to reduce possible other disadvantages of adding derivative action.

If we use the same value for the tuning constant (e.g. $\tau_c = \theta$) as for the original SIMC PI controller in (3.15), then the addition of the derivative action in (3.16) mainly improves robustness (lower M_s). However, the main reason for introducing derivative action is usually to improve performance, and to achieve this one should reduce τ_c. In the original SIMC rule (Skogestad, 2003) it was recommended to select $\tau_c = \theta$ to achieve "tight control" with acceptable robustness (M_s about 1.6 to 1.7). However, as will become clearer from the results below, for the *i*SIMC PID rule we recommend reducing the value of τ_c and select $\tau_c \geq \theta/2$.

The SIMC PI-tunings parameters with $\tau_c = \theta$ and the *i*SIMC PID-tunings with $\tau_c = \theta/2$ are given for four first-order plus delay processes in Table 3.4. As seen from, *i*SIMC PID-improves IAE performance (J) by about 30% compared to the original SIMC PI controller, while keeping about the same robustness level (M_{st} about 1.7).

Note that we have put "improved" in quotes for the *i*SIMC rule. Indeed, in the original SIMC chapter, Skogestad (2003) considered adding the derivative time $\tau_d = \theta/2$ to counteract time delay, but concluded that it was probably not worth the increased complexity of the controller and the increased sensitivity to measurement noise and input usage. Therefore, in most practical situations in industry, the original SIMC PI-rule is most likely preferable. Nevertheless, if performance is important and τ_c is adjusted as mentioned above,

Table 3.4: SIMC PI controller and *i*SIMC PID controller with tuning constant $\tau_c = \theta$ and $\tau_c = \theta/2$, respectively.

	SIMC (PI, $\tau_c = \theta$)						*i*SIMC (PID, $\tau_c = \theta/2$)						
Process	k_c	τ_i	IAE$_{dy}$	IAE$_{du}$	J	M_{ST}	k_c	τ_i	τ_d	IAE$_{dy}$	IAE$_{du}$	J	M_{ST}
e^{-s}	0.50[a]	0	2.17	2.17	1.35	1.59	0.67*	0	0.33	1.50	1.50	0.93	1.66
$\frac{e^{-s}}{(s+1)}$	0.50	1.00	2.17	2.04	1.03	1.59	0.67	1.00	0.33	1.50	1.50	0.73	1.66
$\frac{e^{-s}}{(8s+1)}$	4.00	8.00	2.17	2.00	1.38	1.59	5.33	6.00	0.33	1.80	1.12	0.91	1.67
$\frac{e^{-s}}{s}$	0.50	8.00	3.92	16.00	1.43	1.70	0.67	6.00	0.33	2.83	9.00	0.95	1.73

[a] Integral gain $k_i = k_c/\tau_i$.

then the results of this chapter show (e.g. see Figure 3.9), that significant improvements in performance may be achieved with the *i*SIMC rule. Additionally, we have found that PID control with *i*SIMC tunings is better in almost all respects than a well-tuned Smith Predictor controller (Grimholt and Skogestad, 2018c).

Alternative improved SIMC rule without derivative action (*i*SIMC-PI) for delay dominant processes

Note that for a pure time delay process ($\tau = 0$), the *i*SIMC PID-controller in (3.15-3.16) is actually a ID-controller, since $k_c = 0$. As noted earlier this ID-controller (with $\tau_d = \theta/3$ and $\tau_i = 0$) may be realized instead as a PI-controller (with $\tau_i = \theta/3$ and $\tau_d = 0$). This is the basis for the following "improved" SIMC PI rule for a FOPTD process, denoted *i*SIMC-PI (Grimholt and Skogestad, 2012b):

*i*SIMC-PI

$$k_c = \frac{1}{k}\frac{\tau + \theta/3}{(\tau_c + \theta)}, \quad \tau_i = \min\left\{\tau + \theta/3,\ 4(\tau_c + \theta)\right\}. \tag{3.17}$$

Note that for a pure time delay process ($\tau = 0$), the *i*SIMC PID-controller in (3.15-3.16) and the *i*SIMC-PI PI-controller in (3.17) are identical. The *i*SIMC-PI tunings in (3.17) may give significant performance improvements benefits compared to the original SIMC PI-tunings for delay-dominant processes, but at the expense of larger input usage. However, for processes with $\tau > \theta/2$, approximately, we find that the benefits are marginal or even negative.

3.5 Evaluation of the SIMC and *i*SIMC rules

SIMC PI-rule (original)

We compare in Figure 3.7 the Pareto-optimal IAE performance (J) of the SIMC PI controller with the IAE-optimal PI controller for four different processes. The PI settings for M_{ST} = 1.59 are given in Table 3.2. In addition, the SIMC controllers for three specific choices of the tuning parameter,

- $\tau_c = 1.5\theta$ (smoother tuning)
- $\tau_c = \theta$ (tight/recommended tuning)
- $\tau_c = 0.5\theta$ (more aggressive tuning)

are shown by circles. For the SIMC controller (Figure 3.7), the trade-off curves were generated by varying the tuning parameter τ_c from a large to a small value.

Except for the pure time delay process, the IAE-performance SIMC PI-controller is very close (within 10%) to the IAE-optimal PI controller for all robustness levels. In other words, by adjusting τ_c we can generate a close-to-optimal PI-optimal controller with a given desired robustness. Another important observation is that the default PID-recommendation for "tight" control, τ_c=θ (as given by middle of the three circles), in all cases is located in a desired part of the trade-off region, well before we reach the minimum. Also, the recommended choice gives a fairly constant M_S-value, in the range from 1.59 to 1.7. From this we conclude that, except for the pure time delay process, there is little room to improve on the SIMC PI rule, at least when performance and robustness are as defined above (J and M_S).

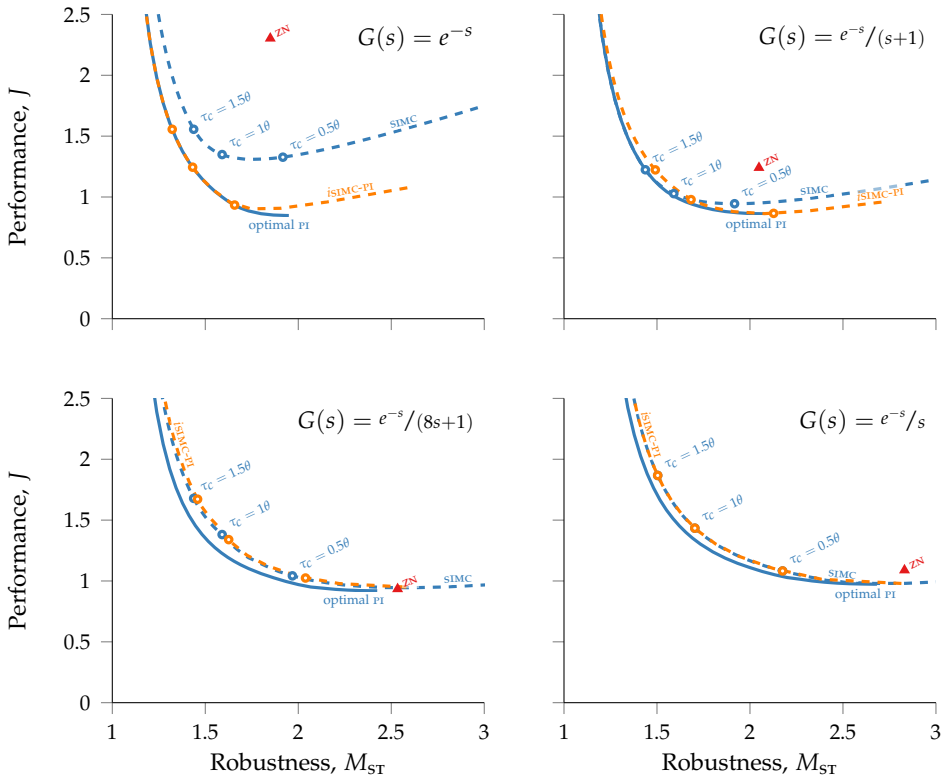

Figure 3.7: PI control: Pareto optimal IAE-performance (J) for optimal-PI, SIMC (3.15), and *i*SIMC-PI (3.17) control for four processes. The trade-off for SIMC and *i*SIMC-PI is generated by changing the value of the closed-loop tuning constant τ_c.

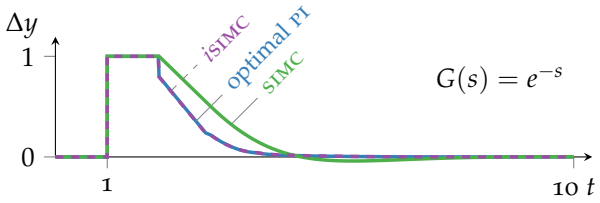

Figure 3.8: Time response for optimal, SIMC and *i*SIMC controllers (M_{ST}=1.59) for an input/output disturbance (at time 0/1) for the pure time delay process $G(s) = e^{-s}$. For pure time delay process the input u response is equal to the output y response time shifted one time delay earlier. Also the response is the same for an input or output disturbance (but shifted). Note that for a pure time delay process there is no benefit of adding derivative action ($\tau_d = 0$ is optimal).

Improved SIMC PI rule (*i*SIMC-PI)

The main "problem" with the original SIMC rule is for pure time delay processes, where the IAE-performance (J) is about 40% higher than the optimal (Figure 3.7). The proposed *i*SIMC-PI rule in (3.17) rectifies this. As seen from Figure 3.7 (upper left), the proposed *i*SIMC-PI rule is almost identical to the IAE-optimal controller when τ_c is adjusted to give the same robustness (M_{ST}). This is further illustrated by the simulation in Figure 3.8.

Improved SIMC PID rule (*i*SIMC)

Next, we consider PID control, that is, the addition of derivative action using $\tau_d = \theta/3$, as proposed with the *i*SIMC rule (3.16). We compare in Figure 3.9 the IAE performance (J) of the *i*SIMC PID controller with the IAE optimal PID controller with the same robustness (M_{ST}) for four different processes (green curves). PID settings for

M_{ST} = 1.59 are given in Table 3.3. To illustrate the benefits of derivative action we also show in Figure 3.9 the curves with PI control (blue curves).

We see from Figure 3.9 that the SIMC PID-controller (dashed green curve) is close to the optimal PID-controller (solid green line) for all four processes and all robustness levels. By considering the location of the middle green circles, we see that if we keep the value of τ_c unchanged at $\tau_c = \theta$, then adding derivative action mainly improves robustness.

For example, for an integrating process and $\tau_c = \theta$, the value of M_{ST} is improved from 1.70 for PI to 1.46 for PID, but there is only a 6% improvement in performance. However, by reducing τ_c we can significantly improve performance for a given M_{ST} value.

For the four processes, we see from Figure 3.9 that $\tau_c = \theta/2$ (rightmost green circles) is a good choice for the tuning constant for the *i*SIMC PID controller. Compared to SIMC PI controller tuned with $\tau_c = \theta$, this gives about 30% better IAE-performance and similar robustness (M_{ST} about 1.7).

We said that the SIMC PID-controller is close to the optimal PID-controller. However, we see from the two lower plots in Figure 3.9 that the performance loss is somewhat larger for processes with large time constants. To study this further, we compare in Figure 3.10 the step responses for various PI and PID controllers for an integrating process. Because of a larger integral time, the SIMC and *i*SIMC controllers settle more slowly than the optimal PI and PID controllers for both input and output disturbances. This results in a 22% higher average IAE-performance (J) for the *i*SIMC PID controller when compared with the optimal PID controller. However, it is usually the maximum deviation that is of main concern in industry.

Because of a larger controller gain, the SIMC controllers have

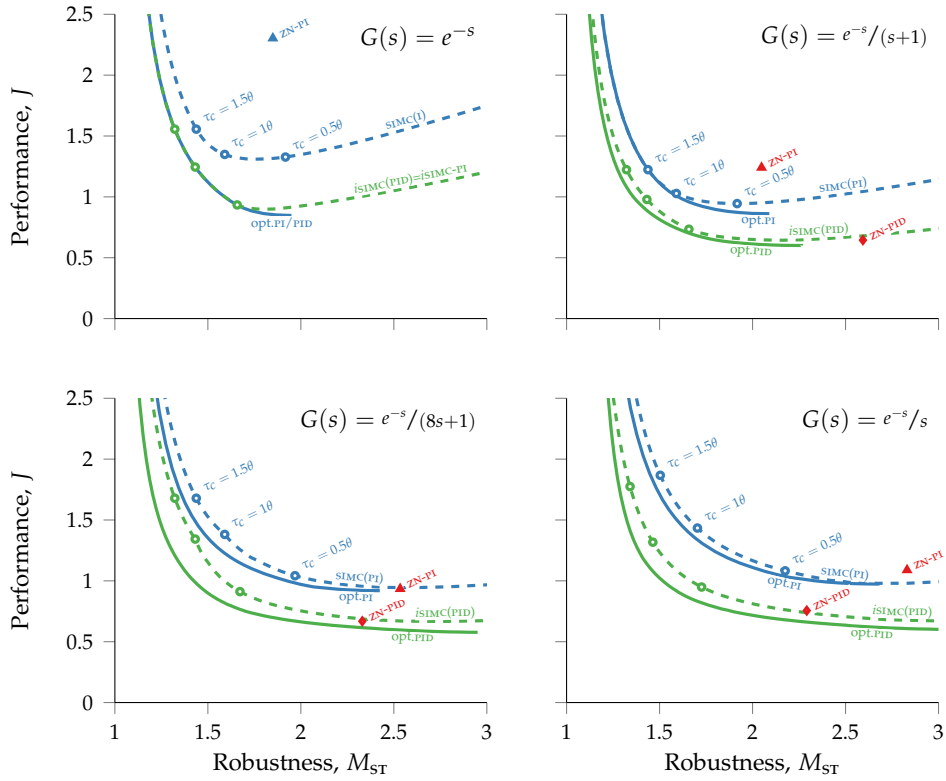

Figure 3.9: Pareto optimal IAE-performance (J) as a function of robustness M_{ST} for optimal PI and PID, and SIMC (3.15), *i*SIMC (3.16) and ZN for four processes. The trade-off for the various SIMC rules are generated by changing the value of the closed-loop tuning constant τ_c. PI controllers are shown with blue and PID controllers with green.

roughly the same peak deviation as the optimal PI and PID controllers for input disturbance (see Figure 3.10), and a smaller overshoot for output disturbances (setpoint) than the optimal. Thus, we conclude that the performance of the SIMC controllers are better than indicated from the IAE-value (J), when we take into account other aspects of performance.

In conclusion, also for PID control, we conclude that *i*SIMC is close to the optimal PID-controller, so the benefit of looking for even more "improved" rules for first-order plus time delay processes is limited.

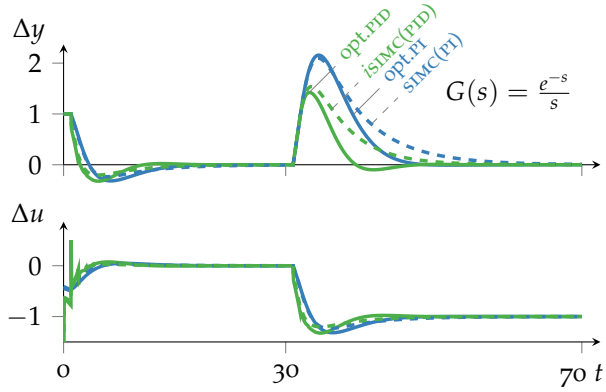

Figure 3.10: Time response for optimal PI and PID compared with SIMC and *i*SIMC controllers (M_{ST} = 1.59) for an output disturbance (at time 0) and an input disturbances (at time 20), for an integrating process $G(s) = e^{-s}/s$. To get a proper system, a first order measurement filter with $\tau_f = 0.02$ was applied to the PID controllers.

3.6 Discussion

Input usage and filtering

Input usage is an important aspect for control, but have not been explicitly treated in this work. From Figure 3.1 we have

$$u = -T\, d_u - KS\, \Big(d_y + n\Big)\,.$$

Thus, input usage is determined by two transfer functions: $T = GK/(1+GK)$ (for input disturbance) and $KS = K/(1+GK)$ (for output disturbance and noise). Input disturbances will not pose a new problem because T is closely related to performance and is in addition already bounded by M_{T}.

The important new transfer function is therefore KS, and by limiting its peak one can adjust input usage related to measurement noise and output disturbances (Kristiansson and Lennartson, 2002). For PI control, KS has a peak at intermediate frequencies which is approximately (Åström and Hägglund, 2006):

$$\|KS\|_{\infty}^{\mathrm{PI}} \approx k_c M_{\mathrm{S}}. \tag{3.18}$$

Here, M_{S} is already bounded in our analysis, and is typically smaller than 1.7. Thus, we find that the controller gain k_c provides direct information about the input usage related to measurement noise and output disturbances.

For PID control, the value of $|KS|$ is generally higher at higher frequencies, and we find the input usage is dominated by the selected measurement filter. If there is no measurement filtering, $\tau_f = 0$, then the high-frequency peak goes to infinity. Therefore, for PID control it is important to filter out the high frequency noise, and the resulting peak will depend heavily on the selected filter time

constant. With a first or second order measurement filter

$$F_1(s) = \frac{1}{\tau_f s + 1} \quad \text{or} \quad F_2(s) = \frac{1}{(\tau_f s)^2 + \sqrt{2}\tau_f s + 1}, \tag{3.19}$$

the high-frequency peak can be approximated by

$$\|KS\|_\infty^{\text{PID}} \approx \alpha k_c \tag{3.20}$$

where $\alpha = \tau_d/\tau_f$. Note that τ_f here is for the cascade PID-controller in (3.2) and not for the ideal form in (3.3). Typically, α is between 5 and 10. The ratio in input magnitude between PI and PID related to measurement noise and output disturbances can then be expressed as

$$\frac{\|KS\|_\infty^{\text{PID}}}{\|KS\|_\infty^{\text{PI}}} \approx \frac{{k_c}^{\text{PID}}}{{k_c}^{\text{PI}}} \frac{\alpha}{{M_s}^{\text{PI}}}. \tag{3.21}$$

With the recommended tight tuning ($\tau_c = \theta$ for SIMC PI and $\tau_c = \theta/2$ for *i*SIMC PID), we get $\frac{{k_c}^{\text{PID}}}{{k_c}^{\text{PI}}} = (\theta + \theta)/(\theta/2 + \theta) = 1.33$ and the ratio in input usage can be expressed as

$$\frac{\|KS\|_\infty^{i\text{SIMC(PID)}}}{\|KS\|_\infty^{\text{SIMC(PI)}}} \approx \frac{1.33}{1.6}\alpha \approx 0.8\alpha \tag{3.22}$$

where we have assumed ${M_s}^{\text{PI}}$ to be 1.6. Since α is typically larger than 5, this means that the improved IAE performance of PID control may require an input magnitude related to measurement noise and output disturbances which is at least 4 times larger than for PI control [*].

Trade-off curves for *i*SIMC with different first-order measurement filters are shown in Figure 3.11. For the recommended PID

[*] We have assumed that the selected filter does not influence controller performance and robustness in a significant way. Otherwise, we have a PIDF controller where also the filter constant τ_f should be considered a degree of freedom in the optimization problem.

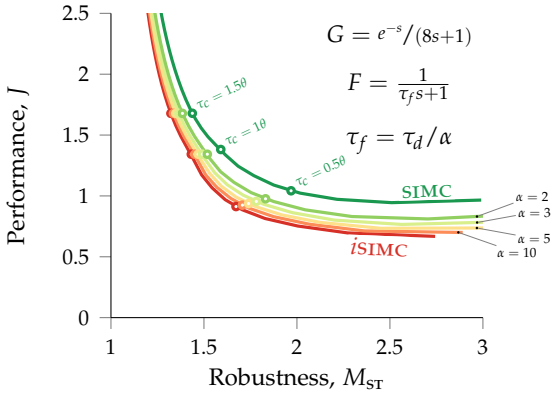

Figure 3.11: Performance/robustness trade-off curve for the *i*SIMC (PID) controller with measurement filtering. The SIMC (PI) controller corresponds to the case with $\alpha = 1$.

tuning, $\tau_c = \theta/2$, performance and robustness will deteriorate with increased measurement filtering. With $\tau_c = \theta/2$ and $\alpha = 3$, the robustness is quite low, and a retuning of the controller by selecting a larger τ_c might be necessary. With α=1, we recover the original SIMC PI-controller for which we recommend $\tau_c = \theta$ to get a good trade-off between performance and robustness.

Based on Figure 3.11, we recommend for PID-control to choose α in the range from 5 to 10, which gives most of the benefit of the D-action. The high-frequency input usage may then increase by a factor 4 to 8 compared to PI-control. This increase in input usage may be undesirable, so for many real process applications where performance is not the key issue, the original SIMC rule, which gives a PI-controller, is the best choice.

If noise filtering is an important factor, an iterative design approach can be used in combination with SIMC to ensure that the con-

troller has both good robustness and low noise sensitivity (Segovia et al., 2014).

Trade-off between input and output disturbance response

As already noted from Table 3.1 and observed from the simulations in Figure 3.12, the optimal controller that minimizes the average IAE performance (J) in (5.12), puts more emphasis on disturbance rejection at the input (IAE_{du}) than at the output (IAE_{dy}). Especially for larger values of the process time constant. For example, for an integrating process we find $\widetilde{\text{IAE}}_{du}$ = 1.02 (close to optimal for input disturbance), whereas $\widetilde{\text{IAE}}_{dy}$=1.99 (twice the optimal).

This is further illustrated in Figure 3.13, where we show ratio between the two terms in the IAE-performance index J (Huba, 2013), which in this chapter we term *controller balance*,

$$\text{controller balance} = \frac{\text{IAE}_{du}}{\text{IAE}^{\circ}_{du}} / \frac{\text{IAE}_{dy}}{\text{IAE}^{\circ}_{dy}}, \tag{3.23}$$

as a function of τ/θ. From Figure 3.13, we also note that for time constants less than about 3θ, the optimal controller has roughly equal weight on input and output (also seen in Figure 3.12, top).

For larger time constants, the emphasis shifts towards input disturbances. Interestingly, if we use a cost function with only a small weight (1%) on input disturbances

$$J(p) = \left(0.99 \, \frac{\text{IAE}_{dy}(p)}{\text{IAE}^{\circ}_{dy}} + 0.01 \, \frac{\text{IAE}_{du}(p)}{\text{IAE}^{\circ}_{du}} \right), \tag{3.24}$$

we find for an integrating process the optimal settings k_c = 0.462 and τ_i = 12.2θ, with $\widetilde{\text{IAE}}_{du}$ = 1.72 and $\widetilde{\text{IAE}}_{dy}$ = 1.91. We notice that there is only a minor improvement in setpoint performance (IAE_{dy}

decreases about 4%), whereas disturbance rejection is much worse (IAE$_{du}$ increases about 69%).

The conclusion is this that we may put emphasis mainly on input disturbances when tuning PI controllers for lag-dominated processes.

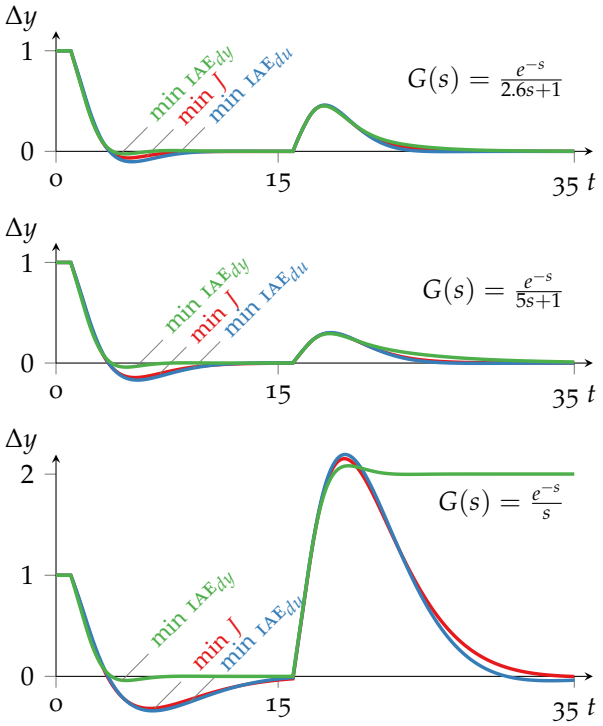

Figure 3.12: Comparison of step disturbance responses of IAE optimal controller which minimizes J, with the two reference controllers that consider disturbances only at the output (IAE$^{\circ}_{dy}$) and input (IAE$^{\circ}_{du}$), respectively. All controllers have $M_{ST} = 1.59$ (Table 3.1).

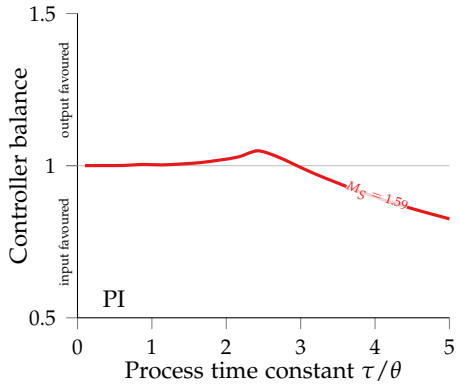

Figure 3.13: Controller balance between input and output disturbances, defined as $(\text{IAE}_{du}/\text{IAE}^{\circ}_{du})/(\text{IAE}_{dy}/\text{IAE}^{\circ}_{dy})$, for optimal PI control as a function of the time constant τ/θ ($M_{\text{ST}} = 1.59$).

Further evaluation of SIMC PI rule for integrating processes

When comparing the optimal PI settings with the original SIMC rule for an integrating process, we find that the SIMC integral time is larger than the optimal (Figure 3.4, bottom). Specifically, for an integrating process with $\tau_c = \theta$ (giving $M_{\text{ST}} = 1.70$), the SIMC rule gives $\tau_i/\theta = 8$, whereas the optimal performance (J) for the same robustness is with τ_i/θ=5.6. This indicates that the SIMC rule puts more emphasis on output disturbances than input disturbances, than for the IAE-optimal controller with equal weighting.

To shift the trade-off between output (setpoint) and input disturbance, one may introduce an extra parameter in the tuning rule (Alcántara et al., 2010; Di Ruscio, 2010). Haugen (2010) suggested to introduce an extra servo/regulator trade-off parameter c in the

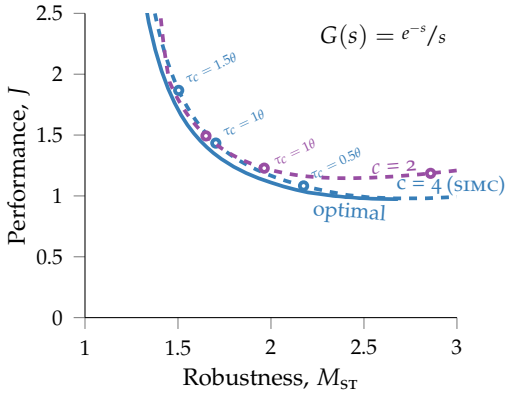

Figure 3.14: PI control: Evaluation of suggested modification $c = 2$ in (3.24) for integrating process.

expression for the integral time,

$$\tau_i = \min\{\tau,\ c(\tau_c + \theta)\}, \tag{3.25}$$

where $c = 4$ gives the original SIMC rule. However, introducing an extra parameter adds complexity, and the potential performance benefit of approximately 10% (see Figure 3.7) does not seem sufficiently large to justify it. Nevertheless, one may consider choosing another (lower) fixed value for c, and Haugen (2010) suggests using $c = 2$ to improve performance for input disturbances.

If we use the recommended tuning $\tau_c = \theta$, we find indeed that IAE performance J is improved compared to SIMC (see Figure 3.14). However, robustness is worse, with M_{ST} close to 2 (where the SIMC rule gives M_{ST} close to 1.7). More importantly, as seen from Figure 3.14 the SIMC performance is better if we decrease τ_c to get the same robustness in terms of M_{ST}.

In fact, SIMC is closer to the Pareto optimal curve for most values of M_{ST}. Actually, a better fixed value would be $c = 3$. However, changing the parameter c causes the recommended tuning $\tau_c = \theta$ to shift to the less robust region.

In summary, we find that the value $c = 4$ in the original SIMC rule provides a well balanced servo/regulator trade-off. To improve performance for input disturbances on an integrating process, we recommend decreasing the tuning constant τ_c/θ, say to around 0.7, rather than changing the value of c.

*i*SIMC for second-order plus delay process

For a second-order plus delay process,

$$G = \frac{ke^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \tag{3.26}$$

where $\tau_1 > \tau_2$, the original SIMC rule gives a PID controller on the series form (3.2) with

$$k_c = \frac{1}{k}\frac{\tau}{(\tau_c + \theta)}, \quad \tau_i = \min\left\{\tau,\ 4(\tau_c + \theta)\right\}, \quad \tau_d = \tau_2. \tag{3.27}$$

The direct extension of the *i*SIMC rule would be to add another derivative term, $(\frac{\theta}{3}s + 1)$, to the numerator of the PID controller in (3.2). First, this would not be a standard industrial controller and, second, it would give even more aggressive input usage. Thus, to get a PID controller, the following modified derivative time is recommended [*]

$$i\text{SIMC}: \quad \tau_d = \tau_2 + \theta/3 \tag{3.28}$$

[*] If τ_2 is very large, specifically if $\tau_2 > 4(\tau_c + \theta)$, then one should approximate the process as a double integrating process, $G(s) \approx k''e^{-\theta s}/s^2$ with $k'' = k/(\tau_1\tau_2)$, and use the PID-tunings for a double integrating process (Skogestad, 2003).

with the controller gain and integral time as given in (3.27). To get a good trade-off between performance and robustness, we may select $\tau_c = \theta$, but τ_c may be reduced towards $\theta/2$ for processes where τ_2 is smaller than $\theta/3$. Again, to get settings for the parallel (ideal) PID-controller in (3.3) one must compute the factor $f = 1 + \tau_d/\tau_i$, and use (3.4).

Ziegler-Nichols tuning rule

We also show in Figure 3.9 by red triangles the location of the classical Ziegler and Nichols (1942) (ZN) PI and PID controllers. The ZN-tunings are obtained by first bringing the process to sustained oscillations using a P-controller and recording the resulting "ultimate" period and controller gain. Since the ZN rules have no tuning parameter we get a single point in Figure 3.9. With exception of the pure time delay process (where ZN-PID is unstable and ZN-PI has very poor performance), the IAE performance for ZN is very good, but the ZN controllers are located in the "flat" trade-off region with poor robustness (large M_{ST} value).

The Ziegler-Nichols PID tuning rules were the by far most used rules for about 50 years, up to about 1990. The very poor performance of the ZN rules for pure time delay processes may then partly explain the myth that "time delay compensator", such as the Smith Predictor, may give significant performance benefits compared to PI- or PID-control for processes with large time delays (Grimholt and Skogestad, 2018c).

3.7 Conclusion

The IAE-optimal PI- and PID-settings for a first-order plus delay process (3.1) are shown for various robustness levels (as expressed by the M_s-value) in Figures 3.4 and 3.5, respectively. However, in practice, we recommend using the SIMC-rules for PI- and PID-tuning.

For PI-control, Figure 3.7 shows that the "original" SIMC rule in (3.15) (Skogestad, 2003) gives close-to optimal PI-performance. That is, by adjusting the tuning constant τ_c to get a desired robustness, we can closely track the Pareto-optimal trade-off curve between performance and robustness. The only exception is for delay-dominant FOPTD processes, where the SIMC proportional gain is too small, but this can be corrected for by using the *i*SIMC-PI rule in (3.17).

For PID-control, we propose the *i*SIMC rule where derivative action with $\tau_d = \theta/3$ (3.16) is added. Note that this is for the cascade PID-controller in (3.2). Figure 3.9 shows that this rule gives close-to optimal PID-performance, even for delay-dominant processes. For a pure time delay process, the *i*SIMC PID-controller is an ID-controller which can by rewritten to give the *i*SIMC-PI rule in (3.17).

The improved performance/robustness trade-off of the *i*SIMC-PI and *i*SIMC rules, comes at the expense of increased input usage in response to measurement noise, output disturbances and setpoint changes. Thus, for most industrial cases where output performance is not the main concern, the original SIMC rule may be the best choice.

Chapter Four

Optimal PID control of double integrating processes

In this chapter we investigate optimal PID control of a double integrating plus delay process and compare with the SIMC rules. What makes the double integrating process special is that derivative action is actually necessary for stabilization. In control, there is generally a trade-off between performance and robustness, so there does not exist a single optimal controller. However, for a given robustness level (here defined in terms of the M_s-value) we can find the optimal controller which minimizes the performance J (here defined as the IAE-value for disturbances). Interestingly, the SIMC PID controller is almost identical to the optimal PID controller. This can be seen by comparing the Pareto-optimal curve for J as a function of M_s, with the curve found by varying the SIMC tuning parameter τ_c.

4.1 Introduction

In this chapter we investigate optimal PID control of a double integrating plus delay process,

$$G(s) = \frac{k'' e^{-\theta s}}{s^2} \tag{4.1}$$

where k'' is the process gain and θ is the time delay. We will mostly consider the serial (or cascade) PID form,

$$K_{\text{PID}}(s) = \frac{k_c(\tau_i s + 1)(\tau_d s + 1)}{\tau_i s}, \tag{4.2}$$

where k_c, τ_i and τ_d are the controller gain, integral time and derivative time. However, we will also compare with the more general parallel (ideal) form, which can have complex zeros. For other notation, see Figure 4.1.

What makes the double integrating process special, is that derivative action is actually necessary for stabilization. Because the feedback system is unstable with proportional only controllers, traditional tuning methods like Ziegler and Nichols (1942) cannot be applied.

The SIMC method for PID controller tuning (Skogestad, 2003) has already found wide industrial usage. The SIMC rules are analytically derived, and from a first or second order process we can easily find the PI and PID controller setting, respectively. Even though the rule was originally derived mainly with simplicity in mind, recent studies have found that the resulting settings are very close to optimal (Grimholt and Skogestad, 2012a, 2013). For the double integrating process, the SIMC rule gives the following PID settings for the serial form in (4.2):

$$k_c = \frac{1}{k''} \frac{1}{4(\tau_c + \theta)^2}, \quad \tau_i = 4(\tau_c + \theta), \quad \tau_d = 4(\tau_c + \theta). \tag{4.3}$$

The SIMC rule has one tuning parameter τ_c which can be used to trade off between performance (favored by small τ_c) and robustness (favored by large τ_c). For most processes, the recommended value for "tight control" (good performance subject to acceptable robustness) is $\tau_c = \theta$, but, as we will see, a value closer to 1.5θ may be better for the double integrating process.

There are many industrial and mechanical systems that have double integrating behavior. Furthermore, the double integrating process is a special case of second-order processes

$$G(s) = \frac{ke^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)}, \tag{4.4}$$

with

$$k'' = k/(\tau_1 \tau_2). \tag{4.5}$$

The original SIMC PID tunings for a second-order process are (Skogestad, 2003)

$$\begin{aligned} k_c &= \frac{1}{k}\frac{\tau_1}{(\tau_c + \theta)}, \\ \tau_i &= \min\{\tau_1,\ 4(\tau_c + \theta)\}, \quad \tau_d = \tau_2. \end{aligned} \tag{4.6}$$

The SIMC rule in (4.6) does not apply to double integrating processes, but by considering the SIMC rule for double integrating process in (4.3), we can generalize (4.6) to get a single SIMC PID-rule which covers all second-order processes:

$$\begin{aligned} k_c &= \frac{1}{k}\frac{\tau_1}{(\tau_c + \theta)}\frac{\tau_2}{\tau_d}, \\ \tau_i &= \min\{\tau_1,\ 4(\tau_c + \theta)\}, \quad \tau_d = \min\{\tau_2,\ 4(\tau_c + \theta)\}. \end{aligned} \tag{4.7}$$

For processes with $\tau_1 > \tau_2 > 4(\tau_c + \theta)$ these settings are identical to those for the double integrating process in (4.3). Thus, a

Table 4.1: Reference and optimal PID controllers for double integrating processes ($k'' = 1$ and $\theta = 1$) with $M_S = 1.59$.

	k_c	τ_i	τ_d	IAE$_{dy}$	IAE$_{du}$	J	M_S	M_T	$1/\text{GM}_l$	GM	DM
Output wheight	0.02	∞	24.13	**4.15**	∞	∞	1.59	1.09	∞	3.20	2.07
Input wheight	0.04	10.12	10.12	6.28	**288.56**	1.257	1.59	1.71	5.79	4.02	1.80
Optimal PID	0.04	10.74	10.79	5.80	303.68	1.225	1.59	1.61	6.72	3.76	1.78

IAE$_{dy}$ and IAE$_{du}$ are for a unit step disturbance on output (y) and input (u), respectively.

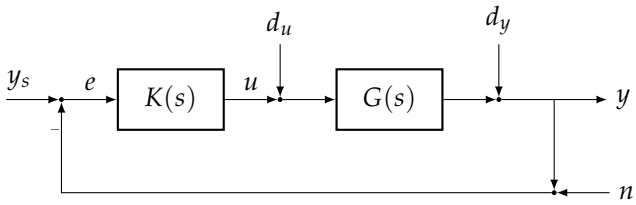

Figure 4.1: Block diagram of the feedback control system. We do not consider setpoint changes (y_s) or noise (n).

second-order process with large time constants τ_1 and τ_2, may be represented as a double integrating process.

It is generally difficult to achieve good performance for a double integrating process if the time delay θ is large, especially for disturbances at the input (d_u) which result in ramp deviations at the output (y). Because a ramp increases with t^2, the achievable IAE of the output increases proportionally with θ^3 (Skogestad, 2003) for a double integrating process, resulting in poor performance for double integrating processes with a large delay. Thus, in practice, cascade control is used for control of double integrating processes whenever possible, which results in control of two integrating processes (for which IAE only increases with θ^2). For example, for mechanical systems, the process from u = force (acceleration) to y = position is a double integrating process, but by measuring and controlling also y_2 = velocity, we instead get two integrating processes. Rao and Bernstein (2001) study control of a double integrator with input saturation. They mostly assume full state feedback, which requires two measurements, similar to the use of cascade control. However, in this chapter we consider the case where only the output y is measured and the process is double integrating.

Because many important industrial processes can be classified as

double integrating, we want in this chapter to investigate optimal PID control and the optimality of SIMC for this type of processes. Optimality is generally difficult to define as there are many issues to consider, including:

- Output performance
- Stability robustness
- Input usage
- Noise sensitivity

This may be considered a multi-objective optimization problem, but we consider only the main dimension of the trade-off space, namely high versus low controller gain. High controller gain favors good output performance, whereas low controller gain favors the three other objectives listed above. We can then simplify and say that there are two main objectives:

1. Performance (here measured in terms of IAE)
2. Robustness (here measured in terms of M_s-value)

Pareto optimality applies to multi-objective problems, and implies that no further improvement can be made in objective 1 without sacrificing objective 2. The idea is then to find the Pareto optimal controller, and compare with the SIMC tuning.

The chapter is structured as follows. First the evaluation criteria and the optimization problem are defined. Then the optimal trade-off between performance and robustness is found and compared with SIMC. Following this, is a small comparison between serial and parallel PID controller. The chapter ends with a time domain comparison between the different controllers, and a discussion.

4.2 Evaluation criteria

Performance

In this chapter we choose to quantify performance terms of the integral absolute error (IAE),

$$\text{IAE} = \int_0^\infty \left| y(t) - y_s(t) \right| \mathrm{d}t. \tag{4.8}$$

To balance the servo/regulatory trade-off, we choose as the performance index a weighted average of IAE for a step input disturbance d_u and step output d_y,

$$J(p) = 0.5 \left(\frac{\text{IAE}_{dy}(p)}{\text{IAE}^{\circ}_{dy}} + \frac{\text{IAE}_{du}(p)}{\text{IAE}^{\circ}_{du}} \right) \tag{4.9}$$

where IAE°_{dy} and IAE°_{du} are weighting factors, and p is the controller parameters. In this chapter, we select the two weighting factors as the optimal IAE values when using PID control, for input and output disturbances, separately (as recommended by Boyd and Barratt (1991)). To ensure robust reference PID controllers, they are required to have $M_s = 1.59$, and the weighting factors are $\text{IAE}^{\circ}_{dy} = 4.15$ and $\text{IAE}^{\circ}_{du} = 288.56$ (see Table 4.1).

As seen from Table 4.1, the optimal PID controller for combined input and output disturbances (J) favors input disturbances, and is almost identical to the optimal controller when only considering input disturbance (d_u). Therefore, it would be sufficient for double integrating plus delay process to only consider input disturbances. Nevertheless, to keep this analysis similar to other studies we have conducted on optimal controller tuning (Grimholt and Skogestad, 2012a, 2013), we have chosen to include both input and output disturbances in the cost function.

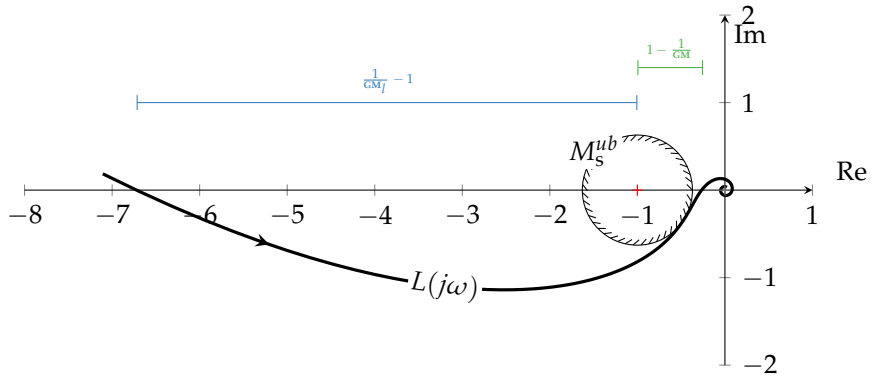

Figure 4.2: Nyquist plot of the optimal PID controller for the control of double integration process (4.1) with $k'' = 1$ and $\theta = 1$ and robustness $M_s^{ub} = 1.59$ (Table 4.1). The robustness constraint M_s^{ub} and the gain margins are marked.

Robustness for design: M_s

In this chapter, we quantify robustness in terms of M_s, defined as

$$M_s = \max_{\omega} \left| S(j\omega) \right| = \left\| S(j\omega) \right\|_\infty , \tag{4.10}$$

where $\|\cdot\|_\infty$ is the H_∞ norm (maximum peak as a function of frequency), and the sensitivity transfer functions are defined as

$$S(s) = 1/\big(1+G(s)\,K(s)\big) \quad \text{and} \quad T(s) = 1 - S(s). \tag{4.11}$$

In robustness terms, M_s is the inverse of the closest distance from the loop function $L = GK$ to the critical point, -1, in a Nyquist plot (see Figure 4.2). Originally, we considered using the largest peak of *S and T*, denoted M_{ST}, as the robustness criterion, but as discussed later we decided to use M_s.

Robustness for analysis: gain and delay margin

In addition to the M_s value, we consider for robustness analysis the gain margin (GM) and the delay margin (DM), which have a clear physical meaning. The GM is defined as the factor by which we can multiply the controller gain (or more generally, the loop gain) before getting instability. Actually, as illustrated in Figure 4.2 and Figure 4.3, there are two gain margins in our case. The "normal" gain margin (GM) is the factor by which we can increase the loop gain, and the "lower" gain margin (GM_l) is the factor by which we can *decrease* the loop gain. For stability we need $\text{GM} > 1$ and $\text{GM}_l < 1$, but for acceptable robustness we typically want $\text{GM} > 3$ and $\text{GM}_l < 0.33$. In the Tables, we show $1/\text{GM}_l$ (the factor by which the gain can be reduced) which typically should be larger than 3.

The delay margin is the allowed increase in delay in the feedback loop, $\Delta\theta_{\text{MAX}}$, before we get instability. Note that $\Delta\theta_{\text{MAX}} = \text{PM}/\omega_c$ where PM [radians] is the phase margin and ω_c [rad/s] is the gain crossover frequency. In this chapter, we will use the relative delay margin, defined as $\text{DM} = \Delta\theta_{\text{MAX}}/\theta$.

Optimal trade-off

The optimal PID controllers are found by solving the following optimization problem,

$$\min_p \quad J(p) = 0.5\left(\frac{\text{IAE}_{dy}(p)}{\text{IAE}_{dy}^{\circ}} + \frac{\text{IAE}_{du}(p)}{\text{IAE}_{du}^{\circ}}\right) \tag{4.12}$$

$$\text{subject to:} \quad M_s(p) \leq M_s^{ub} \tag{4.13}$$

where p is the parameters of a PID controller K_{PID}. For more details on how to solve the optimization problem, see Grimholt and Sko-

gestad (2018a). To find the optimal trade-off between performance (J) and robustness (M_s), the optimization problem is solved repeatedly with different upper limits on the robustness (M_s^{ub}).

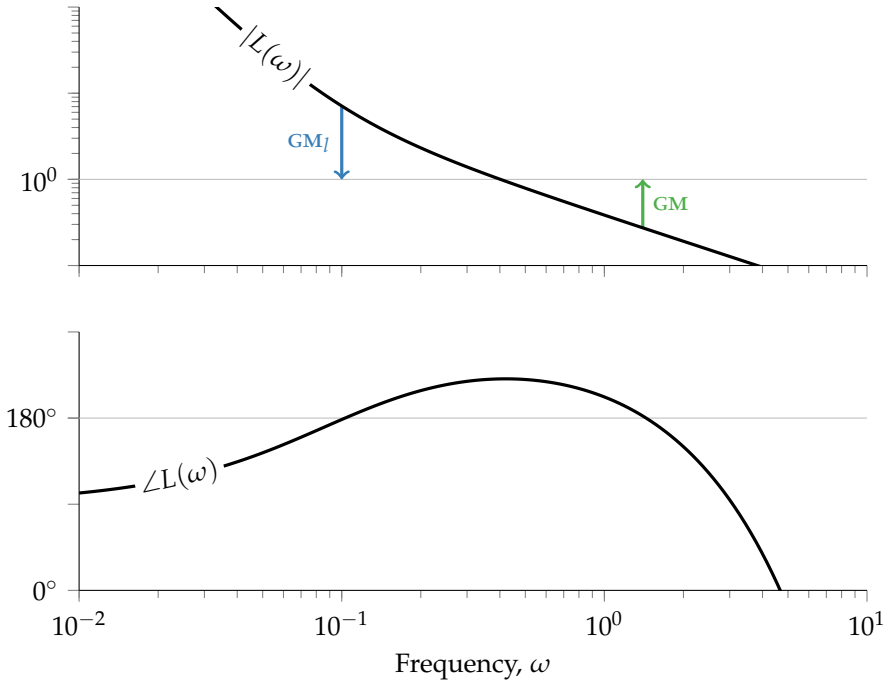

Figure 4.3: Bode plot of the optimal PID controller for the control of double integration process (4.1) with $k'' = 1$ and $\theta = 1$ and robustness $M_s^{ub} = 1.59$ (Table 4.1). The gain margins are marked.

Table 4.2: Optimal and SIMC for double integrating processes ($k'' = 1$ and $\theta = 1$) with robustness $M_s = 1.40$, 1.59, 1.80, and 2.00.

	$k_c k'' \theta^2$	τ_i/θ	τ_d/θ	IAE_{dy}/θ	$\text{IAE}_{du}/k''\theta^3$	J	M_s	M_T	$1/\text{GM}_l$	GM	DM	τ_c/θ
Optimal PID	0.0209	13.67	13.67	7.25	653.1	2.006	1.40	1.56	6.69	5.14	2.62	–
SIMC	0.0209	13.84	13.84	7.16	662.7	2.011	1.40	1.55	6.86	5.08	2.62	2.46
Optimal PID	0.0354	10.74	10.79	5.80	303.7	1.225	1.59	1.61	6.72	3.76	1.78	–
SIMC	0.0356	10.60	10.60	5.88	297.7	1.225	1.59	1.63	6.52	3.80	1.78	1.65
Optimal PID	0.0505	9.37	9.38	4.96	185.7	0.919	1.80	1.66	7.02	2.99	1.32	–
SIMC	0.0512	8.84	8.84	5.24	172.7	0.931	1.80	1.74	6.23	3.10	1.31	1.21
Optimal PID	0.0625	8.64	8.76	4.46	138.2	0.777	2.00	1.72	7.39	2.55	1.06	–
SIMC	0.0651	7.84	7.84	4.91	120.5	0.801	2.00	1.86	6.01	2.70	1.05	0.96

IAE_{dy} and IAE_{du} are for a unit step disturbance on output (y) and input (u), respectively.

4.3 Optimal PID settings and comparison with SIMC

The optimal and SIMC PID controllers are given in Table 4.2 for four values of M_s (1.4, 1.59, 1.8 and 2). The Pareto-optimal trade-off between performance (J = IAE) and robustness (M_s) is shown in Figure 4.4 (green curve) and compared with the SIMC PID-controller (blue curve). The trade-off curves for the SIMC controllers were generated by varying the tuning parameter τ_c from a large to a small value. The SIMC controllers corresponding to three specific choices are shown by circles:

- $\tau_c = 1.5\theta$ (smoother tuning)
- $\tau_c = \theta$ (default tight tuning)
- $\tau_c = 0.5\theta$ (more aggressive tuning)

For all robustness levels (in terms of M_s), we find that the SIMC rule is very close to the optimal. However, for the normally recommended tuning ($\tau_c = \theta$), M_s is quite high, being close to $M_s = 2$. A better value for the SIMC PID tuning constant in (4.3) is $\tau_c = 1.5\theta$ which gives $M_s = 1.65$.

The corresponding optimal and SIMC PID tuning parameters are shown in Figure 4.5 as a function of the robustness M_s. We find that the optimal PID-controller (serial form) always has $\tau_i = \tau_d$. The results also show that in the more robust region ($M_s < 1.6$), the SIMC tuning parameters are almost identical to the optimal PID controller. In the less robust region with higher performance ($M_s > 1.6$), the SIMC controller gain is slightly higher and the integral and derivative time slightly smaller than the optimal. However, as seen from Figure 4.4, this deviation from the optimal PID parameters has little effect on performance.

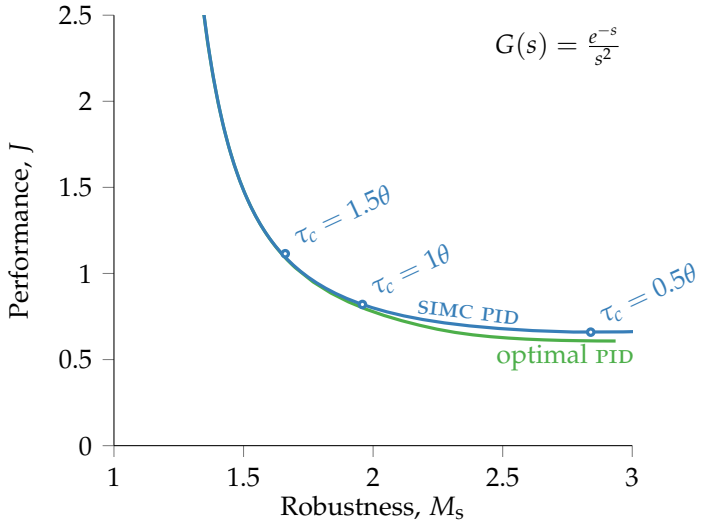

Figure 4.4: Trade-off between performance and robustness for Pareto optimal PID controllers (green curve) and SIMC PID (blue curve).

4.4 Parallel vs. serial PID controller

The analysis in this chapter is for the serial PID controller in (4.2). A more general PID controller is the parallel, or ideal, PID controller which allows for complex zeros,

$$K_{\text{PID}}^{\text{PARALLEL}} = \tilde{k}_c \left(1 + \frac{1}{\tilde{\tau}_i s} + \tilde{\tau}_d s\right). \tag{4.14}$$

Parallel PID controller parameters (4.14) can be calculated from serial PID controller parameters (4.2) by

$$f = 1 + \tau_d/\tau_i,\ \tilde{k}_c = k_c f,\ \tilde{\tau}_i = \tau_i f, \text{ and } \tilde{\tau}_d = \tau_d/f. \tag{4.15}$$

From Figure 4.5, we see that the optimal serial controller has equal integral and derivative times, which means that $f = 2$ and that

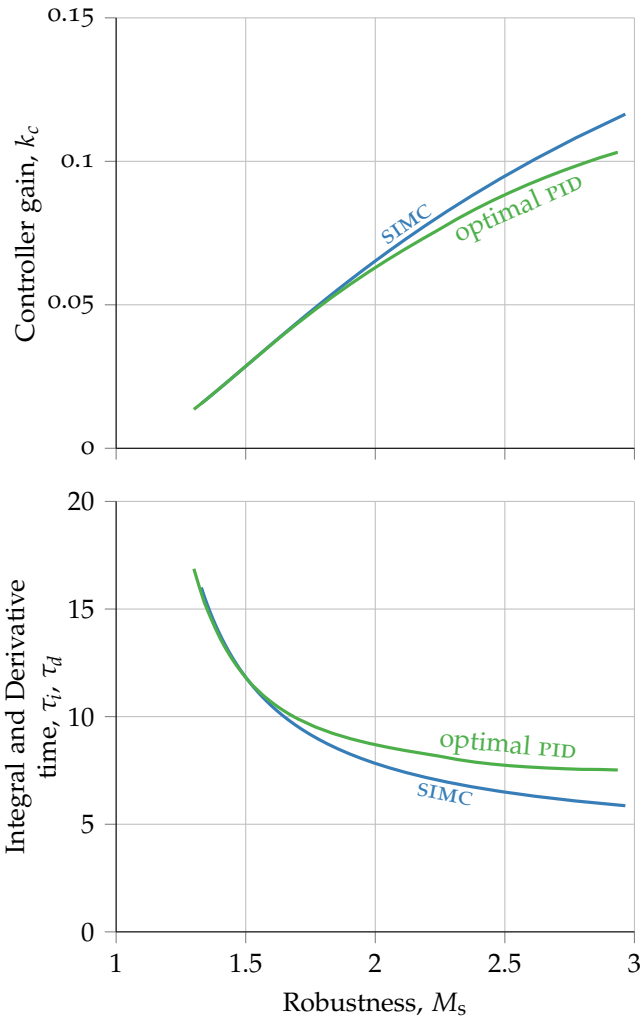

Figure 4.5: Optimal and SIMC PID tuning parameters as a function of robustness for control of double integration process (4.1) with $k'' = 1$ and $\theta = 1$. For both controllers $\tau_i = \tau_d$.

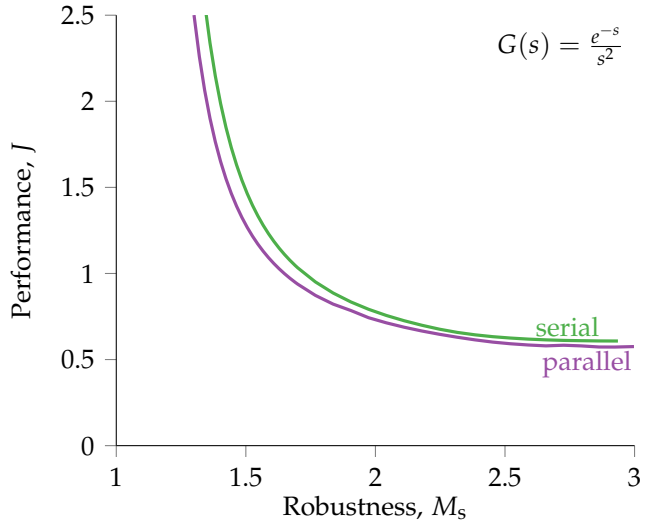

Figure 4.6: Pareto optimal trade-off curves for optimal serial PID controller (the form used in this chapter) and optimal parallel PID controller.

we are just at the limit to having complex zeros. This indicates that better performance can be achieved by allowing for complex zeros by using a parallel PID controller.

A comparison of the optimal trade-off curves for serial and parallel PID controllers is shown in Figure 4.6. Although the benefit of using the parallel PID form increases with increasing robustness, we see that the overall improvement is quite small. Thus, the serial implementation is sufficient for double integrating plus delay processes. A selection of optimal tunings are given in Table 4.3 with corresponding gain and delay margins.

Table 4.3: Optimal parallel PID controllers of double integrating processes ($k'' = 1$ and $\theta = 1$) with robustness $M_S = 1.40$, 1.59, 1.80, and 2.00.

$\tilde{k}_c k''\theta^2$	$\tilde{\tau}_i/\theta$	$\tilde{\tau}_d/\theta$	IAE_{dy}/θ	$\text{IAE}_{du}/k''\theta^3$	J	M_S	M_T	$1/\text{GM}_l$	GM	DM
0.0416	16.39	7.17	7.88	411.6	1.663	1.40	1.64	4.22	4.95	2.63
0.0694	13.39	5.76	6.17	198.4	1.088	1.59	1.64	4.46	3.62	1.80
0.0974	11.98	5.09	5.21	126.1	0.847	1.80	1.63	4.81	2.89	1.34
0.1215	11.28	4.68	4.70	94.7	0.731	2.00	1.68	5.10	2.49	1.08

IAE_{dy} and IAE_{du} are for a unit step disturbance on output (y) and input (u), respectively.

4.5 Simulations

The responses to setpoint change and input disturbance for optimal serial, optimal parallel and SIMC PID controllers with M_s-value 1.40, 1.59, and 1.80 are shown in Figure 4.7. The corresponding tuning parameters are given in Tables 4.2 and 4.3.

As expected from the trade-off curves, the responses to the input disturbance are similar for the three controllers. The optimal parallel and serial PID controllers have almost the same peak deviation, but the parallel controller has better settling time. The SIMC and optimal PID controllers have almost identical responses.

4.6 Discussion

Comparison with previous work

There is relatively little work on PID control of double integrating processes. Shamsuzzoha and Lee (2008) use IMC as a basis for designing PID controllers with λ (equivalent to τ_c) as the tuning parameter. For the double integrating process $G = e^{-0.8s}/s^2$ and $\lambda = 1.25$ they obtain a parallel PID controller with $\tilde{k}_c = 0.3510$, $\tilde{\tau}_i = 5.880$, and $\tilde{\tau}_d = 2.343$. In the more general form the parameters are $\tilde{k}_c k'' \theta^2 = 0.2246$, $\tilde{\tau}_i/\theta = 7.35$, and $\tau_d/\theta = 2.93$. This controller has poor robustness with M_s= 2.87. In terms of our IAE performance measure, it gives $J = 0.69$ which is a little above the optimal $J = 0.57$ in Figure 4.6. Shamsuzzoha and Lee (2008) claim that their PID controller is significantly better than the SIMC PID controller, but this is incorrect. The reason for the error is that they follow Liu et al. (2004) who failed to use (4.15) to translate the double integrating SIMC-settings (4.3) from serial to parallel form.

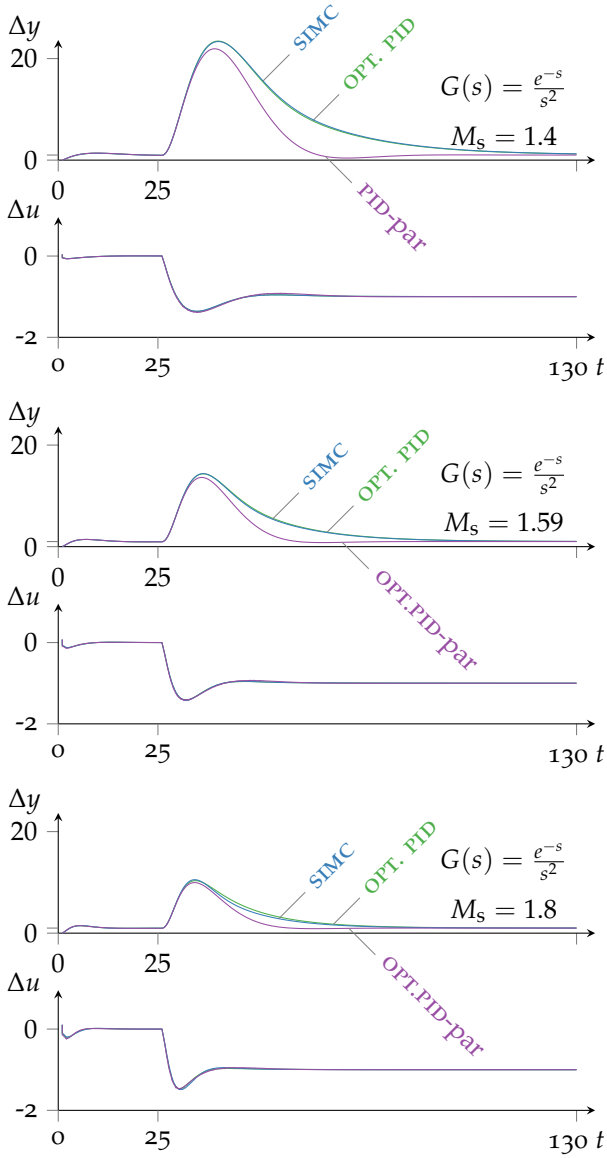

Figure 4.7: Step responses to a setpoint step change ($t = 0$), and an input disturbance ($t = 25$) for the optimal PID, optimal parallel PID, and SIMC PID controllers with robustness $M_s = 1.4$ (top), $M_s = 1.59$ (middle), and $M_s = 1.8$ (bottom).

More recently, Hassaan (2015) has considered optimal PID control of double integrating processes using the serial PID structure (which he calls the PD-PI controller). He considers a variety of performance objectives, including IAE, integrated time absolute error (ITAE) and ISE. However, he includes no robustness requirement, which means that his "optimal" solution has poor robustness. It would correspond to the largest M_S-value (where J has its minimum) on our robustness-performance trade-off plots.

Criteria for robustness

In this chapter, we quantify robustness in terms of M_S, defined as the peak value of S. Actually, we originally considered using the largest value of M_S and M_T (Garpinger and Hägglund, 2008),

$$M_{ST} = \max\{M_S, M_T\}, \tag{4.16}$$

where

$$M_T = \max_{\omega} \left|T(j\omega)\right| = \left\|T(j\omega)\right\|_\infty .$$

For most stable processes, $M_S \geq M_T$, but for unstable process, including the double integrating process, it may happen that $M_T > M_S$, and this is why we originally used the largest value of M_S and M_T (Garpinger and Hägglund, 2008) as the robustness criterion.

As seen from the (Figure 4.4), performance (J) gets very poor when M_S approaches about 1.3, and the corresponding value for M_T is a little higher. Thus, there will be problems when trying to specify too low values for M_S or M_T. In particular, we encountered this problem with the SIMC controller when specifying low values for M_T. For example, when specifying $M_T = 1.4$ and using SIMC, we had to increase τ_c/θ to 32.5 (compared to 2.46 with $M_S = 1.4$),

resulting in very poor performance with $J = 58.9$ (compared to 2.85). However, as noted, specifying low values for M_s was much less of a problem and resulted in reasonable designs. Also, when we analyzed more carefully the results, we could not see that the lower value for M_T was giving any benefit in terms of improved upper and lower gain margins and delay margins. We therefore decided to base the robustness criterion on M_s only.

Generalized SIMC - for second-order processes

As noted in the introduction, second-order processes with sufficiently large time constants ($\tau_2 > 4(\tau_c + \theta)$) should be approximated as double integrating processes. For example, consider the process

$$G = \frac{40e^{-s}}{(20s+1)^2}. \tag{4.17}$$

If we use the original SIMC rules for a second-order process in (4.6) then $\tau_d = 20$, and the derivative time will be larger than the integral time in most cases. That is, with $\tau_c = \theta = 1$, we get $\tau_i = 8$. However, if we use the "generalized" rule in (4.7), which is equivalent to representing the process as a double integrating process with $k'' = 40/20^2 = 0.1$, then we get $\tau_d = \tau_i = 8$.

To confirm that this gives better performance, consider the trade-off curve in Figure 4.8.

We see that the PID controller based on the double integrating process, that is, using the generalized SIMC settings in (4.7), is almost identical to the optimal PID controller, whereas the PID controller based on a second-order process, using the standard SIMC settings in (4.6), has significantly poorer performance for input disturbances. For output changes (and setpoint changes) the standard

settings in (4.6) are a little better, but not significantly. This is also illustrated by the simulations in Figure 4.9.

4.7 Conclusion

In this chapter we have derived optimal PID controller settings for a double integrating the process and compared the performance versus robustness trade-off with that obtained when varying the tuning parameter τ_c for the SIMC-controller in (4.3). As seen from Figure 4.4, the SIMC controller has almost identical performance with the optimal, in particular for more robust designs (with lower value of M_s). This means that the simple SIMC PID tuning rules given in

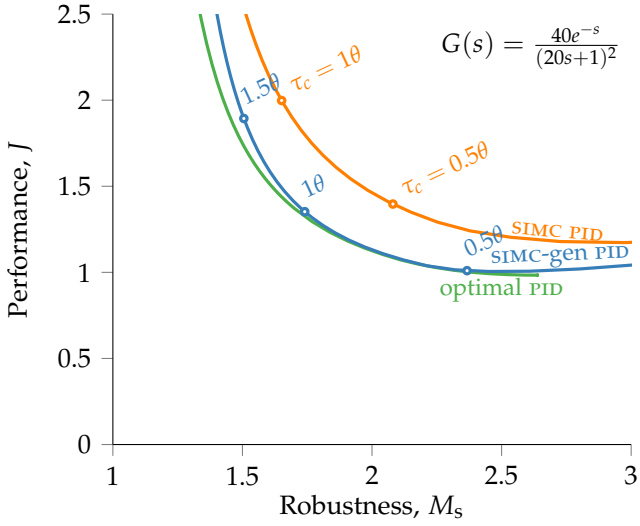

Figure 4.8: Trade-off plot for Pareto optimal PID controller, SIMC PID in (4.6) and generalized SIMC PID controller in (4.7) of a almost double integrating second-order process.

(4.3) are essentially the optimal. This is quite surprising, because the double integrating SIMC rules were originally derived in a fairly *ad hoc* manner, aiming more towards simplicity than optimality.

We also find that for PID tuning, a second-order process (4.4) with $\tau_2 > 4(\tau_c + \theta)$ should be approximated as a double integrating process (4.1) with gain $k''=k/(\tau_1\tau_2)$. Alternatively and equivalently, we may use the "generalized" SIMC rules in (4.7).

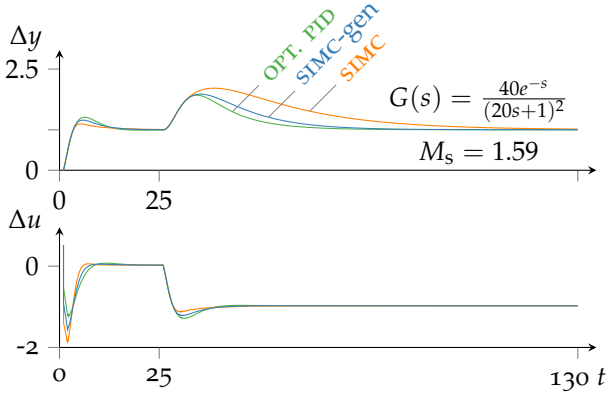

Figure 4.9: Step responses to a setpoint step change ($t = 0$), and an input disturbance ($t = 25$) for the optimal PID controller, SIMC rule (4.6) second-order processes, and generalized SIMC (4.7) for second-order processes, all with robustness $M_s = 1.59$.

Chapter Five

Should we forget the Smith predictor?

The PI/PID controller is the most used controller in industry. However, for processes with large time delays, the common belief is that PI and PID controllers have sluggish performance, and that a Smith predictor or similar dead-time compensator can give much improved performance. We claim in this chapter that this is a myth. For a given robustness level in terms of the peak sensitivity (M_s), we find that the performance improvement with the Smith predictor is small even for a pure time delay process. For other first-order processes a PID controller is generally better for a given robustness level. In addition, the Smith Predictor is much more sensitive to time delay errors than PI and PID controllers.

5.1 Introduction

We find time delays in most industrial processes $G(s)$, see Figure 5.1. Time delay is an important aspect to consider when applying feedback control because it imposes serious limitations on the performance (Skogestad and Postlethwaite, 2005). For the controller $K(s)$, we mostly use the proportional-integral (PI) controller, which is the work-horse of the process industry with more that 95% of all control application being of this type (Åström et al., 1995). For integrating processes with large time delays, PI control is somewhat sluggish, and to improve performance we can add derivative action, i.e. using PID control (Grimholt and Skogestad, 2013).

An alternative to PID control, is to make use of a SP, also known as a dead time compensator, (Smith, 1957). The SP uses a model of the process without a time delay to predict the process output, and this new process is controlled by a conventional controller, for example, a PI controller. See Figure 5.2. The SP controller has good setpoint response because it removes the internal delays from the closed-loop transfer function. One drawback is that it has poor performance for input (load) disturbances for processes with slow dynamics because the original open-loop process poles remain unchanged. However, this can be rectified by alternative designs, e.g. see Normey-Rico and Camacho (2007).

(Kristiansson and Lennartson, 2001; Ingimundarson and Hägglund, 2002; Larsson and Hägglund, 2012) have compared the performance of PI, PID, and SP controllers. These papers investigate performance for load disturbances for a fixed robustness. Most of these papers conclude that PID has better performance than SP, and our work further confirms this.

In this chapter, we examine the optimal performance-robustness

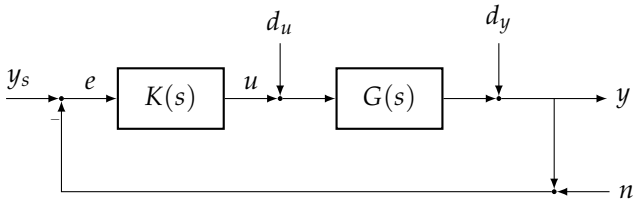

Figure 5.1: Block diagram of the feedback control system. We may treat an output disturbance (d_y) as a special case of setpoint change (y_s)

trade-off for a SP and compare it to the optimal trade-off for PI and PID with the same robustness level. For performance we consider an average IAE performance of step disturbances at the process input and output. The disturbance at the process output is a special case of set-point response. Robustness is quantified by the peak of the sensitivity and complimentary sensitivity function (M_S and M_T). In addition, we consider time delay robustness which is not captured by M_S and M_T.

5.2 The feedback systems

We consider a range of FOPD processes

$$G = \frac{ke^{-\theta s}}{\tau s + 1} = G_\circ e^{-\theta s}, \tag{5.1}$$

where k, τ, and θ is the gain, time constant, and time delay of the process, respectively. The process without time delay is represented with $G_\circ$.

PID control

There are several different parameterizations of the PI and PID controller. For optimization purposes, we use the following linear parametrization

$$K_{\text{PI}} = k_p + k_i/s \quad \text{and} \quad K_{\text{PID}} = k_p + k_i/s + k_d s, \tag{5.2}$$

We can transform the controller (5.2) to the standard parallel ("ideal") PID controller,

$$K_{\text{PID}}^{\text{PARALLEL}} = \tilde{k}_c \left(1 + \frac{1}{\tilde{\tau}_i s} + \tilde{\tau}_d s\right) \tag{5.3}$$

by the following transformation

$$\tilde{k}_c = k_p, \quad \tilde{\tau}_i = {}^{k_p}/_{k_i}, \quad \text{and} \quad \tilde{\tau}_d = {}^{k_d}/_{k_p}. \tag{5.4}$$

The Smith Predictor

In this chapter we consider the "original" Smith predictor controller

$$K_{\text{SP}} = \frac{K_\circ(s)}{1 + K_\circ(s)G_\circ(s)\left(1 - e^{-\theta s}\right)} \tag{5.5}$$

where $K_\circ$ in the primary conventional controller, which in this chapter is a PI controllers, and $G_\circ$ and G are the internal delay-free and delayed models, respectively. A block diagram of the SP for the case where $K_\circ$ is a PI controller is shown in Figure 5.2.

The main advantage of the SP is the potential excellent setpoint response. However, because it is impossible to eliminate the open-loop poles in the input disturbance (load disturbance) transfer function, the "original" SP has slow settling time for for input disturbances (load disturbances). It is possible to improve this by using a modified SP as discussed later.

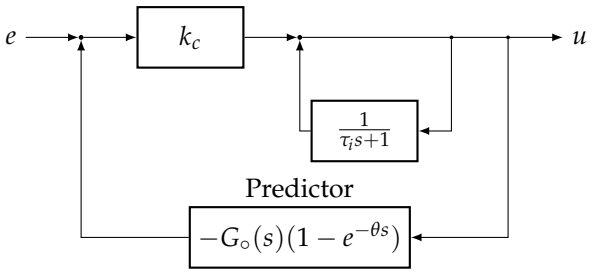

Figure 5.2: Block digram of the SP controller, K_{SP}, when $K_\circ$ is a PI controller.

5.3 Quantifying the optimal controller

Performance

In this chapter we choose to quantify performance in terms of the IAE,

$$\text{IAE} = \int_0^\infty \left| y(t) - y_s(t) \right| \mathrm{d}t. \tag{5.6}$$

To balance the servo/regulatory trade-off, we choose as the performance index a weighted average of IAE for a step input disturbance d_u and step output d_y,

$$J(p) = 0.5 \left(\frac{\text{IAE}_{dy}(p)}{\text{IAE}^\circ_{dy}} + \frac{\text{IAE}_{du}(p)}{\text{IAE}^\circ_{du}} \right) \tag{5.7}$$

where IAE°_{dy} and IAE°_{du} are weighting factors, and p is the controller parameters. In this chapter, we select the two weighting factors as the optimal IAE values when using PI control, for input and output disturbances, respectively (as recommended by Boyd and Barratt (1991)). To ensure robust reference PI controllers, they are required

to have $M_S = 1.59^*$, and the resulting weighting factors are given for four processes in Table 5.3.

It may be argued that a two-degree of freedom controller with a setpoint filter can be used to enhance setpoint performance, and thus we only need to consider input disturbances. But note that although a change on the output d_y, is equivalent to a setpoint change y_s for the system in Figure 5.1, it is not affected by a setpoint filter. Thus, we consider disturbance rejection which can only be handled by the feedback controller $K(s)$ (Figure 5.1).

Robustness

In this chapter, we quantify robustness in terms of M_{ST}, defined as the largest value of M_S and M_T (Garpinger and Hägglund, 2008),

$$M_{ST} = \max\{M_S, M_T\}. \tag{5.8}$$

where M_S and M_T are the largest peaks of the sensitivity $S(s)$ and complimentary sensitivity $T(s)$ functions, respectively. Mathematically,

$$M_S = \max_\omega \left|S(j\omega)\right| = \left\|S(j\omega)\right\|_\infty,$$
$$M_T = \max_\omega \left|T(j\omega)\right| = \left\|T(j\omega)\right\|_\infty,$$

where $\|\cdot\|_\infty$ is the H_∞ norm (maximum peak as a function of frequency), and the sensitivity transfer functions are defined as

$$S(s) = 1/\big(1+G(s)\,K(s)\big) \quad \text{and} \quad T(s) = 1 - S(s). \tag{5.9}$$

---

* For those that are curious about the origin of this specific value $M_S = 1.59$, it is the resulting M_S value for a SIMC tuned PI controller with $\tau_c = \theta$ on FOPTD process with $\tau \leq 8\theta$.

Table 5.1: Cost function weights and optimal controllers with $M_{ST} = 1.59$.

	Weights in J		Optimal PI					Optimal SP					Optimal PID					
Process	IAE°_{dy}	IAE°_{du}	k_c	τ_i	IAE_{dy}	IAE_{du}	J	k_c	τ_i	IAE_{dy}	IAE_{du}	J	$\tilde{k}_c$	$\tilde{\tau}_i$	$\tilde{\tau}_d$	IAE_{dy}	IAE_{du}	J
e^{-s}	1.61	1.61	0.20	0.32	1.61	1.61	1.00	0.73	0.32	1.45	1.45	0.90	0.20	0.32	—	1.61	1.61	1.00
$e^{-s}/(s+1)$	2.07	2.02	0.54	1.10	2.08	2.04	1.01	1.37	0.93	1.69	1.68	0.83	0.86	1.13	0.34	1.58	1.44	0.74
$e^{-s}/(8s+1)$	2.17	1.13	3.47	4.04	3.10	1.16	1.23	9.94	3.35	2.04	1.38	1.08	5.16	3.23	0.41	2.29	0.66	0.82
$e^{-s}/(20s+1)$	2.17	0.60	8.42	5.16	3.67	0.61	1.36	22.7	3.31	2.34	1.12	1.47	12.6	3.70	0.41	2.64	0.31	0.87

IAE_{dy} and IAE_{du} are for a unit step disturbance on output (y) and input (u), respectively.

For most stable processes, $M_S \geq M_T$. For a given M_S we are guaranteed the following GM and PM,

$$\text{GM} \geq \frac{M_S}{M_S - 1} \quad \text{and} \quad \text{PM} \geq 2 \arcsin\left(\frac{1}{2M_T}\right) \geq \frac{1}{M_T}. \tag{5.10}$$

For example, $M_S = 1.6$ guarantees GM ≥ 2.7 and PM $\geq 36.4^\circ = 0.64$ rad. Another important robustness measure is the DM, (Åström and Hägglund, 2006),

$$\text{DM} = \frac{\text{PM}}{\omega_c} \tag{5.11}$$

where ω_c is the crossover frequency. Note that the units for PM [rad] and ω_c [rad/s] must be consistent. The delay margin is the smallest change in time-delay that will cause the closed-loop to becomes unstable. We will see for the SP, that robustness in term of M_{ST} does not guarantee robustness in term of DM.

5.4 Optimal controller

For a given first-order plus delay process, the IAE-optimal PI, PID or SP controller are found for a specified robustness level by solving the following optimization problem:

$$\min_p \quad J(p) = 0.5 \left(\frac{\text{IAE}_{dy}(p)}{\text{IAE}^\circ_{dy}} + \frac{\text{IAE}_{du}(p)}{\text{IAE}^\circ_{du}} \right) \tag{5.12}$$

$$\text{subject to:} \quad M_S(p) \leq M^{ub} \tag{5.13}$$

$$M_T(p) \leq M^{ub} \tag{5.14}$$

where the two or three parameters in p are for a PI or PID controller. Our Smith predictor controller always uses a PI controller (two parameters in p, see Figure 5.2) For more details on how to solve the

optimization problem, see Grimholt and Skogestad (2018a). The problem is solved repeatedly for different values of M^{ub}. One of the bounds in (5.13) or (5.14) will be active if there is a trade-off between robustness and performance. This is the case for values of M^{ub} less than about 2 to 3.

5.5 Optimal trade-off

In this section, we present the optimal IAE-performance for PI, PID and SP controllers as a function of the robustness level M_{ST}. We have considered four FOPD process models, defined in (5.1),

Pure time delay: $\tau/\theta = 0$
Balanced dynamics: $\tau/\theta = 1$
Lag dominant: $\tau/\theta = 8$
Close to integrating: $\tau/\theta = 20$

We have not included results for an integrating process model because SP does not work for integrating processes.

Performance

In Figure 5.3, we show the Pareto optimal IAE-performance (J) as a function of robustness (M_{ST}) for the four processes.

For the pure time delay process (top left), there is only a small difference between the optimal PI/PID and SP controllers. For very robust controllers with $M_{ST} = 1.1$, the improvement with SP is only 2%. For robust controllers with $M_{ST} = 1.69$, the improvement is 11%. This small improvement is surprising, because SP was expected to substantially improve performance for a delay-dominant processes.

Note that we write optimal "PI/PID" controller. This is because for a pure time delay process there is no advantage in using deriva-

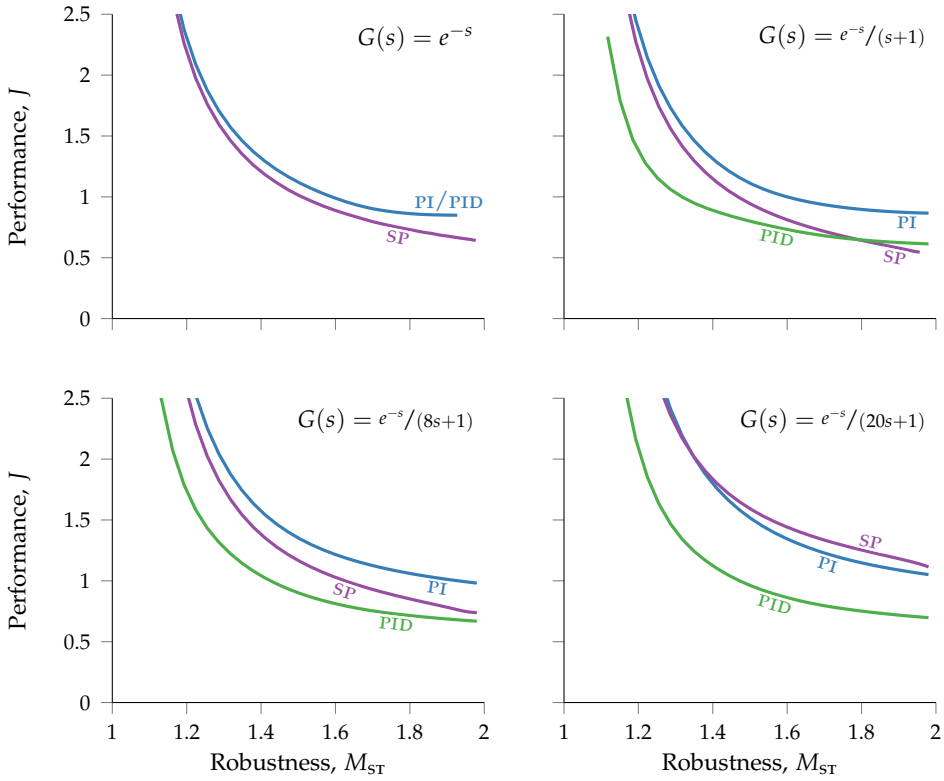

Figure 5.3: Pareto-optimal trade-off between IAE-performance and M_{ST}-robustness for PI, PID and SP control.

tive action, so the optimal PID is a PI controller. Note that the maximum value of M_{ST} is 1.92 for PI and PID control. Performance (J) actually gets worse if M_{ST} is increases beyond this value, and this region should be avoided.

For balanced and lag dominant dynamics (top right and bottom left in Figure 5.3), SP has somewhat better performance than PI. For a robust controller with $M_{ST} = 1.69$, the SP has 23% and 17% improved performance, respectively. However, the PID controller is even better, with 27% and 33% improvement relative to the PI controller.

For a close to integrating process (bottom right Figure 5.3), SP has worse performance than the PI controller. This is because the "original" SP retains the large time constant from the open-loop process model (Hägglund, 1996). For a robust controller with $M_{ST} = 1.69$, the SP controller has -8% worse performance than PI. On the other side, the PID controller has an improved performance of 35%.

In summary, if we consider performance for a given robustness level in terms of M_{ST}, the PID controller is better than the SP with PI, except for a pure time delay process where the potential improved performance is marginal, especially for the interesting cases with M_S less than 1.6.

Delay margin (DM)

In the above discussion the trade-off between IAE performance and M_{ST} robustness, we did not consider the delay error which for SP is not captured by the M_{ST} value and which is actually the main disadvantage of the Smith predictor. The delay margins corresponding to Figure 5.3 are shown in Figure 5.4. For PI and PID control the delay margin follows our M_S robustness measure quite smoothly.

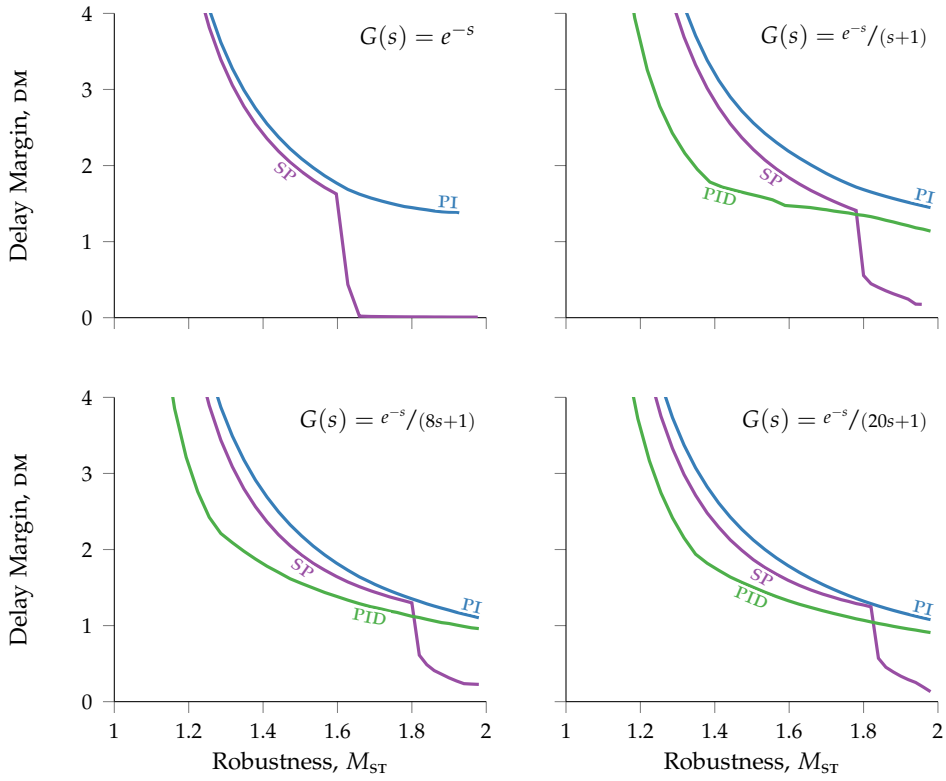

Figure 5.4: Delay Margins (DM) for the Pareto-optimal controllers in Figure 5.3.

This is not the case for the SP, which has a sudden and large drop in DM for higher M_{ST} values. For the four process models in this chapter, the drop in DM occurs at M_{ST} equal to 1.63, 1.80, 1.82, and 1.84, respectively.

In practice, this means that the the SP controller is unusable for higher M_{ST} values. Thus, the SP controller is actually worse than what is shown in Figure 5.3.

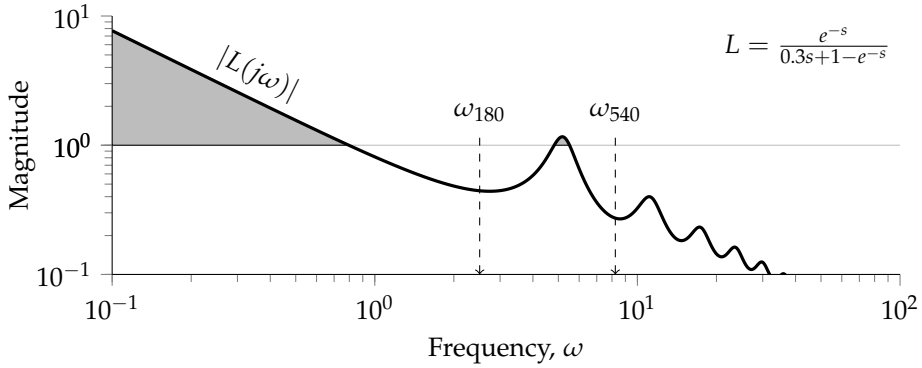

Figure 5.5: Bode plot of the SP loop function $L = GK_{\text{SP}}$ for $G(s) = e^{-s}/(s+1)$ and $M_{\text{ST}} = 1.81$.

5.6 Discussion

Robustness

Palmor (1980) showed that a SP with very high gains will have good GM and PM, but have arbitrary small DM. This agrees with Figure 5.4. Thus, when designing a Smith predictor using only the classical robustness margins (GM and PM) you easily end up with an aggressive controller with very small DM. Adam et al. (2000) showed that these robustness toward error in time delay can be arbitrary small and unsymmetrical, and this is further discussed below.

Figure 5.5 shows for the FOPD process

$$G(s) = e^{-s}/(s+1) \tag{5.15}$$

the loop function L for a SP synthesized with the IMC method ($\tau_i = \tau$) with $M_s = 1.81$. The resulting L has three crossover frequencies, and two regions with magnitude larger than one. The closed-loop

becomes unstable if the Bode stability criteria is violated,

$$\left|L(\omega)\right| < 1 \quad \text{for } \omega_{180}, \omega_{540}, \omega_{900}, \ldots, \tag{5.16}$$

where ω_{180}, ω_{540}, and ω_{900} are the frequencies where the phase is -180°, -540°, and -900° respectively. Note that the first "sudden drops" in Figure 5.4 occurs when we get $|L| \geq 1$ at ω between ω_{180} and ω_{540}, which makes ω_c jump to a higher frequency. Further jumps occur as as ω_c jumps to even higher frequencies (Gudin and Mirkin, 2007).

The two phase crossover frequencies ω_{180} and ω_{540} are marked in Figure 5.5. From the Bode criterion, the system becomes unstable if these frequencies move into the shaded regions where the magnitude is larger than one, and these frequencies shift with time delay error, $\Delta\theta$.

Since we can have $|L| > 1$ also for $\omega > \omega_{180}$ with SP, a SP with high gain can become unstable both for positive and *negative* time delay errors. This is not the case for PID which becomes unstable only for positive time delay errors.

For example, for the system in Figure 5.5 which has a nominal delay of 1, the system with SP becomes unstable for time delays errors $\Delta\theta$ in the intervals,

$$[-0.66, -0.45]\,, \;\; [0.48, 0.85]\,, \text{ and } \; [1.38, \infty)\,. \tag{5.17}$$

Closed-loop responses for a SP and a PI controller tuned for $M_{ST} = 1.81$ for three values of $\Delta\theta$ are shown in Figure 5.6. We see that the Smith predictor may start to oscillate both for negative ($\Delta\theta = -0.45$) and positive time delay errors ($\Delta\theta = 0.40$).

One solution to deal with the problem of multiple instability regions, is to limit the loop gain such that it is strictly smaller than 1

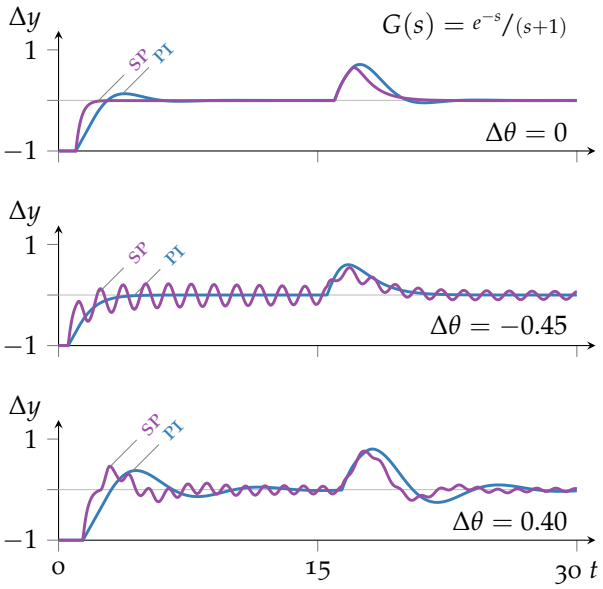

Figure 5.6: Step response for PI and SP control with different time delay errors, $\Delta\theta$. That is, the controllers are tuned with $M_{ST} = 1.81$ for a model with $\theta_m = 1$, and tested on the real process with $\theta = \theta_m + \Delta\theta$.

at higher frequencies (Ingimundarson and Hägglund, 2002), that is

$$\left|L(\omega)\right| < 1 \quad \text{for } \omega > \omega_c. \tag{5.18}$$

where ω_c is the first crossover frequency. This will ensure that the system only becomes unstable if the delay error is $\Delta\theta$ is positive, as is the case with PID. However, for system with high loop gain where one of the peaks of $L(\omega)$ is almost one, a small underestimation of the process gain can bring back the peaks. Thus for SP it is important to consider both errors in gain and time delays. A more conservative approach would be to ensure that the condition

in (5.18) holds for gains up to some factor of GM $\cdot k$.

To really ensure robustness against combined gain-delay errors we should look at the robust stability (RS) criteria for multiplicative error (Morari and Zafiriou, 1989)

$$\left|T(j\omega)\right| \leq \frac{1}{\left|l_I(s)\right|}, \tag{5.19}$$

with the following exact bound on the multiplicative error (Lundström, 1994),

$$l_I(\omega) = \begin{cases} \sqrt{r_k^2 + 2(1+r_k)(1-\cos(\theta_{\max}\omega))}, & \text{for } \omega < \pi/\theta_{\max} \\ 2 + r_k, & \text{for } \omega \geq \pi/\theta_{\max} \end{cases} \tag{5.20}$$

where r_k is the relative gain error, and $\theta_{\max}$ is the maximum delay error, which is in this chapter DM $= \theta_{\max}$. This bound can be approximated by the following first order transfer function,

$$w_I(s) = \frac{(1+\frac{r_k}{2})\theta_{\max}s + r_k}{\frac{\theta_{\max}}{2}s + 1} \tag{5.21}$$

However, this bound is conservative it will in most cases be sufficient to satisfy for only time delay error. That is with $r_k = 0$,

$$\left|T(s)\right| < \left|\frac{\frac{\theta_{\max}}{2}s + 1}{\theta_{\max}s}\right| \tag{5.22}$$

which can be approximated with the even less conservative bound,

$$\left|T(j\omega)\right| < 1/2, \quad \text{for } \omega > \pi/\theta_{\max}. \tag{5.23}$$

Thus limiting the magnitude of T is important for delay margin. Concerned about the conservatism of multiplicative bound for time

delay error Larsson and Hagglund (2009) proposed the following robustness bounds,

$$M_{S_\Delta} = \max_{\Delta\theta\in[\Delta\theta_{\min},\,\Delta\theta_{\max}]} \left\|S(s,\,\Delta\theta)\right\|_\infty \leq M_{S_\Delta}^{ub} \tag{5.24}$$

$$M_{T_\Delta} = \max_{\Delta\theta\in[\Delta\theta_{\min},\,\Delta\theta_{\max}]} \left\|T(s,\,\Delta\theta)\right\|_\infty \leq M_{T_\Delta}^{ub} \tag{5.25}$$

which ensures a minimum GM and PM even for maximum delay error.

From this we can conclude that by ensuring sufficiently small M_S and M_T peaks for SP, we get good robustness against delay error. But this also means that SP will not be able to achieve the lowest IAE values in Figure 5.3.

Predictive PI

Another way of avoiding some of these limitations with SP is to use a modified SP. The Predictive PI (PPI) controller (Hägglund, 1996) is a special case of the SP, where the parameters of the internal model, k and τ, is parametrized in terms of the controller parameters k_c and τ_i. This is done by assuming a FOPD and using the lambda tuning method to express the model parameters in term of the controller parameters. As a result, the PPI controller only needs 3 parameters to be specified, namely k_c, τ_i, and θ, which is same as the number of parameters needed for PID controller. The delay-free part of the model, $G_\circ$, is reduced and the PPI controller can be expressed as,

$$K_{\text{PPI}} = \frac{k_c\left(1+\frac{1}{\tau_i s}\right)}{1+\frac{1}{\tau_i s}(1-e^{-k_\theta s})}, \tag{5.26}$$

where k_p, k_i, and k_θ are tuning parameters. The internal controller delay can be treated as a fixed parameter $k_\theta = \theta$ (normally done), or

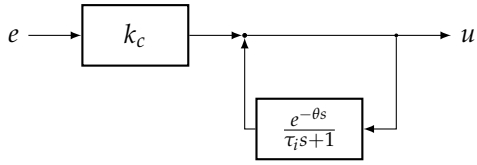

Figure 5.7: Block digram of the PPI controller

as a free parameter to additionally improve performance. Because of the modification, the PPI controller can achieve better disturbance rejection than the SP because it can avoid the zero-pole cancellation between the controller and process. Also, the PPI controller, unlike the SP, works for integrating processes because it retains integral action for these processes. However, otherwise performance compared to PID control is not improved (Larsson and Hägglund, 2012)

Tuning of PID controllers

We find in this chapter than a PID controller generally outperforms a Smith predictor controller. This is in agreement with findings of Ingimundarson and Hägglund (2002) who say that "the PID performs best over a large portion of the area but it has been shown that it might be difficult to obtain this optimal performance with manual tuning". We disagree with the last portion of this statement as we have found that the simple "improved SIMC-tunings" give IAE performance close to the optimal in Figure 5.3 (Grimholt and Skogestad, 2013). For the process in (5.1) using a cascade PID controller

$$K_{\text{PID}}(s) = \frac{k_c(\tau_i s + 1)(\tau_d s + 1)}{\tau_i s}, \tag{5.27}$$

the improved SIMC-tunings are (Grimholt and Skogestad, 2013)

$$k_c = \frac{1}{k}\frac{\tau}{\tau_c + \theta},\ \tau_i = \min\{\tau,\ 4(\tau_c + \theta)\},\ \tau_d = \theta/3 \tag{5.28}$$

These can be translated to the ideal parameters in (5.2) by computing $f = 1 + \tau_d/\tau_i$ and using,

$$\tilde{k}_c = k_c f,\ \tilde{\tau}_i = \tau_i f,\ \tilde{\tau}_d = \tau_d / f \tag{5.29}$$

By varying τ_c we can adjust the robustness. For example, selecting $\tau_c = \theta/2$ gives M_s about 1.6.

For PI control we use the same rules, but set $\tau_d = 0$ which gives the original SIMC rules. In this case, selecting $\tau_c = \theta$ gives M_s about 1.6. Note that the SIMC PI controller for a pure time delay process is an integrating controller which gives poorer performance (but smoother response) than the PI/PID controller in Figure 5.3.

5.7 Conclusion

For processes with large time delays, the common belief is that PI and PID controllers have sluggish performance, and that a Smith Predictor or similar can give much improved performance.

We find in this chapter that this is a myth. We study a wide range of first-order plus delay processes, and we find that with a fixed robustness in terms of the sensitivity function (M_s-value), the potential performance improvement is small or nonexistent (Figure 5.3). In fact, a PID-controller is in almost all cases significantly better than a SP plus PI-controller. We think this is a fair comparison, because the derivative action in the PID controllers adds a similar complexity as the Smith predictor part.

The only exception is for a pure time delay process where there is no benefit of derivative action so the optimal PID is a PI controller. In this case, there is a very small performance benefit by using SP (Figure 5.3, upper left), but this comes at the expense of a high sensitivity to time delay errors when we want tight performance (Fig-

ure 5.4). Thus, in practice the expected benefits of the Smith predictor shown in Figure 5.3 cannot be achieved because of time delay error. Actually, we find that the Smith predictors tuned for tight performance can have arbitrary small margins to time delay errors (Figure 5.4).

There are modifications of the Smith predictor, which can avoid some of the problems of the SP for integrating processes, but still the performance is not better than a PID controller with the same robustness. It has been claimed that Smith predictor is easier to tune than a PID controller, but we claim that this is not true. In summary, we think that we can safely recommend to "forget the Smith predictor", and instead use a well-tuned PI or PID controller.

Chapter Six

Conclusions

In this thesis we found the optimal trade-off between performance and robustness when using PI and PID control, and verified that the SIMC-rules gives a close-to optimal trade-off. The more specific findings are detailed below.

In Chapter 2, we successfully applied the exact gradients for a typical performance (IAE) with constrained robustness M_{ST} optimization problem. By taking advantage of the fixed structure of the problem, the exact gradients were presented in such a way that they can easily be implemented and extended to other fixed-order controllers. Evaluating the gradient is just a simple process of combining and evaluating already defined transfer functions. This enables the user to quickly find the gradients for a linear system for any fixed order controller.

In Chapter 3, the Pareto-optimal PI- and PID-settings for a FOPTD process were found. However, in practice, we recommend using the original SIMC-rules for PI-tuning as they are close-to the optimal settings. The only exception is for delay-dominant FOPTD processes, where the SIMC proportional gain is too small, but this can be cor-

rected for by using the *i*SIMC-PI rule in (3.17).

The original SIMC-rules does give PID settings for FOPTD processes. For close-to optimal PID-control for such processes, we propose the *i*SIMC rule where we added a derivative action with $\tau_d = \theta/3$ (3.16). Note that this is for the cascade PID-controller in (3.2).

The improved performance/robustness trade-off of the *i*SIMC-PI and *i*SIMC rules, comes at the expense of increased input usage in response to measurement noise, output disturbances and setpoint changes. Thus, for most industrial cases where output performance is not the main concern, the original SIMC rule may be the best choice.

In Chapter 4, we found Pareto-optimal PID controller settings for a double integrating the process, and compared with SIMC settings. The SIMC controller has almost identical performance with the optimal, in particular for more robust designs. This means that the simple SIMC PID tuning rules given in (4.3) are essentially the optimal.

In Chapter 5, we explored the benefit of using a Smith predictor. For processes with large time delays, the common belief is that PI and PID controllers have sluggish performance, and that a Smith predictor or similar can give much improved performance. However, we found that the potential performance improvement is small or nonexistent. In fact, a PID-controller is in almost all cases significantly better than a SP plus PI-controller. We think this is a fair comparison, because the derivative action in the PID controllers adds a similar complexity as the Smith predictor part.

The only exception is for a pure time delay process where there is no benefit with derivative action and the optimal PID is a PI controller. In this case, there is a small performance benefit of using SP, but this comes at the expense of a high sensitivity to time delay

errors when we want tight performance. Thus, in practice the expected benefits of the Smith predictor cannot be achieved because of time delay error. Actually, we find that the Smith predictor tuned for tight performance can have arbitrary small margins to time delay errors.

6.1 Further work

In this thesis we have covered FOPD processes including the limiting cases of a pure time delay process (time constant equal to zero) and an integrating process (time constant equal to infinity). We have also covered the double integrating process, which is a special case of SOPD process. This work could be extended by applying it to a wider class of processes. Though we started to explore the optimal trade-off and validating SIMC for more general SOPD processes though student projects (Foss, 2012; Holene, 2013). However, this work has not been completed. Also, it would be interesting to extend this work to find optimal trade-off to common unstable, oscillating and inverse-response processes.

We have shown the difference in performance-robustness benefit between PI, serial PID (can only have real zeros) and parallel PID (can also have complex zeros). A natural extension to this work would find the optimal trade-off for other classes of controllers. For example, to compare with higher order linear controllers like PIDF, and beyond. How will performance improve by allowing complex zeros or poles? When will we reach a point when there is no longer any benefit of increasing the order? However, it could be even more interesting to compare the old-and-trusted PID controller with a more modern approach like fractional order PID control (Podlubny, 1999).

We have also shown that there is little benefit of using a Smith predictor compared with well tuned PID controller. Two closely related controller is the Predictive Functional Control (PFC) proposed by Richalet et al. (1978), see also the more recent in the book Richalet and O'Donovan (2009) and Internal Model Control (IMC) proposed by Morari (see Morari and Zafiriou (1989)). It would be interesting to compare PID with PFC and IMC using the method presented in this thesis. In particular, it would be interesting to focus on the time delay robustness, because predictive methods are usually sensitive to modeling error in time delay. This may also apply more generally to other tightly tuned model-based, including Model Predictive Control (MPC), controllers.

Appendix A

Calculation alorithm and derivation of exact gradients

A.1 Pseudo code for the calculation of gradients

Algorithm 1: CostFunction(p)

$t \leftarrow$ uniform distributed time points
$e_{dy}(t) \leftarrow$ get step response of $S(s;p)$ with time steps t
$e_{du}(t) \leftarrow$ get step response of $GS(s;p)$ with time steps t
calculate IAE for e_{dy} and e_{du} using numerical integration
$J \leftarrow$ calculate cost function using (5.7)
return $(J, e_{dy}(t), e_{du}(t))$

Algorithm 2: ConstraintFunction(p)

$\omega \leftarrow$ logarithmically spaced frequency points in Ω
$c_{\text{S}} \leftarrow \left|S(j\omega)\right| - M_{\text{S}}^{ub}$
$c_{\text{T}} \leftarrow \left|T(j\omega)\right| - M_{\text{T}}^{ub}$
$c \leftarrow$ stack c_{S} and c_{T} into one vector
return (c)

Algorithm 3: GradCostFunction($p, e_{dy}(t), e_{du}(t)$)

$t \leftarrow$ uniform distributed time points
for $i \leftarrow 1$ **to** number of parameters
do
- $\nabla e_{dy}(t) \leftarrow$ get time response of $\nabla e_{dy}(s)$ from (2.25) with time steps t
- $\nabla e_{du}(t) \leftarrow$ get time response of $\nabla e_{du}(s)$ from (2.26) with time steps t
- $\nabla \text{IAE}_{dy} \leftarrow$ numerical integration of (2.21) using e_{dy} and $\nabla e_{dy}(t)$
- $\nabla \text{IAE}_{du} \leftarrow$ numerical integration of (2.22) using e_{du} and $\nabla e_{du}(t)$
- $\nabla J \leftarrow$ calculate from (2.20)

$\nabla J \leftarrow$ stack the cost function sensitivities into one vector
return (∇J)

Algorithm 4: GradConstraintFunction(p)

$\omega \leftarrow$ logarithmically spaced frequency points
for $i \leftarrow 1$ **to** number of parameters
do
- $\nabla c_{\text{S}} \leftarrow$ evaluate (2.29) for frequencies ω
- $\nabla c_{\text{T}} \leftarrow$ evaluate (2.30) for frequencies ω

$\nabla c \leftarrow$ combine ∇c_{S} and $\nabla c_{\text{T}} p_i$ into a matrix
return (∇c)

A.2 Derivation of the exact sensitivities of the cost function

Sensitivity of the absolute value

Lemma 1. *Let $g(t;p)$, abbreviated as $g(t)$, be a function that depends on time t and parameters p. The partial derivative of the absolute value of $g(t)$ with respects to the parameter p is then*

$$\frac{\partial}{\partial p}\left|g(t)\right| = sign\left\{g(t)\right\}\frac{\partial}{\partial p}\left(g(t)\right).$$

Proof. The absolute value can be written as the multiplication of the function $g(t)$ and its sign,

$$\left|g(t)\right| = \text{sign}\left\{g(t)\right\}\, g(t),$$

where the sign function has the following values

$$\text{sign}\left\{g(t)\right\} = \begin{cases} -1 & \text{if } g(t) < 0, \\ 1 & \text{if } g(t) > 0. \end{cases}$$

Using the product rule we get,

$$\frac{\partial}{\partial p}\left(\text{sign}\left\{g(t)\right\}\, g(t)\right) = \text{sign}\left\{g(t)\right\}\,\frac{\partial g(t)}{\partial p} + g(t)\,\frac{\partial \text{sign}\left\{g(t)\right\}}{\partial p}. \tag{A.1}$$

The sign function is piecewise constant and differentiable with the derivative equal o for all values except $g(t) = 0$, where the derivative is not defined. Hence the following conclusion is true,

$$g(t)\,\frac{\partial \text{sign}\left\{g(t)\right\}}{\partial p} = 0,$$

and the differential of $|g(t)|$ is as stated above. □

Sensitivity of the integrated absolute value

Theorem 1. *Let $g(t;p)$, abbreviated as $g(t)$, be a function that depends on the time t and the parameter p. The differential of the integrated absolute value of $g(t)$ on the interval from t_α to t_β with respects to the parameter p can be written as*

$$\frac{\mathrm{d}}{\mathrm{d}p}\left(\int_{t_\alpha}^{t_\beta}\left|g(t)\right|\mathrm{d}t\right)=\int_{t_\alpha}^{t_\beta}sign\left\{g(t)\right\}\left(\frac{\partial g(t)}{\partial p}\right)\mathrm{d}t \tag{A.2}$$

Proof. If $g(t)$ and its partial derivative $\frac{\partial g(t)}{\partial p}$ are continuous wrt. t and p on the intervals $\left[t_\alpha, t_\beta\right]$ and $\left[p_\alpha, p_\beta\right]$, and the integration limits are constant, then using Leibniz's rule, we can write the the integral

$$\frac{\mathrm{d}}{\mathrm{d}p}\left(\int_{t_\alpha}^{t_\beta}\left|g(t)\right|\mathrm{d}t\right)=\int_{t_\alpha}^{t_\beta}\frac{\partial\left|g(t)\right|}{\partial p}\,\mathrm{d}t.$$

Then using Lemma 1, we obtain the stated expression. □

Obtaining the sensitivities from Laplace

Theorem 2. *Let $g(t;p)$ be a linear function that depends on time t and the parameter p, and $G(s;p)$ its corresponding Laplace transform (abbreviated $g(t)$ and $G(s)$). Then the partial derivative of $g(t)$ can be expressed in terms of the inverse Laplace transform of $G(s)$,*

$$\frac{\partial g(t)}{\partial p}=\mathcal{L}^{-1}\left\{\frac{\partial G(s)}{\partial p}\right\}.$$

Proof. Differentiating the definition of the Laplace transform with respect to the parameter p we get,

$$\frac{\partial G(s)}{\partial p}=\frac{\partial}{\partial p}\int_0^\infty e^{-st}g(t)\,\mathrm{d}t.$$

Assuming that $g(t)$ and $\frac{\partial g(t)}{\partial p}$ is continuous on the integration interval, Leibniz's rule gives

$$\frac{\partial G(s)}{\partial p} = \int_0^\infty e^{-st} \frac{\partial g(t)}{\partial p} \, \mathrm{d}t,$$

which is equivalent to

$$\frac{\partial G(s)}{\partial p} = \mathcal{L}\left\{ \frac{\partial g(t)}{\partial p} \right\}.$$

Taking the inverse Laplace on both sides gives the stated expression. □

Sensitivity of $S(s)$ and $GS(s)$

We have $S = (1 + L)^{-1}$ where $L = GK$. Using the chain-rule on the definition of S, and dropping the argument s for simplicity,

$$\frac{\partial S}{\partial p_i} = \frac{\partial S}{\partial L} \frac{\partial L}{\partial K} \frac{\partial K}{\partial p_i}. \tag{A.3}$$

We have

$$\frac{\partial S}{\partial L} = \frac{\partial}{\partial L}(1 + L)^{-1} = -(1 + L)^{-2} = -S^2. \tag{A.4}$$

and $\frac{\partial L}{\partial K} = G$ Thus,

$$\frac{\partial S}{\partial p_i} = -S^2 G \frac{\partial K}{\partial p_i}. \tag{A.5}$$

Similarly,

$$\frac{\partial GS}{\partial p_i} = G \frac{\partial S}{\partial p_i} = -(GS)^2 \frac{\partial K}{\partial p_i}. \tag{A.6}$$

A.3 Derivation of the exact sensitivities for the constraints

Sensitivity of $G(j\omega)G^*(j\omega;p)$ for a specific frequency point

Lemma 2. *Let $G(j\omega;p)$ be a general transfer function, and $G^*(j\omega;p)$ its complex conjugate (abbreviated $G(j\omega)$ and $G^*(j\omega)$). Then the differential of the product of the two with respect to the parameter p is*

$$\frac{\partial}{\partial p}\left(G(j\omega)\,G^*(j\omega)\right) = 2\Re\left(G^*(j\omega)\tfrac{\partial G(j\omega)}{\partial p}\right),$$

where $\Re\{\,\cdot\,\}$ is the real part of the argument.

Proof. Write the transfer function out as their the complex numbers

$$G(j\omega) = x + jy, \tag{A.7}$$
$$G^*(j\omega) = x - jy. \tag{A.8}$$

The product rule gives

$$\Big(x-jy\Big)\frac{\partial\Big(x+jy\Big)}{\partial p} + \Big(x+jy\Big)\frac{\partial\Big(x-jy\Big)}{\partial p},$$

and becomes

$$2\left(x\frac{\partial x}{\partial p} + y\frac{\partial y}{\partial p}\right) = 2\Re\left(G^*(j\omega)\frac{\partial G(j\omega)}{\partial p}\right).$$

□

Sensitivity of $\big|G(j\omega);\,p\big|$ for a specific frequency point

Theorem 3. *Let $G(j\omega;p)$, abbreviated as $G(j\omega)$ be a general transfer function evaluated at the frequency $j\omega$. The partial derivative of the magnitude*

of $G(j\omega)$ with respects to the parameter p is then

$$\frac{\partial |G(j\omega)|}{\partial p} = \frac{1}{|G(j\omega)|} \Re\left\{ G^*(j\omega) \frac{\partial G(j\omega)}{\partial p} \right\}.$$

Proof. The derivative can be expressed in term of the squared magnitude,

$$\frac{\partial |G(j\omega)|}{\partial p} = \frac{\partial}{\partial p} \sqrt{|G(j\omega)|^2} = \frac{1}{2|G(j\omega)|} \frac{\partial |G(j\omega)|^2}{\partial p}.$$

The squared magnitude can be written as the transfer function multiplied by its complex conjugate $G^*(j\omega)$

$$|G(j\omega)|^2 = G(j\omega)\, G^*(j\omega).$$

Using the product rule presented in lemma 2, we get

$$\frac{\partial |G(j\omega)|^2}{\partial p} = 2\Re\left\{ G^*(j\omega) \tfrac{\partial G(j\omega)}{\partial p} \right\}.$$

□

Some readers might have wondered why the sensitivity of the transfer function is not written using the chain-rule with

$$\frac{\partial \left|G(j\omega)\right|^2}{\partial G(j\omega)} = \frac{\partial (G(j\omega)\, G(j\omega)^*)}{\partial G(j\omega)} = G^*(j\omega) + G(j\omega) \frac{\partial G^*(j\omega)}{\partial G(j\omega)}.$$

This is because the derivative $\frac{\partial G^*(j\omega)}{\partial G(j\omega)}$ is non-analytic and do not exist anywhere (Spiegel, 1965).

Appendix B

Simultaneous design of proportional-integral-derivative controller and measurement filter by optimization

IET Control Theory & Applications

Research Article

IET Journals
The Institution of Engineering and Technology

Simultaneous design of proportional–integral–derivative controller and measurement filter by optimisation

ISSN 1751-8644
Received on 22nd March 2016
Revised 22nd August 2016
Accepted on 29th October 2016
E-First on 2nd December 2016
doi: 10.1049/iet-cta.2016.0297
www.ietdl.org

Kristian Soltesz[1] ✉, *Chriss Grimholt*[2], *Sigurd Skogestad*[2]

[1]*Department of Automatic Control, Lund University, Lund, Sweden*
[2]*Department of Chemical Engineering, Norwegian University of Science and Technology, Trondheim, Norway*
✉ *E-mail: kristian@control.lth.se*

Abstract: A method for optimisation of proportional–integral–derivative controller parameters and measurement filter time constant is presented. The method differs from the traditional approach in that the controller and filter parameters are simultaneously optimised, as opposed to standard, sequential, design. Control performance is maximised through minimisation of the integrated absolute error caused by a unit step load disturbance. Robustness is achieved through $\mathscr{H}_\infty$ constraints on sensitivity and complementary sensitivity. At the same time, noise attenuation is enforced by limiting either the $\mathscr{H}_2$ or $\mathscr{H}_\infty$ norm of the transfer function from measurement noise to control signal. The use of exact gradients makes the synthesis method faster and more numerically robust than previously proposed alternatives.

1 Introduction

1.1 Motivation

The proportional–integral–derivative (PID) controller is by far the most widely used controller structure. Consequently, there exist an abundance of methods for PID synthesis. A majority of these aim at achieving sufficient load disturbance rejection (regulatory control) and robustness to plant model uncertainty. Most PID synthesis methods do not explicitly consider reference tracking (servo control). One reason for this might be that reference tracking can be achieved independently through a two degrees of freedom design scheme, such as [1]. Furthermore, regulatory control performance is more important than servo *ditto* in most (process) industrial applications, and controllers which achieve adequate regulatory control often also have acceptable reference tracking behaviour.

There exist several performance measures to evaluate regulatory control performance. Two well-established such measures are the integrated error (IE) and integrated absolute error (IAE), defined through (7) and (6), respectively. In their context *error* refers to that resulting from a load disturbance unit step. The IAE has an advantage over the IE in that it punishes oscillatory load responses. The two performance measures are further discussed in Section 2.2.1.

When minimising the IAE (but also the IE, or other measures of regulatory control performance) – even under robustness constraints – it is common to end up with controllers of very high gain from measurement (plant output) noise to control signal (plant input). These controllers can be practically useless, as will be demonstrated in Section 5, and shown in Fig. 6. The typical approach to alleviate this problem, is to replace the derivative term of the PID control law, by a low-pass filtered version [2]. This introduces at least one additional parameter – that of the filter. In many industrial controllers, the filter parameter is automatically set to a fixed ratio of the derivative time T_d of (4).

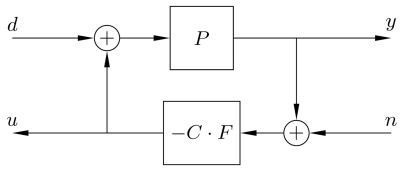

Fig. 1 *Considered control system structure*

It was argued in [3], that filtering the entire measurement signal is preferential to only filtering the derivative term. This corresponds to connecting a low-pass filter in series with the PID controller, as shown in Fig. 1. Furthermore, it was suggested to use (at least) a first-order filter for proportional–integral (PI) controllers, and a second-order filter for PID controllers, to achieve high-frequency roll-off.

Regardless of which of the above (or other) filter structures is used, the industrially established synthesis procedure is sequential. It comprises first choosing the gains of the PID controller, and subsequently tuning the filter to achieve acceptable noise rejection. Alternatively, the filter is initially fixed, whereupon the controller is synthesised. This approach works well when the resulting filter bandwidth lies significantly above that of the controller. However, if the noise spectrum is such that the bandwidths overlap, the filter will not only affect noise attenuation, but also robustness and control performance. This problem occurs regularly and suggests that the controller and filter should be simultaneously designed.

1.2 Previous work

The problem of simultaneous design of controller and filter has been studied in a number of publications. A survey of relevant ones known to the authors is given below, to produce a reference frame for the work to be presented.

In [4], an approximate formula is used to compute a filter time constant, from the derivative gain and a noise sensitivity constraint. One step further towards truly simultaneous design is taken in [5], where an iteration between controller synthesis (by optimisation) and filter design, is used. First, a controller which minimises IE, subject to an $\mathscr{H}_\infty$ constraint on S, is obtained. Subsequently, a filter is designed for the series connection of the plant and the obtained controller. The filter is then assumed to be part of the plant model, and the procedure is iterated, until it converges. Similar, iteration-based, methods are presented in [6, 7]. For industrially relevant problems it seems (but has not been proven) that these iterative methods converge.

All methods mentioned so far, involve either solving a sequence of optimisation problems [6] (which can be very time consuming), or using approximate formulae [7] (which does not guarantee optimality or constraints). The contributions [8, 9] propose a truly

simultaneous design. An optimisation problem with objective min $\| PS/s \|_\infty$, and $\mathscr{H}_\infty$ constraints on S and Q is considered. Optimisation is carried out over the parameters of a parallel form PID controller in series with a first-order low-pass filter. While the constraints are industrially well-established, the $\mathscr{H}_\infty$-objective is not. (However, it can be thought of as an approximation of IAE, as shown in [10].) In [11], Matlab-code is presented to perform the suggested optimisation. The code is very compact and simple to read, but suffers numerical issues, as pointed out in Section 3.3.

In [12], a parallel form PID controller, given by any chosen design method, is converted into a PID controller with derivative filter. The filter is chosen in relation to the closed-loop cut-off frequency or high-frequency gain, such that the effect on nominal performance and robustness is limited. A similar approach is presented in [13], in order to produce tuning rules for the measurement-filter time constant. These rules are based solely on the controller parameters, and different rules apply, depending on which method was used for controller synthesis (SIMC [14] or AMIGO [15]). The method provides a free tuning parameter, affecting the trade-off between robustness and noise sensitivity.

In [16], a particle swarm method is applied to a mixed-objective optimisation problem, aiming at maximising controller gain, while punishing the $\mathscr{H}_\infty$ norm of S and T, as well as the $\mathscr{H}_2$ norm of Q. A weighing of the different objective terms is proposed, but not motivated. In particular, this weighing may need to be changed depending on P and the spectral density of the noise signal.

Finally, Larsson and Hägglund [3, 17] propose the problem formulation adopted in this work, where IAE is minimised, under $\mathscr{H}_\infty$ constraints on S, T and an $\mathscr{H}_2$ constraint on Q. (We will also consider the $\mathscr{H}_\infty$ version of the last constraint.) This formulation is based on industrially established performance and robustness measures. However, the optimisation used in [3, 17] relies on finite difference approximations of (objective and constraint) gradients. Apart from slow execution time, the use of finite differences easily leads to poor (or no) convergence, even in the simpler case of controller synthesis with a fixed filter [18].

To summarise, most of the mentioned methods [3, 6, 8, 11, 16, 17] propose simultaneous controller synthesis and filter design by optimisation. However, they use the (simplex-like) Nelder–Mead method [19], gradient methods based on finite difference approximations [3, 8, 11, 17] or particle swarm methods [16] to find a solution. These methods all come with disadvantages: Gradient-free methods are known to be slow and care must be taken when gridding the parameter space. Using gradient methods, with finite difference approximations of the gradients, results in poor (or no) convergence within the considered context, as pointed out in [18]. Particle swarm methods require carefully chosen heuristics and provide little insight into the problem to be solved.

1.3 Novelty

This paper introduces an optimisation-based tuning method for simultaneous PID controller and measurement filter synthesis. As opposed to most previous work, the measurement filter and controller are simultaneously designed by constrained optimisation. The main novelty lies in the use of exact (analytic) gradients, to facilitate numerical robustness. Furthermore, the method eliminates the need for manual *a priori* selection of PID subtype (P, PI, PID, I, ID, PD, D). This selection is instead implicitly handled by the optimisation.

2 Problem formulation

2.1 Definitions

A synthesis scenario for the control system shown in Fig. 1 will be considered. It consists of a linear time invariant single-input–single-output (SISO) plant P, controller C, and measurement filter F. The objective is to achieve plant output $y = 0$ by means of the control signal u, in the presence of load disturbances d and measurement noise n. The standard deviations of n is denoted σ_n, and the resulting standard deviation of u is σ_u.

The SISO controller C, the filter F and their combined parameter vector θ are defined

$$C(s) = \left(k_\mathrm{p} + \frac{k_\mathrm{i}}{s} + k_\mathrm{d}s\right), \tag{1}$$

$$F(s) = \frac{1}{T_\mathrm{f}^2 s^2 + T_\mathrm{f}\sqrt{2}s + 1}, \tag{2}$$

$$\theta = \begin{bmatrix} k_\mathrm{p} & k_\mathrm{i} & k_\mathrm{d} & T_\mathrm{f} \end{bmatrix}^\mathrm{T} \in \mathbb{R}_+^4. \tag{3}$$

The parameterisation (1) is chosen in favour of the classic one

$$C(s) = K\left(1 + \frac{1}{T_i s} + T_\mathrm{d}s\right), \tag{4}$$

as the former is linear in its parameters. (It is also more general, as for instance $[k_\mathrm{p} \quad k_\mathrm{i} \quad k_\mathrm{d}] = [0 \quad 1 \quad 1]$ lacks an equivalent controller on the form (4).)

This paper focuses on using analytic expressions of gradients to optimise the controller parameters. The gradient operator, with respect to θ, will be denoted ∇. From here on, arguments of transfer functions and signals will be dropped, whenever the transform domain is clearly given by the context. The loop-transfer function is denoted $G = PCF$. Sensitivity and complementary sensitivity are defined $S = 1/(1 + G)$, and $T = 1 - S = G/(1 + G)$, respectively. Furthermore, the noise sensitivity, being the transfer function from measurement noise n to control signal u, is $Q = -CFS$.

The error caused by a unit load step disturbance d (see Fig. 1) is

$$e(t) = -\mathscr{L}^{-1}\left(S(s)P(s)\frac{1}{s}\right). \tag{5}$$

The *load step* IAE is then defined as

$$\mathrm{IAE} = \int_0^\infty |e(\tau)|\,\mathrm{d}\tau. \tag{6}$$

In addition, we will consider the mathematically more convenient IE

$$\mathrm{IE} = \int_0^\infty e(\tau)\,\mathrm{d}\tau. \tag{7}$$

2.2 Optimisation problem

The considered optimisation problem is

$$\begin{aligned} \underset{\theta}{\text{minimise}} \quad & \mathrm{IAE}, \\ \text{subject to} \quad & \| S \|_\infty \le M_s, \\ & \| T \|_\infty \le M_t, \\ & \frac{\sigma_u}{\sigma_n} \le M_q, \end{aligned} \tag{8}$$

where M_s, M_t and M_q are scalar constraint levels. It was recommended in [2] to keep M_s and M_t within the range 1.4– 1.8. The third constraint, enforcing noise attenuation by constraining the ratio between control signal and noise standard deviations, is the topic of Section 4.

2.2.1 Performance: The idea of posing PID design as a constrained optimisation problem stretches back at least three decades. In early work [5, 20] the performance objective was to minimise the IE, as defined in Section 2.1. It was shown in [5] that minimisation of the IE is equivalent to maximisation of the integral gain k_i of (1). This constitutes a convex objective in the parameter

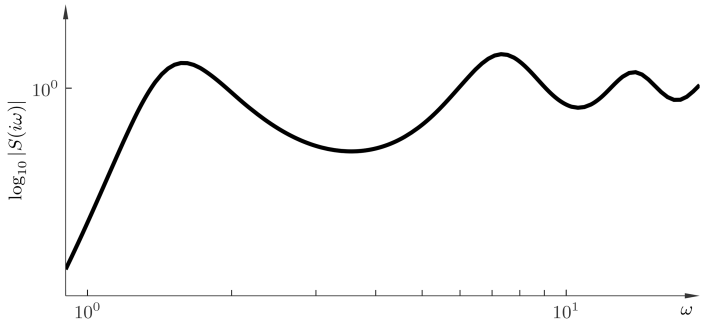

Fig. 2.

Fig. 2 *Sensitivity magnitude with multiple peaks, obtained during gradient search for IAE optimal controller. The example was obtained with* $P(s) = e^{-s}/(s+1)$, $C(s) = 1 + 4/s$ *and* $F(s) = 1$

vector θ (3), motivating the popularity of the IE as (inverse) performance measure.

If the load step response e (5) lacks zero crossings, it is evident from (7) and (6) that IE = IAE. Furthermore, well-damped control systems yield IE $\simeq$ IAE. However, oscillatory systems, with consecutive zero crossings of e, may result in IE $\ll$ IAE. It is therefore preferential to minimise the IAE, in favour of the IE, in order to avoid oscillatory behaviour. While minimisation of the IAE does not constitute a convex objective, it can be efficiently performed using gradient methods [18], as further explained in Section 3.3.

2.2.2 Robustness: It is customary, and industrially well-established, to enforce robustness of the control system through $\mathscr{H}_\infty$ constraints on S and T. These constraints limit the magnitude of S and T, and it is well-known that large values of these magnitudes make the control system sensitive to process variations [2].

Each of the two constraints imply that the Nyquist curve of the loop-transfer function G avoids one circular disc in the complex plane. Consequently, the constraints are not convex in the optimisation variable θ. However, the comparison with results of [21] in Section 5.4 indicates that minimisation of the IAE constrained by $\| S \|_\infty \leq M_s$ and $\| T \|_\infty \leq M_t$ lacks local minima for industrially relevant plant models P and constraint levels M_s, M_t.

2.2.3 Noise attenuation: The probably most common way to quantify activity of the control signal u, is through the variance σ_u^2, as in for example LQG control. In this work, a constraint on the standard deviation ratio σ_u/σ_n (or equivalently the variance ratio σ_u^2/σ_n^2) is considered. This is closely related to limiting the noise gain, defined in [7] as the ratio σ_u/σ_{y_f} between the standard deviations of the control signal and filtered measurement signal $y_f(s) = F(s)y(s)$. However, our definition has the advantage of being independent of the filter dynamics.

If an estimate $\hat{\sigma}_n$ of σ_n is available (it can be obtained in open loop at the input of F), one can constrain an upper bound $\bar{\sigma}_u$ on σ_u through $M_q = \bar{\sigma}_u/\hat{\sigma}_n$.

In Section 4.2, we will see how two different assumptions on the spectral density of n lead to $\mathscr{H}_2$ and $\mathscr{H}_\infty$ constraints on Q, respectively.

3 Optimisation

3.1 Gradient methods

It was pointed out in [18] that the efficiency of solving (8) (without considering the constraint on Q) can be significantly improved by moving from gradient-free methods to gradient-based ones. Such methods rely on (approximations) of the objective and constraint gradients, with respect to the optimisation variable. It is well-known that the convergence rate of gradient methods is significantly improved if exact gradients are supplied, as opposed to finite difference approximations. Furthermore, exact gradients improve accuracy for cases where the cost is flat in a vicinity of the optimum.

Implementation of an active-set-based method for constrained optimisation of PID parameters was presented in [11], see Section 1.2. While not relying on exact gradients, the method performs adequately on several test cases, but encounters difficulties with some dynamics, including $P(s) = e^{-s}/(s+1)$, and $P(s) = 1/(s+1)^4$.

Next, we will introduce two measures to improve convergence of gradient methods: the discretisation of $\mathscr{H}_\infty$ constraints in Section 3.2 and the use of exact gradients in Section 3.3.

3.2 Constraint discretisation

The $\mathscr{H}_\infty$ constraint on S often poses a problem for gradient-based methods, as pointed out in [18]. The reason is that close to the optimal solution, it is common for the sensitivity function to have several sensitivity peaks of equal magnitude. This results in the optimiser discretely jumping between these peaks in subsequent iterations. The situation is illustrated in Fig. 2. One solution to this problem, as proposed in [18], is to discretise the $\mathscr{H}_\infty$ constraint on S over a frequency grid $\Omega = \{\omega_1 < \omega_2 < \cdots < \omega_m\}$, resulting in m (number of grid points) constraints

$$|S(i\omega_k)| \leq M_s, \quad k = 1, \ldots, m. \tag{9}$$

Instead of requiring $\| S \|_\infty \leq M_s$, it is required that $|S(i\omega)| \leq M_s, \forall \omega \in \Omega$. This modification is adopted for *all* $\mathscr{H}_\infty$ constraints in this paper.

For the examples in this paper, a grid based on P is *a priorily* chosen, consisting of $m = 500$ logarithmically spaced grid points between the -5° and -355° phase angles of P. This choice was motivated by extensive verification, see Section 5.4.

An alternative would be to produce the grid iteratively, using for instance the cutting set method suggested in [22]. It could also be mentioned that the KYP lemma provides an alternative formulation, altogether avoiding frequency gridding [23]. However, this formulation brakes down for (continuous) time processes with time delay, which is why it has not been applied in this paper.

3.3 Exact gradients

3.3.1 Objective gradient: The exact gradient of the objective, with respect to the optimisation variable can be expressed

$$\nabla \text{IAE} = \int_0^\infty \text{sign}(e(t)) \nabla e(t) \, \text{d}t. \tag{10}$$

It was shown in [18] that the integral (10) can be evaluated by utilising the fact that $\nabla e(t) = \mathscr{L}^{-1}(\nabla e(s))$. From the definition (5) of e, it is clear that $\nabla e(t) = \mathscr{L}^{-1}(P(s)\nabla S(s)/s)$, and we have

$$\nabla S = -S^2 \nabla G, \tag{11}$$

$$\nabla G = \nabla PCF = P(\nabla C)F + PC\nabla F, \tag{12}$$

$$\nabla C = [1 \quad 1 \quad 1 \quad 0], \tag{13}$$

$$\nabla F = \begin{bmatrix} 0 & 0 & 0 & \partial F/\partial T_{\mathrm{f}} \end{bmatrix}, \tag{14}$$

$$\partial F/\partial T_{\mathrm{f}} = -F^2 T_{\mathrm{f}}(2s + T_{\mathrm{f}}/\sqrt{2}) \tag{15}$$

This allows us to evaluate $\nabla e(t)$ through a step response simulation of ∇S, defined through (11)–(15), which in term enables the evaluation of (10).

If, instead, the IE is used, the minimisation objective becomes $-k_{\mathrm{i}}$ (see Section 2.2.1), with corresponding gradient $\nabla(-k_{\mathrm{i}}) = [0 \quad -1 \quad 0 \quad 0]$.

3.3.2 Robustness constraint gradients: The gradients of the discretised robustness constraints $\nabla|S|$ and $\nabla|T|$, see Section 3.2, were presented in [18]. In this paper, we will utilise the fact that the constraints represent circular discs, which must be avoided by the open-loop transfer function G. Expressions for the centra c_c and radii r_c of these constraint discs, as functions of the constraint levels M_s and M_t, are found in [2]. This allows for an equivalent reformulation of the robustness constraints on the form

$$|G - c_c| - r_c \leq 0, \tag{16}$$

where $|S| \leq M_s$ and $|T| \leq M_t$ corresponds to

$$c_s = -1, \quad r_s = \frac{1}{M_s}, \quad c_t = -\frac{M_t^2}{M_t^2 - 1}, \quad r_t = \frac{M_t}{M_t^2 - 1}. \tag{17}$$

The advantage of this reformulation is that it yields less complicated expressions for both the constraints (16) and their gradients

$$\nabla|G - c_c| - r_c = \frac{1}{|G - c_c|}\mathrm{Re}(G^* \nabla G), \tag{18}$$

where * denotes conjugation: $G^*(s) = -G(-s)$, and ∇G can be evaluated using (12)–(15).

4 Noise attenuation

4.1 Filter structure

A second-order measurement filter, (2), is chosen to guarantee high frequency roll-off. The filter has a fixed damping ratio as suggested in [3]. (Note that the damping ratio ζ is that of the filter poles, whereas ζ is used in [11] to denote the damping ratio of the filtered controller zeros.) It is possible to treat $\zeta > 0$ as a free optimisation parameter, but experience has shown that this will only marginally increase performance, while significantly increasing computation time.

4.2 Variance bounds

This section is devoted to the last constraint of (8), namely that on the noise amplification ratio σ_u/σ_n. The purpose of the constraint is to limit variance of the control signal u, resulting from the noise n, not to exceed a user-specified tolerable level σ_u^2. The variance σ_n^2 is assumed to be known (it can easily be estimated in open loop). For cases where σ_n^2 is not known, the constraint level M_q of (8) can be considered a free tuning parameter, constituting a trade-off between performance (IAE) and noise attenuation (σ_u/σ_n).

Two cases will be studied: one for (band-limited) white noise in Section 4.2.1, and one for noise of unknown spectral density in Section 4.2.2.

4.2.1 White noise: White noise n is characterised by a constant spectral density $\Phi_n(\omega) = \Phi_0$, resulting in infinite variance, or equivalently, infinite energy. However, white noise does not occur in nature, where the bandwidth of any signal is limited by the mechanism by which it is generated or measured. We will denote by ω_B the bandwidth of the sensor used in the control system. (For a periodically sampled digital controller, the Nyquist frequency constitutes an upper bound on ω_B.) The variance of the noise seen by the sensor is

$$\sigma_n^2 = \frac{1}{2\pi}\int_{-\omega_B}^{\omega_B} \Phi_0 \,\mathrm{d}\omega = \frac{\omega_B}{\pi}\Phi_0. \tag{19}$$

The resulting control signal variance is

$$\sigma_u^2 = \frac{1}{2\pi}\int_{-\omega_B}^{\omega_B} \Phi_0 |Q(i\omega)|^2 \,\mathrm{d}\omega = \sigma_n^2 \,\|\, Q \,\|_2^2, \tag{20}$$

where the last equality holds as a consequence of the aforementioned band-limiting sensor. If the noise is (assumed to be) white, we consequently choose the $\mathscr{H}_2$ constraint

$$\frac{1}{\pi}\int_0^{\infty} |Q(i\omega)|^2 \,\mathrm{d}\omega = \,\|\, Q \,\|_2^2 \leq M_q^2, \tag{21}$$

with corresponding gradient

$$\nabla\frac{1}{\pi}\int_0^{\infty} |Q(i\omega)|^2 \,\mathrm{d}\omega = \frac{2}{\pi}\int_0^{\infty} |Q(i\omega)|\nabla|Q(i\omega)| \,\mathrm{d}\omega, \tag{22}$$

where

$$\nabla|Q| = \frac{|S|}{|CF|}\left(|C|^2\mathrm{Re}(SF^*\nabla F) + |F|^2\mathrm{Re}(SC^*\nabla C)\right). \tag{23}$$

4.2.2 Unclassified noise: If the spectral density of n is unknown, we will instead make use of the following inequality:

$$\sigma_u^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} |Q(i\omega)|^2 \Phi_n(\omega) \,\mathrm{d}\omega \leq \,\|\, Q \,\|_\infty^2 \,\frac{1}{2\pi}\int_{-\infty}^{\infty} \Phi_n(\omega) \,\mathrm{d}\omega = \,\|\, Q \,\|_\infty^2 \,\sigma_n^2, \tag{24}$$

which can be rewritten $\sigma_u/\sigma_n \leq \,\|\, Q \,\|_\infty$. Consequently, constraining the $\mathscr{H}_\infty$ norm of Q by M_q (conservatively) guarantees $\sigma_u/\sigma_n \leq M_q$, regardless of the spectral density of n. Upon discretisation, motivated in Section 3.2, this corresponds to a set of constraints (one for each frequency $\omega \in \Omega$) on the form

$$|Q| = \frac{|CF|}{|1 + G|} \leq M_q, \tag{25}$$

$$|CF| - M_q|1 + G| \leq 0. \tag{26}$$

The corresponding gradients are given by

$$\nabla|CF| - M_q\nabla|1 + G|, \tag{27}$$

and can be evaluated using the methodology of Section 3.3.

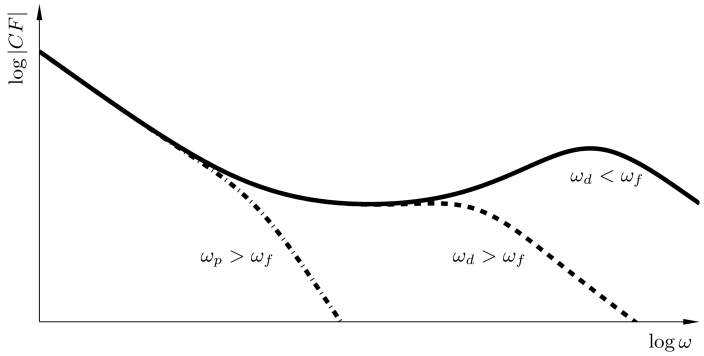

Fig. 3.

Fig. 3 *Bode magnitude of PID controller with combined measurement filter (solid) and the same controller with two slower filters resulting in the loss of derivative action (dashed) and proportional action (dashed-dotted)*

4.3 Filter time constant bounds

To avoid solutions where the filter time constant T_f is unbound from above, it is essential to achieve high-frequency roll-off within the frequency grid Ω. This is done by constraining the corner frequency $\omega_\mathrm{f} = 1/T_\mathrm{f}$ of F to lie below the highest frequency ω_m of the grid. Similarity, ω_f needs to lie above the smallest grid frequency ω_1 to avoid solutions which are unbound from below. Consequently, the following hard constraints are imposes:

$$\omega_1 \leq \omega_\mathrm{f} \leq \omega_m. \tag{28}$$

The corner frequency ω_d, introduced by derivative action, is

$$\omega_d = -\arg \min_{|s|} |C(s) = 0| = \frac{1}{2k_\mathrm{d}}|s_0|, \tag{29}$$

$$s_0 = k_\mathrm{p} + \sqrt{k_\mathrm{p}^2 - 4k_\mathrm{i}k_\mathrm{d}} \tag{30}$$

Derivative action is lost if $\omega_d > \omega_\mathrm{f}$, as illustrated in Fig. 3. This may cause ambiguity in the solution of (8). It can be avoided by enforcing $\omega_d \leq \omega_\mathrm{f}$ through the equivalent constraint

$$T_\mathrm{f}|s_0| - 2k_\mathrm{d} \leq 0, \tag{31}$$

with corresponding gradient

$$\nabla(T_\mathrm{f}|s_0| - 2k_\mathrm{d}) = \frac{T_\mathrm{f}}{|s_0|}\mathrm{Re}(s_0^*\nabla s_0) + \begin{bmatrix}0 & 0 & -2 & |s_0|\end{bmatrix}, \tag{32}$$

$$\nabla s_0 = \frac{1}{s_0 - k_\mathrm{p}}\begin{bmatrix}s_0 & -2k_\mathrm{d} & -2k_\mathrm{i} & 0\end{bmatrix}. \tag{33}$$

For controllers with $k_\mathrm{d} = 0$, integral action is lost if $\omega_p > \omega_\mathrm{f}$, where $\omega_p = k_\mathrm{i}/k_\mathrm{p}$ is the corner frequency corresponding to the zero introduced by proportional action. However, this situation never arises if (31) is imposed. For $k_\mathrm{d} = 0$, (31) then becomes $2k_\mathrm{p}T_\mathrm{f} \leq 0$, which can only be fulfilled with $k_\mathrm{p} = 0$ due to (28). Either the optimisation will produce a controller with $k_\mathrm{d} > 0$, for which case (31) is valid, or an I (purely integrating) controller, for which (28) provides sufficient bounds on T_f, since the I controller lacks zeros.

It should be noted that the above does not exclude any PID controller subtypes. For instance, filtered PI behaviour is achieved at the limit $\omega_d \to \omega_\mathrm{f}$. More importantly, the user does not have to make an *a priori* PID subtype selection (such as to include or exclude derivative action), as this is handled implicitly by the optimisation.

5 Results

5.1 Method summary

Before considering a case example in Section 5.2, the method outlined in Sections 2–4, is briefly summarised below.

i. Obtain a model P of the plant to be controlled (using any method).
ii. Impose robustness constraint levels M_s and (possibly) M_t.
iii. Decide on an upper bound on the control signal variance $\bar{\sigma}_u^2$, induced by measurement noise. Obtain a measurement or estimate $\hat{\sigma}_n^2$ of the noise variance, and set the constraint level $M_q = \bar{\sigma}_u/\hat{\sigma}_n$. If n can be assumed to be white, impose an $\mathscr{H}_2$ constraint on the noise sensitivity Q, otherwise an $\mathscr{H}_\infty$ constraint.
iv. Construct a frequency grid Ω, and discretise all $\mathscr{H}_\infty$ constraints according to Section 3.2.
v. Solve (8) using a gradient method, and provide it with exact gradients, as defined throughout this paper.

5.2 Case example

The proposed synthesis method is demonstrated through a realistic example, in which the plant to be controlled is

$$P(s) = \frac{e^{-s}}{s+1}. \tag{34}$$

This structure is a commonly used model in process industry, where higher-order dynamics are often lumped together and distributed between the time constant and delay. It should, however, be noted that the method proposed herein can be used directly on higher-order dynamical models, when such are available – see Section 5.3 for an example.

To demonstrate the effects of the noise constraint, the system is subject to white noise of standard deviation $\sigma_n = 0.1$, band-limited by the Nyquist frequency corresponding to the controller sampling period $h = 0.02$ (chosen to give 100 samples per average residence time of P). Robustness constraints are fixed at industrially relevant values of $M_s = M_t = 1.4$, and it is (correctly) assumed that the noise is white, i.e. the case of Section 4.2.1 is considered.

The optimisation problem (8) and corresponding constraint gradients were provided to a solver (invoked through the Matlab fmincon command). The solution for unconstrained noise sensitivity ($M_q = \infty$) is shown on the last row of Table 1. The large corresponding value $\| Q \|_2 \simeq 125$ results in high noise amplification from n to u, as shown in the light grey curve of Fig. 6*b*. For most industrial scenarios, this would result in unacceptable, actuator wear.

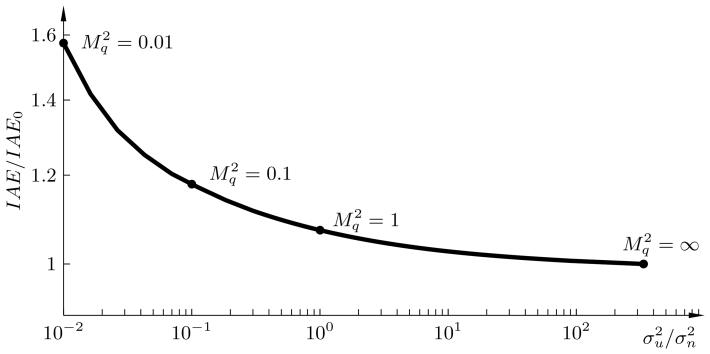

Fig. 4.

Fig. 4 *Pareto front, relating σ_u^2/σ_n^2, constrained by M_q^2, to performance decrease IAE/IAE$_0$. The objective IAE$_0$ results from the unconstrained case $M_q = \infty$. Markers correspond to the controllers evaluated in Fig. 6*

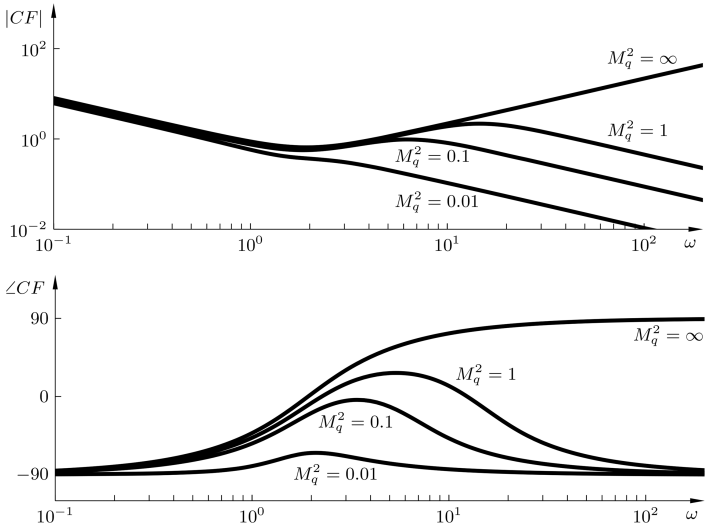

Fig. 5.

Fig. 5 *Bode plots of Pareto-optimal controller with filter, CF, from Table 1*

By solving (8) for additional values of M_q, we obtain the Pareto front shown in Fig. 4. It relates the (variance ratio) constraint level M_q^2 to the corresponding increase in optimisation objective. The controller and filter parameters corresponding to the design cases shown in Fig. 4 are enlisted in Table 1. Bode plots for *CF* of Table 1 are shown in Fig. 5. They clearly show how smaller values of M_q result in more aggressive filtering, at the cost of losing phase advance.

To illustrate the trade-off, controllers corresponding to the (variance ratio) constraint levels $M_q^2 \in \{0.01, 0.1, 1, \infty\}$, marked in Fig. 4, were evaluated in a load unit step test. The resulting plant outputs *y* and control signals *u* are shown in Fig. 6.

Table 1 *Parallel form (4) PID parameters and filter time constant T_f, together with increase in IAE compared with IAE$_0$, resulting from $M_q = \infty$*

M_q^2	K	T_i	T_d	T_f	IAE/IAE$_0$
0.01	0.461	0.748	0.530	0.478	1.57
0.1	0.566	0.796	0.374	0.155	1.18
1	0.612	0.806	0.341	0.068	1.07
∞	0.651	0.809	0.333	$1/\omega_m \simeq 0$	1.00

5.3 Case Example 2

To demonstrate the usefulness of the method on more complicated dynamics, a process with two poles, one zero and time delay is considered:

$$P(s) = \frac{s+2}{(s+1)(s+5)} e^{-3s}. \tag{35}$$

We will again consider $\sigma_n^2 = 0.1$. For this example, the noise constraint is fixed to $M_q = 1$, while different levels or robustness, constituted through $M = M_s = M_t \in \{1.2, 1.4, 1.6\}$, are evaluated. The load disturbance responses of resulting control systems are shown in Fig. 7. In this particular case, there is no major performance loss associated with an increase in robustness margin from $M = 1.6$ to $M = 1.2$.

5.4 Verification

For the very simple case of unconstrained PI control of $P(s) = e^{-sL}/s$ (delayed integrator), the problem (8) has been shown to be convex [24]. The likely lack of convexity for other instances of (8) is, however, not sufficient for the existence of local minima. Furthermore, it is possible to construct cases which do hold local minima. One example is conditionally stable plants, as described in

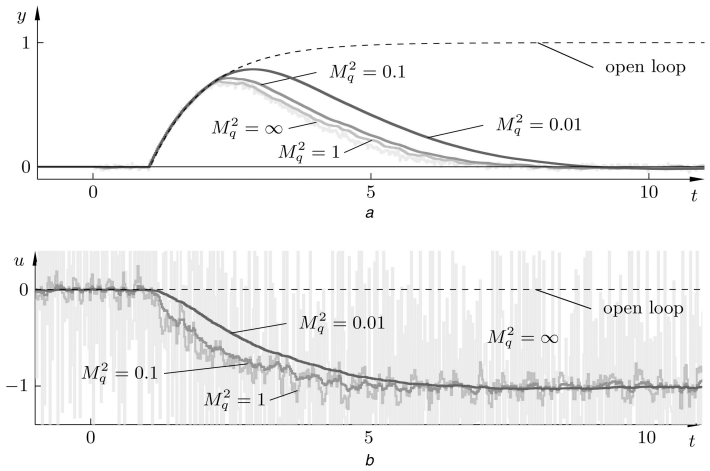

Fig. 6.

Fig. 6 *Simulated response to load step disturbances for four Pareto optimal controllers, obtained with* $M_q^2 = 0.01$ *(black),* $M_q^2 = 0.1$ *(dark grey),* $M_q^2 = 1$ *(grey) and* $M_q^2 = \infty$ *(light grey, truncated). Dashed line shows open-loop disturbance response (with* $u = 0$*)*

(a) Plant output *y*, ***(b)*** Control signal *u*

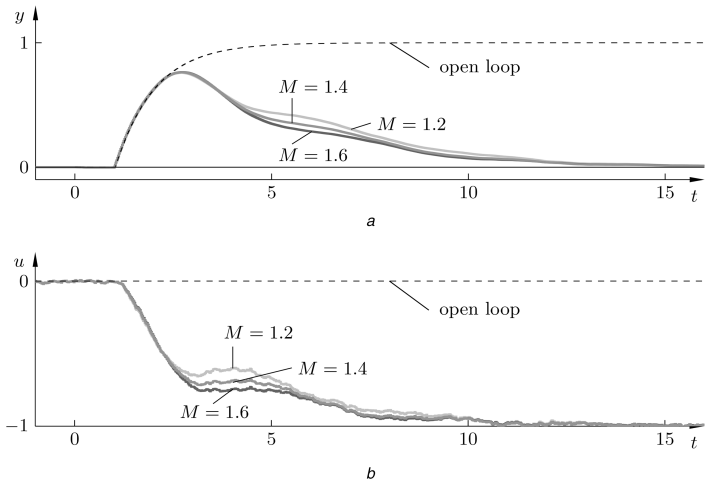

Fig. 7 *Simulated response to load step disturbances. Three controllers were synthesised for the plant (35), with* $M_q^2 = 1$ *and* $M = 1.2$ *(light grey),* $M = 1.4$ *(grey),* $M = 1.6$ *(dark grey)*

(a) Plant output *y*, ***(b)*** Control signal *u*

[25]. To the knowledge of the authors there exists no complete classification of which cases result in local minima.

To verify optimality of solutions provided by the proposed method, it was evaluated over a set industrially representative process dynamics. This set constitutes the AMIGO test batch, found in [2]. Optimal PID and filter parameter for several constraint levels for the set were recently reported in [21]. The proposed method found the optimal solution (within numerical tolerance imposed by Ω) for all test cases. Execution times ranged 2–10 s on a standard desktop computer. (The code was not optimised for speed.) In all cases, including the examples of Sections 5.2 and 5.3, the method was initialised with the IE-optimal PID controller with fixed $F = 1$, found using the convex–concave procedure presented in [26].

6 Conclusion

A method for simultaneous PID controller and measurement filter design has been proposed. The method minimises the load step IAE, under constraints enforcing robustness and noise attenuation. The corresponding constraint level can be viewed as a free design parameter, constituting a trade-off between performance and noise attenuation from the control signal. Typically, as illustrated by Fig. 6, an increase in noise attenuation performance comes at the cost of decreased load attenuation performance.

The main novelty lies in the use of explicit gradient of the objective and constraint functions with respect to the optimisation variable. This makes the proposed method more robust than if finite difference approximations would be utilised.

Selection of PID subtype (such as PI, P, or PID) is handled implicitly by the optimisation – no *a priori* selection is necessary.

Numeric robustness and optimality of the proposed method has been demonstrated through evaluation over a large set of industrially relevant design scenarios, for which the optimal solutions have been previously reported.

7 Acknowledgments

This work was partially financed by Kungliga Ingenjörsvetenskapsakademien (IVA). The first author would also

like to acknowledge the Lund University LCCC and ELLIIT research centra. The authors would like also to acknowledge Dr. Olof Garpinger for sharing his knowledge and experience on the subject.

8 References

[1] Hast, M., Hägglund, T.: 'Optimal proportional–integral–derivative set-point weighting and tuning rules for proportional set-point weights', *IET Control Theory Appl.*, 2015, **9**, (15), pp. 2266–2272

[2] Åström, K.J., Hägglund, T.: '*Advanced PID control*' (ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, 2006)

[3] Larsson, P.-O., Hägglund, T.: 'Comparison between robust PID and predictive PI controllers with constrained control signal noise sensitivity'. Conf. on Advances in PID Control. IFAC, Brescia, Italy, 2012

[4] Šekara, T.B., Mataušek, M.R.: 'Optimization of PID controller based on maximization of the proportional gain under constraints on robustness and sensitivity to measurement noise', *IEEE Trans. Autom. Control*, 2009, **54**, (1), pp. 184–189

[5] Panagopoulos, H., Åström, K.J., Hägglund, T.: 'Design of PID controllers based on constrained optimisation', *IEEE Proc. Control Theory Appl.*, 2002, **149**, (1), pp. 32–40

[6] Garpinger, O., Hägglund, T.: 'Software-based optimal PID design with robustness and noise sensitivity constraints', *J. Process Control*, 2015, **33**, pp. 90–101

[7] Romero Segovia, V., Hägglund, T.: 'Measurement noise filtering for PID controllers', *J. Process Control*, 2014, **24**, (4), pp. 299–313

[8] Kristiansson, B., Lennartson, B.: 'Robust and optimal tuning of PI and PID controllers', *IEEE Proc. Control Theory Appl.*, 2002, **149**, (1), pp. 17–25

[9] Kristiansson, B., Lennartson, B.: 'Evaluation of simple tuning of PID controllers with high-frequency robustness', *J. Process Control*, 2006, **16**, (2), pp. 91–102

[10] Balchen, J.G.: 'A performance index for feedback control systems based on the fourier transform of the control deviation', *Acta Polytechn. Scandinavica*, 1958, **247**, pp. 1–18

[11] Lennartson, B.: 'Multi criteria hinf optimal PID controllers from an undergraduate perspective'. IFAC Conf. on Advances in PID Control, Brescia, Italy, March 2012

[12] Leva, A., Maggio, M.: 'A systematic way to extend ideal PID tuning rules to the real structure', *J. Process Control*, 2011, **21**, (1), pp. 130–136

[13] Romero Segovia, V., Hägglund, T., Åström, K.J.: 'Measurement noise filtering for common PID tuning rules', *Control Eng. Pract.*, 2014, **32**, pp. 43–63

[14] Skogestad, S.: 'Simple analytic rules for model reduction and PID controller tuning', *J. Process Control*, 2003, **13**, (4), pp. 291–309

[15] Åström, K.J., Hägglund, T.: 'Revisiting the Ziegler–Nichols step response method for PID control', *J. Process Control*, 2004, **14**, pp. 635–650

[16] Micić, A.D., Mataušek, M.R.: 'Optimization of PID controller with higher-order noise filter', *J. Process Control*, 2014, **24**, (5), pp. 694–700

[17] Larsson, P.-O., Hägglund, T.: 'Control signal constraints and filter order selection for PID controllers'. Proc. of American Control Conf., San Francisco, California, USA, 2011

[18] Grimholt, C., Skogestad, S.: 'Improved optimization-based design of PID controllers using exact gradients'. European Symp. on Computer Aided Process Engineering, Copenhagen, Denmark, June 2015

[19] Garpingar, O., Hägglund, T.: 'A software tool for robust PID design'. 17th IFAC-World Congress, Seoul, Korea, July 2008

[20] Åström, K.J., Panagopoulos, H., Hägglund, T.: 'Design of PI controllers based on non-convex optimization', *Automatica*, 1998, **34**, (5), pp. 585–601

[21] Garpinger, O. 'Optimal PI and PID parameters for a batch of benchmark process models representative for the process industry'. Dept. Automatic Control, Lund University, Sweden, Technical Report 7645, 2015

[22] Lipp, T., Boyd, S.: 'Variations and extension of the convex–concave procedure', *Optim. Eng.*, 2014, **17**, (2), pp. 1–25

[23] Hara, S., Iwasaki, T., Shiokata, D.: 'Robust PID control using generalized KYP synthesis: direct open-loop shaping in multiple frequency ranges', *IEEE Control Syst.*, 2006, **26**, (1), pp. 80–91

[24] Esch, J., Knings, T., Ding, S.X.: 'Optimal performance tuning of a PI-controller for an integrator plant with uncertain parameters as a convex optimisation problem'. 52nd Conf. on Decision and Control, 2013, pp. 1977–1982

[25] Gerry, J.P.: 'Conditionally stable tuning of PI and PID controllers on common processes'. American Control Conf., 1984, pp. 512–517

[26] Hast, M., Åström, K.J., Bernhardsson, B., *et al.*: 'Pid design by convex-concave optimization'. 2013 European Control Conf., Zürich, Switzerland, July 2013

Appendix C

A comparison between Internal Model Control, optimal PIDF and robust controllers for unstable flow in risers

Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

A comparison between Internal Model Control, optimal PIDF and robust controllers for unstable flow in risers

E. Jahanshahi, V. De Oliveira, C. Grimholt, S. Skogestad [1]

Department of Chemical Engineering, Norwegian Univ. of Science and technology, Trondheim, NO-7491 (e-mail: skoge@ntnu.no).

Abstract: Anti-slug control of multiphase risers involves stabilizing an open-loop unstable operating point. PID control is the preferred choice in the industry, but appropriate tuning is required for robustness. In this paper, we define PIDF as a PID with a low-pass filter on its derivative action where the low-pass filter is crucial for the dynamics. We compared a new PIDF tuning based on Internal Model Control (IMC), together with two other tunings from the literature, with an optimal PIDF controller. The optimal PIDF tuning was found by minimizing a performance cost function while satisfying robustness requirements (input usage and complementary sensitivity peak). Next, we considered two types of robust $\mathcal{H}_\infty$ controller (mixed-sensitivity and loop-shaping). We compared the controllers based on their pareto-optimality, and we tested the controllers experimentally. We found that the new IMC-PIDF controllers is the closest to the optimal PIDF controller, but the robustness can be further improved by $\mathcal{H}_\infty$ loop-shaping.

Keywords: Anti-slug control, unstable systems, robust control, Internal Model Control

1. INTRODUCTION

The severe-slugging flow regime which is common at offshore oilfields is characterized by large oscillatory variations in pressure and flow rates. This multi-phase flow regime in pipelines and risers is undesirable and an effective solution is needed to suppress it (Godhavn et al., 2005). One way to prevent this behaviour is to reduce the opening of the top-side choke valve. However, this conventional solution reduces the production rate from the oil wells. The recommended solution to maintain a non-oscillatory flow regime together with the maximum possible production rate is active control of the topside choke valve (Havre et al., 2000). Measurements such as pressure, flow rate or fluid density are used as the controlled variables and the topside choke valve is the main manipulated variable.

Existing anti-slug control systems are not robust and tend to become unstable after some time, because of inflow disturbances or plant changes. The main objective of our research is to find a robust solution for anti-slug control systems. The nonlinearity of the system is problematic for stabilization as the gain changes drastically between different operating conditions. In addition, another difficulty for stabilization is the effective time delay .

One solution is to use nonlinear model-based controllers to counteract the nonlinearity (e.g. Di Meglio et al., 2010). However, we have found that these solutions are not robust against time delays or plant/model mismatch (Jahanshahi and Skogestad, 2013b).

An alternative approach is to use PID controllers to stabilize the unstable flow. The PI and PID controllers are still the most widely used controllers in the industry and even from the academic point of view they are unbeatable in combined robustness and performance.

The purpose of this paper is to verify different tuning rules when applied to anti-slugging control and to give recommendations about the most appropriate rules to use. For this, we compare PID controllers with optimal controllers in simulations and experiments.

Jahanshahi and Skogestad (2013a) showed that a linear model with two unstable poles and one stable zero is sufficient for designing an anti-slug controller. They identified such a model from a closed-loop step test and proposed a PIDF tuning based on Internal Model Control (IMC) for this system. This tuning rules were slightly modified by including the derivative action filter.

We here define a four-parameter PIDF controller as a PID controller with filtered derivative action (Åström and Hägglund, 2006).

$$K_{\mathrm{PIDF}}(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{T_f s + 1} \tag{1}$$

where K_p is the proportional gain, K_i is the integral action gain, K_d is the derivative action gain and T_f is the filter time constant. We differentiate this from a standard PID controller, because the low-pass filter is a crucial part of the controller for our application. That is, the filter time constant cannot be reduced without sacrificing performance.

One of the optimal controllers used for the comparison, is a PIDF where optimal tuning are found by minimizing

[1] Corresponding author

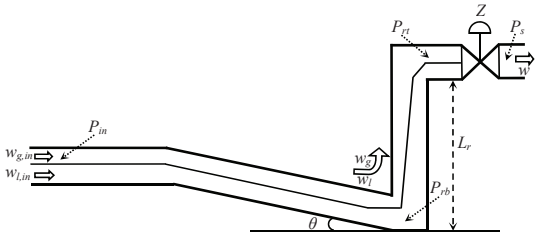

Fig. 1. Schematic presentation of system

a performance cost function while specifying robustness requirement (input usage and complementary sensitivity peak). Then, we consider use of two $\mathcal{H}_\infty$ robust controllers. $\mathcal{H}_\infty$ mixed-sensitivity design minimizes $\overline{\sigma}(S)$ for performance, $\overline{\sigma}(T)$ for robustness and low sensitivity to noise, and $\overline{\sigma}(KS)$ to penalize large inputs. In $\mathcal{H}_\infty$ loop-shaping design, we specify an initial plant loop shape, then the loop-shaping procedure increases robustness by maximizing the stability margin (Skogestad and Postlethwaite, 2005). The PIDF controller found by (Jahanshahi and Skogestad, 2013a) is used to make the initially shaped plant for the loop-shaping design.

For sake of completeness, we have also included in our study the simple PID tuning rules for unstable processes proposed by Rao and Chidambaram (2006) and Lee et al. (2006).

This paper is organized as follows. Section 2 describes the pipeline-riser system. The new PIDF tuning is presented in Section 3, and the optimal PIDF tuning is introduced in Section 4. Mixed-sensitivity and loop-shaping designs are presented in Section 5 and Section 6, respectively. The results are presented in Section 7. Finally, we summarize the main conclusions and remarks in Section 8.

2. SYSTEMS DESCRIPTION

Fig. 1 shows a schematic presentation of the system. The inflow rates of gas and liquid to the system, $w_{g,in}$ and $w_{l,in}$, are assumed to be independent disturbances and the top-side choke valve opening ($0 < Z < 100\%$) is the manipulated variable. A fourth-order dynamic model for this system was presented by Jahanshahi and Skogestad (2011). The state variables of this model are as:

- m_{gp}: mass of gas in pipeline [kg]
- m_{lp}: mass of liquid in pipeline [kg]
- m_{gr}: mass of gas in riser [kg]
- m_{lr}: mass of liquid in riser [kg]

The four state equations of the model are

$$\dot{m}_{gp} = w_{g,in} - w_g \tag{2}$$
$$\dot{m}_{lp} = w_{l,in} - w_l \tag{3}$$
$$\dot{m}_{gr} = w_g - \alpha w \tag{4}$$
$$\dot{m}_{lr} = w_l - (1-\alpha) w \tag{5}$$

The flow rates of gas and liquid from the pipeline to the riser, w_g and w_l, are determined by pressure drop across the riser-base where they are described by virtual valve equations. The outlet mixture flow rate, w, is determined by the opening percentage of the top-side choke valve, Z. The different flow rates and the gas mass fraction, α, in the

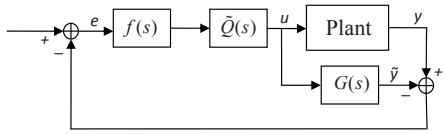

Fig. 2. Block diagram of Internal Model Control system

equations (2)-(5) are given by additional model equations given by Jahanshahi and Skogestad (2011).

However, Jahanshahi and Skogestad (2013a) showed that a second-order model with two unstable poles and one stable zero is enough for the control design purposes, and such a model can be identified by a closed-loop step test.

3. PIDF TUNING BASED ON IMC DESIGN

3.1 IMC design for unstable systems

The Internal Model Control (IMC) design procedure is summarized by Morari and Zafiriou (1989). The block diagram of the IMC structure is shown in Fig. 2. Here, $G(s)$ is the nominal model which in general has some mismatch with the real plant $G_p(s)$. $\tilde{Q}(s)$ is the inverse of the minimum phase part of $G(s)$ and $f(s)$ is a low-pass filter for robustness of the closed-loop system.

The IMC configuration in Fig. 2 cannot be used directly for unstable systems; instead we use the conventional feedback structure with the stabilizing controller

$$C(s) = \frac{\tilde{Q}(s) f(s)}{1 - G(s)\tilde{Q}(s) f(s)}. \tag{6}$$

For internal stability, $\tilde{Q}f$ and $(1 - G\tilde{Q}f)$ have to be stable. We use the identified model with two unstable poles and one stable zero (Jahanshahi and Skogestad, 2013a) as the plant model:

$$G(s) = \frac{\hat{b}_1 s + \hat{b}_0}{s^2 - \hat{a}_1 s + \hat{a}_0} = \frac{k'(s+\varphi)}{(s-\pi_1)(s-\pi_2)} \tag{7}$$

and we get

$$\tilde{Q}(s) = \frac{(1/k')(s-\pi_1)(s-\pi_2)}{s+\varphi} \tag{8}$$

We design the filter $f(s)$ as explained by Morari and Zafiriou (1989), which gives the following third order filter

$$f(s) = \frac{\alpha_2 s^2 + \alpha_1 s + \alpha_0}{(\lambda s + 1)^3}, \tag{9}$$

where λ is an adjustable closed-loop time-constant. We choose $\alpha_0 = 1$ to get integral action and the coefficients α_1 and α_2 are calculated by solving the following system of linear equations:

$$\begin{pmatrix} {\pi_1}^2 & \pi_1 & 1 \\ {\pi_2}^2 & \pi_2 & 1 \end{pmatrix} \begin{pmatrix} \alpha_2 \\ \alpha_1 \\ \alpha_0 \end{pmatrix} = \begin{pmatrix} (\lambda\pi_1 + 1)^3 \\ (\lambda\pi_2 + 1)^3 \end{pmatrix} \tag{10}$$

Finally, from (6) the feedback version of the IMC controller becomes (Jahanshahi and Skogestad, 2013a)

$$C(s) = \frac{[\frac{1}{k'\lambda^3}](\alpha_2 s^2 + \alpha_1 s + 1)}{s(s+\varphi)}. \tag{11}$$

3.2 PIDF implementation of IMC controller

The IMC controller in (11) is a second order transfer function which can be written in form of a PID controller with a low-pass filter.

$$K_{\text{PIDF}}(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{T_f s + 1} \tag{12}$$

where

$$T_f = 1/\varphi \tag{13}$$

$$K_i = \frac{T_f}{k'\lambda^3} \tag{14}$$

$$K_p = K_i\alpha_1 - K_i T_f \tag{15}$$

$$K_d = K_i\alpha_2 - K_p T_f \tag{16}$$

For the controller work in practice, we require that $K_p < 0$ and $K_d < 0$; and we must choose λ such that these two conditions are satisfied.

4. OPTIMAL PIDF CONTROL

For comparison purpose, we will define an optimal PIDF controller. However, optimality is generally difficult to define as we need to balance various factors such as output performance, robustness, input usage and noise sensitivity. We follow Grimholt and Skogestad (2012) and define the output performance as a weighted sum of the integrated square error (ISE) for disturbance at the plant input and output. However, a controller with good performance (low J) may not be robust. Thus, Grimholt and Skogestad (2012) proposed to optimize J for a given robustness (M_s value). This gives a set of pareto-optimal controllers. However, we found that for our application it was necessary to add a third dimension to constraint the input usage (M_{ks}). This results in a pareto optimal surface.

4.1 Evaluation of performance, robustness and input usage

Performance: Output performance is related to the difference between the measurement $y(t)$ and the setpoint y_s, and can be quantified in several different ways. In this paper we chose to quantify the performance in terms of a single scalar, namely the integrated squared error:

$$\text{ISE} = \int_0^\infty (y(t) - y_s(t))^2 \, dt \tag{17}$$

To balance the servo/regulatory trade-off we choose a weighted average of ISE for a step input load disturbance d_i and ISE for a step output load disturbance d_o:

$$J(K) = 0.5\left[\frac{\text{ISE}_{do}(K)}{\text{ISE}^\circ_{do}} + \frac{\text{ISE}_{di}(K)}{\text{ISE}^\circ_{di}}\right] \tag{18}$$

where K is a PIDF-controller. The weighting factors ISE°_{di} and ISE°_{do} are for reference PIDF-controllers, which for the given process is ISE-optimal for a step load change on input and output, respectively. More details about this formulation can be found in Grimholt and Skogestad (2012).

Robustness: Robustness can be quantified in several ways. Most commonly used is the sensitivity peak (M_s), complementarity sensitivity peak (M_t), gain margin (GM), phase margin (PM), and allowable time delay error ($\frac{\Delta\theta}{\theta}$). In this paper we have chosen to quantify robustness as

$$M = \max(M_s, M_t) \tag{19}$$

where $M_s = \|S\|_\infty = \max_\omega |S|$ and $M_t = \|T\|_\infty = \max_\omega |T|$ for all frequencies and

$$S = \frac{1}{1+GK}, \qquad T = \frac{GK}{1+GK} \tag{20}$$

$\|\cdot\|_\infty$ is the $\mathcal{H}_\infty$-norm, which gives the peak value in the frequency domain. A small M tells that large relative perturbations in the process transfer functions are permitted (Åström and Hägglund, 2006). Since our system is unstable, we will normally have $M = M_t$. For stable processes, however, we would generally have $M = M_s$.

Input usage: A major concern in our application is to limit the input usage. This can be achieved by limiting the magnitude peak $M_{ks} = \|KS\|_\infty = \max_\omega |KS|$, where

$$KS = \frac{K}{1+GK} \tag{21}$$

4.2 Optimization problem:

The pareto optimal PIDF controller (K) was found by solving the following optimization problem

$$\min_K \quad J(K) = 0.5\left[\frac{\text{ISE}_{do}(K)}{\text{ISE}^\circ_{do}} + \frac{\text{ISE}_{di}(K)}{\text{ISE}^\circ_{di}}\right]$$
$$\text{s.t.} \quad M = m; \quad M_{ks} = m_{ks} \tag{22}$$

for various combinations of m (the desired M value) and m_{ks} (the specified bound in the magnitude of the input signal).

Computing the optimal controller: We propose solving the above optimization problem using gradient based nonlinear programming (NLP) techniques due to their fast convergence properties. However, the reliability of such methods depends on the quality of the gradients used by the NLP solvers. For this purpose, we use forward sensitivity calculation to obtain the exact gradients ($\nabla_K J$) of the objective function with respect to the parameters of the controller. The forward sensitivity method principle resides on first calculating $E = \frac{dx}{dt}$, where x are the closed-loop states of the system, and then relating this to J through chain-rule. Following the derivation by Biegler (2010), E can be obtained by solving the system

$$\frac{dE}{dt} = \frac{\partial f}{\partial x}E(t) + \frac{\partial f^T}{\partial K}, \quad B(0) = 0 \tag{23}$$

where $f \equiv \frac{dx}{dt} = A(K)x + B(K)u$ represents the state-space model of the closed-loop system. The gradient is then computed by

$$\nabla_K J = \frac{\partial J}{\partial x}E(t_f) + \frac{\partial J}{\partial K} \tag{24}$$

Note that the required partial derivatives may be computed using automatic differentiation or symbolic differentiation tools. The analytical calculation of the constraint gradients is more involved and should be further investigated. Here, the constraint gradients are approximated by central differences. It is worth to point out that, due to the nonconvexity of the optimization problem, we are bound to converge to a local minimum. One possibility to overcome this problem is to initialize the NLP solver

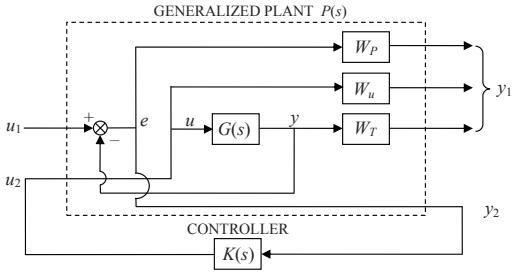

Fig. 3. Closed-loop system for mixed sensitivity control design

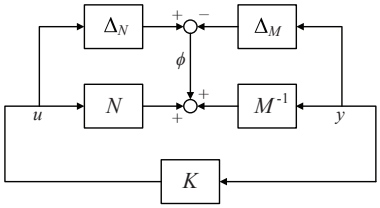

Fig. 4. $\mathcal{H}_\infty$ robust stabilization problem

with several different initial guesses and then choose the best overall solution. Alternatively, one may use a global optimization approach.

5. $\mathcal{H}_\infty$ MIXED-SENSITIVITY DESIGN

We consider an $\mathcal{H}_\infty$ problem where we want to bound $\overline{\sigma}(S)$ for performance, $\overline{\sigma}(T)$ for robustness and low sensitivity to noise, and $\overline{\sigma}(KS)$ to penalize large inputs. These requirements may be combined into a stacked $\mathcal{H}_\infty$ problem (Skogestad and Postlethwaite, 2005).

$$\min_K \|N(K)\|_\infty, \quad N \triangleq \begin{bmatrix} W_u KS \\ W_T T \\ W_P S \end{bmatrix} \tag{25}$$

where W_u, W_T and W_P determine the desired shapes of KS, T and S, respectively. Typically, W_P^{-1} is chosen to be small at low frequencies to achieve good disturbance attenuation (i.e., performance), and W_T^{-1} is chosen to be small outside the control bandwidth, which helps to ensure good stability margin (i.e., robustness). W_u is often chosen as a constant. The solution to this optimization problem gives a stabilizing controller K that satisfies (Doyle et al., 1989; Glover and Doyle, 1988):

$$\begin{aligned} \overline{\sigma}(KS(j\omega)) &\le \gamma \underline{\sigma}(W_u^{-1}(j\omega)) \\ \overline{\sigma}(T(j\omega)) &\le \gamma \underline{\sigma}(W_T^{-1}(j\omega)) \\ \overline{\sigma}(S(j\omega)) &\le \gamma \underline{\sigma}(W_P^{-1}(j\omega)) \end{aligned} \tag{26}$$

y_2 is the particular output for feedback control in the generalized plant in Fig. 3. The value of γ in equation (26) should be as small as possible for good controllability. However, it depends on the design specifications W_u, W_T and W_P.

6. $\mathcal{H}_\infty$ LOOP-SHAPING DESIGN

We consider the stabilization of the plant G which has a normalized left coprime factorization

$$G = M^{-1}N \tag{27}$$

A perturbed plant model G_p can then be written as

$$G_p = (M + \Delta_M)^{-1}(N + \Delta_N) \tag{28}$$

where Δ_M and Δ_N are stable unknown transfer functions which represent the uncertainty in the nominal plant model G. The objective of robust stabilization is to stabilize not only the nominal model G, but a family of perturbed plants defined by

$$G_p = \left\{(M + \Delta_M)^{-1}(N + \Delta_N) : \|[\Delta_N \;\; \Delta_M]\|_\infty < \epsilon\right\} \tag{29}$$

where $\epsilon > 0$ is then the stability margin (Skogestad and Postlethwaite, 2005). To maximize this stability margin is the problem of robust stabilization of normalized coprime factor plant description as introduced and solved by Glover and McFarlane (1989).

For the perturbed feedback system of Fig. 4, the stability property is robust if and only if the nominal feedback system is stable and

$$\gamma_K \triangleq \left\| \begin{bmatrix} K \\ I \end{bmatrix} (I - GK)^{-1} M^{-1} \right\|_\infty \le \frac{1}{\epsilon} \tag{30}$$

Notice that γ_K is the $\mathcal{H}_\infty$ norm from ϕ (see Fig. 4) to $\begin{bmatrix} u \\ y \end{bmatrix}$ and $(I - GK)^{-1}$ is the sensitivity function for this positive feedback arrangement. A small γ_K is corresponding to a large stability margin.

7. RESULTS

All the results (simulation and experimental) in this paper are based on the following model.

$$G(s) = \frac{-0.0098(s + 0.25)}{s^2 - 0.04s + 0.025} \tag{31}$$

This model was identified by Jahanshahi and Skogestad (2013a) from an experimental closed-loop step test around an operating point with the valve opening of $Z = 30\%$.

7.1 Pareto-Optimality Comparison

The optimization problem (22) was solved for a range of desired M_{ks} and M_t values using the linear model (31) (Here we assumed $M = M_t$ for all cases since we have an unstable system). This results of the optimizations form a Pareto front surface, which can be seen in Fig. 5. For simplicity, we did not include T_f as a degree of freedom in the optimization; instead, we fixed $T_f = 4$. This choice makes the filter counteract the effect of the zero of the plant, which is close to optimal this case. The NLP was solved using SNOPT Gill et al. (2005). Some points have been validated using brute force extensive search.

Figure 5 clearly depicts the trade-off between robustness, performance and input usage. The red line in Fig. 5 is the result from the IMC PIDF for different values of closed-loop time constant λ. By decreasing λ we get a faster controller with larger input usage M_{ks}, but M_t remains approximately constant. Note that the performance of the IMC PIDF is close to the pareto optimal surface for a large range of M_{ks}.

Figure 6 shows a cross-section of the PIDF Pareto surface with $M_{ks} = 50$, where the other controllers are also shown. All the controllers are tuned to give $M_{ks} = 50$.

Compared to Chidambaram-PIDF and Lee-PIDF, IMC-PIDF gives a better trade-off between robustness and performance. $\mathcal{H}_\infty$ loop-shaping controller gives a better combined performance and robustness. However, it is a higher order controller. Surprisingly, $\mathcal{H}_\infty$ mixed sensitivity gave a inferior performance compared to PIDF. Perhaps, a better performance could be achieved by a finer tuning of the weighting transfer functions W_P, W_T and W_u.

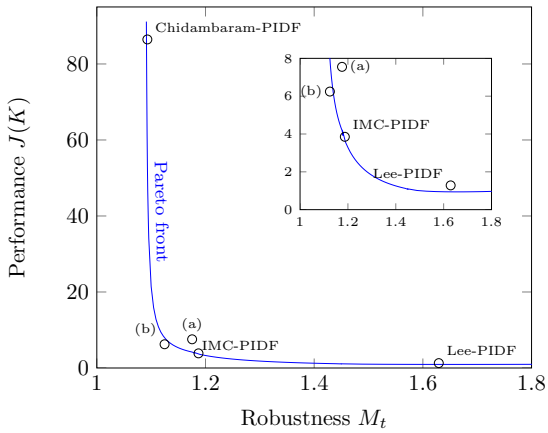

Fig. 6. Pareto front for $M_{ks} = 50$ for PIDF with $T_f = 4$. Point (a) represents the $\mathcal{H}_\infty$ mixed-sensitivity controller; point (b) represents the $\mathcal{H}_\infty$ loop-shaping controller.

7.2 Experimental setup

The experiments were performed on a laboratory setup for anti-slug control at the Chemical Engineering Department of NTNU. Fig. 7 shows a schematic presentation of the laboratory setup. The pipeline and the riser are made from flexible pipes with 2 cm inner diameter. The length of the pipeline is 4 m, and it is inclined with a 15° angle. The height of the riser is 3 m. A buffer tank is used to simulate the effect of a long pipe with the same volume, such that the total resulting length of pipe would be about 70 m.

The topside choke valve is used as the input for control. The separator pressure after the topside choke valve is nominally constant at atmospheric pressure. The feed into the pipeline is assumed to be at constant flow rates, 4 l/min of water and 4.5 l/min of air. With these boundary conditions, the critical valve opening where the system switches from stable (non-slug) to oscillatory (slug) flow is at $Z^* = 15\%$ for the top-side valve. The bifurcation diagrams are shown in Fig. 8.

The desired steady-state (dashed middle line) in slugging conditions ($Z > 15\%$) is unstable, but it can be stabilized by using control. The slope of the steady-state line (in the middle) is the static gain of the system, $k = \partial y/\partial u = \partial P_{in}/\partial Z$. As the valve opening increase this slope decreases, and the gain finally approaches to zero. This makes control of the system with large valve openings very difficult.

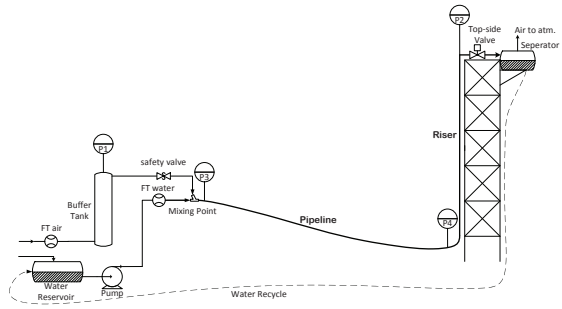

Fig. 7. Experimental setup

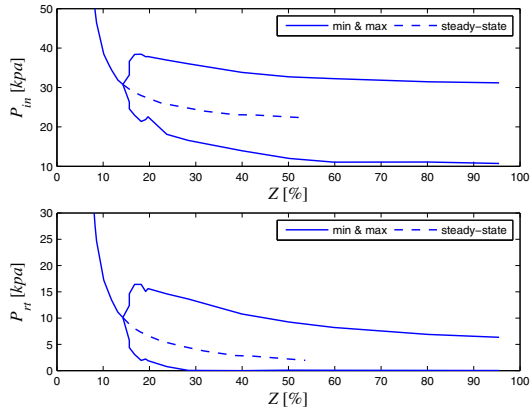

Fig. 8. Bifurcation diagrams for experimental setup

7.3 Experimental results

The controlled output in experiments is the inlet pressure of the pipeline (P_{in}) and we use the same set of descending pressure set-points in all experiments. As mentioned in above controlling the system with large valve openings (low pressure set-points) is difficult. We decrease the controller set-point to see if the controller can stabilize the system with lower set-point. The controllers are tuned (designed) for a valve opening of $Z = 30\%$, and controllers with good gain margin can stabilize the system with larger valve openings (lower set-points). To have an impartial comparison for robustness of the controllers, we tune the controllers with the same values of input usage ($M_{ks} = 50$). One interesting relationship for the KS peak of the PIDF controller in (12) can be written as follows.

$$M_{ks} = -(K_d/T_f + K_p) \tag{32}$$

Optimal PIDF: Fig. 9 and Fig. 10 show experimental result of two optimal PIDF controllers, optimal PIDF (1) and optimal PIDF (2). The controller tunings are given in Table. 1. The optimal PIDF (2) was optimized for a smaller values of M_t which resulted in a better gain margin and less oscillations is observed in Fig. 10 (better robustness). However, the optimal PIDF (2) yields higher values for ISE (Table. 1).

IMC PIDF : We used the identified model in (31) for an IMC design. We chose the filter time constant $\lambda = 6.666\ s$

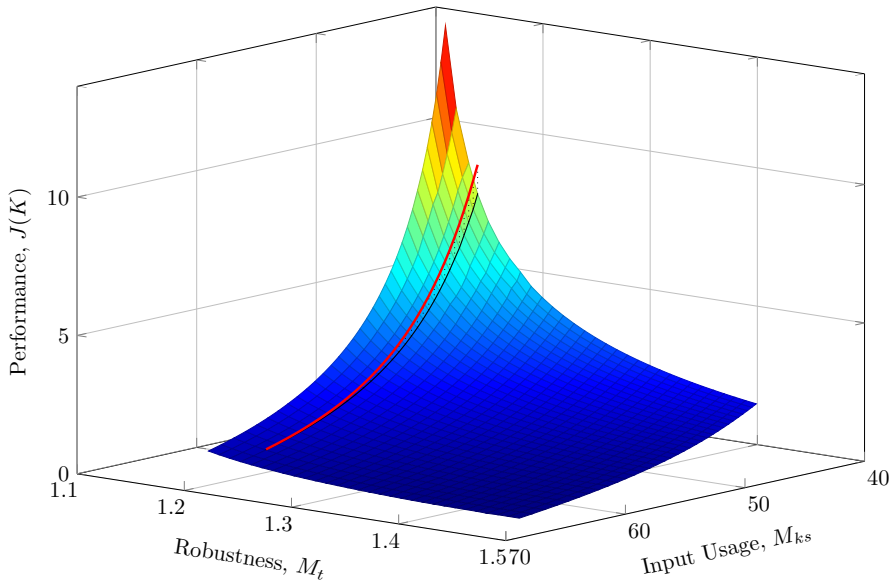

Fig. 5. Pareto optimal PIDF surface and IMC PIDF controller (red line).

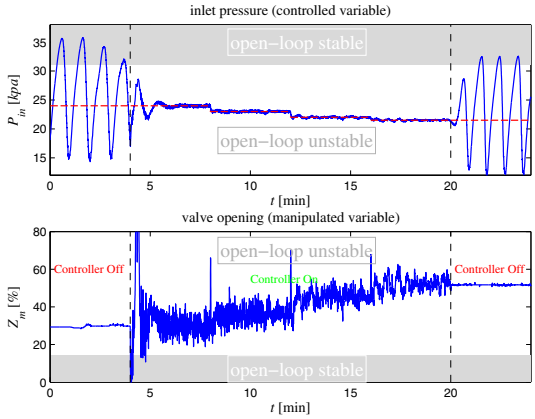

Fig. 9. Experimental result of optimal PIDF (1) with $K_p = -3.089$, $K_i = -1.62$, $K_d = -186.73$, $T_f = 4$

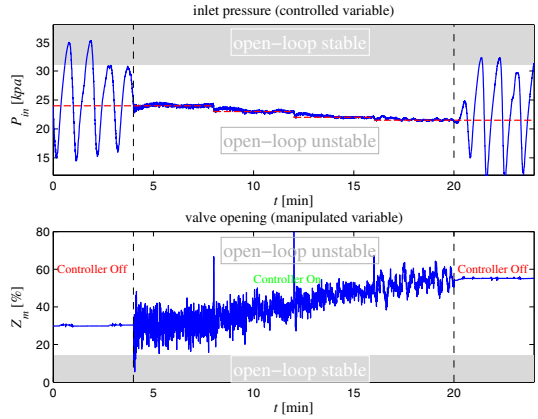

Fig. 10. Experimental result of optimal PIDF (2) with $K_p = 0.15$, $K_i = -0.198$, $K_d = -198.10$, $T_f = 4$

to get $M_{ks} = 50$. The resulting IMC controller becomes

$$C(s) = \frac{-50(s^2 + 0.0867s + 0.0069)}{s(s + 0.25)}. \qquad (33)$$

Note that the integral time for this controller is $\tau_I = {}^{K_p}/_{K_i} = 8.58$ s and the derivative time is $\tau_D = {}^{K_d}/_{K_p} = 12.89$ s. Since we have $\tau_I < 4\tau_D$, the zeros are complex and the controller cannot be implemented in cascade (series) form. The PIDF tuning resulted from this controller is given in Table. 1, and Fig. 11 shows performance of the IMC-PIDF controller in the experiment.

Chidambaram PIDF : The Chidambaram tuning (Rao and Chidambaram, 2006) is for systems with one zero, two unstable zeros and time delay. However, we do not have time delay our system, and we expect the tuning rules with $\theta = 0$ are still valid. The problem with this controller is that it does not have a low-pass filter on the derivative action; this results in a large KS peak and the controller becomes very aggressive. To solve this problem, we added the same low-pass filter as the one used in the IMC-PIDF controller. With this modification the Chidambaram tuning gives good results; the experimental result of this controller is shown in Fig. 12. We used $\tau_c = 20.17$ s to get $M_{ks} = 50$; the resulting tuning is given in Table. 1.

Lee PIDF : The Lee tuning (Lee et al., 2006) is based on analytic IMC-PIDF for first-order unstable systems with time delay. We had to approximate the model in (31) to a first-order model. We neglected the constant terms in the numerator and the denominator which are small values. This is same as what the model reduction

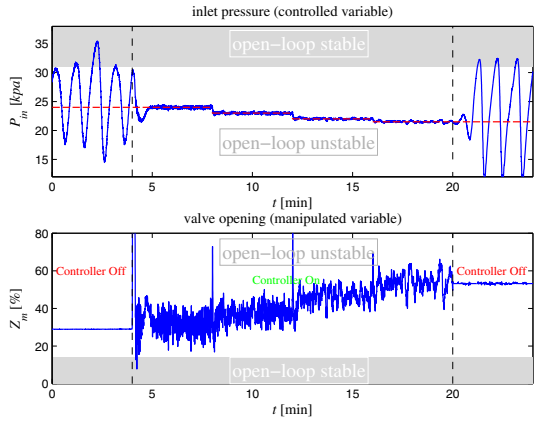

Fig. 11. Experimental result of IMC PIDF with $K_p = -11.84$, $K_i = -1.38$, $K_d = -152.65$, $T_f = 4$

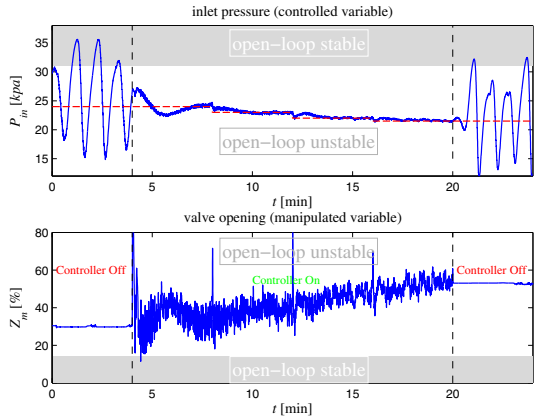

Fig. 12. Experimental result of Chidam. PIDF with $K_p = 1.69$, $K_i = -0.15$, $K_d = -206.91$, $T_f = 4$

toolbox of Matlab (*modred* routine with 'Truncate' option) does which preserves the high-frequency information. The reduced-order model is given in (34), and we used $\lambda = 5.35$ s to get $M_{ks} = 50$; the experimental result is shown in Fig. 13.

$$G_{red} = \frac{-0.245}{25s - 1} \tag{34}$$

$\mathcal{H}_\infty$ loop-shaping: We used the IMC-PIDF controller to obtain the initially shaped plant for the $\mathcal{H}_\infty$ loop-shaping design. The following fifth-order controller was resulted.

$$C(s) = \frac{-188.49(s^2 + 0.02s + 0.005)(s^2 + 0.087s + 0.0069)}{s(s + 0.25)(s + 3.76)(s^2 + 0.082s + 0.0067)} \tag{35}$$

The experimental result of the controller in (35) is shown in Fig. 14.

$\mathcal{H}_\infty$ mixed-sensitivity: We design the $\mathcal{H}_\infty$ mixed-sensitivity controller with the following design specifications:

$$W_P(s) = \frac{s/M_s + \omega_B}{s + \omega_B A}, \tag{36}$$

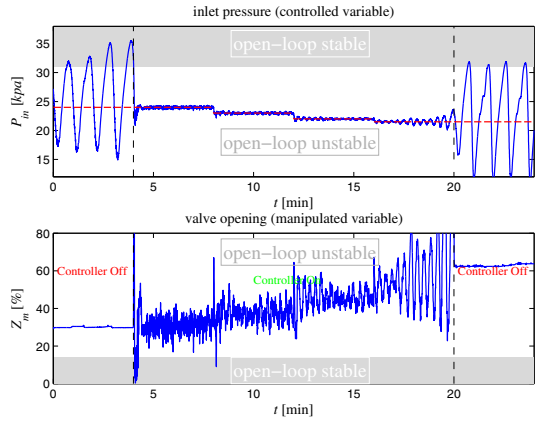

Fig. 13. Experimental result of Lee PIDF with $K_p = -41.05$, $K_i = -3.42$, $K_d = -0.082$, $T_f = 4$

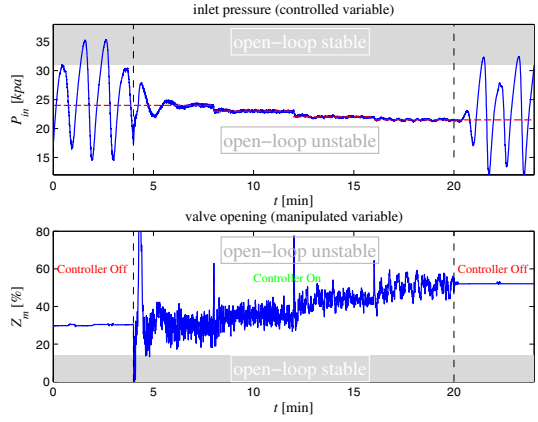

Fig. 14. Experimental result of loop-shaping $\mathcal{H}_\infty$

$$W_T(s) = \frac{s/(10\omega_B) + 1}{0.01s + 1}, \tag{37}$$

$$W_u = 0.0135, \tag{38}$$

where $M_s = 1$, $\omega_B = 0.14$ rad/s and $A = 0.01$. We chose these design specifications so that we achieve $M_{ks} = 50$ and good robustness properties. We get the following fourth-order stabilizing controller.

$$C(s) = \frac{-9.08\times10^6(s + 100)(s^2 + 0.0137s + 0.011)}{(s + 1.8 \times 10^5)(s + 112.5)(s + 0.231)(s + 0.0014)} \tag{39}$$

We achieved $\gamma = 1.21$ with this controller; the experimental performance is shown in Fig. 15.

8. CONCLUSION

In this paper we developed and compared feedback controllers for unstable multiphase flow in risers. The study included three sets of simple PIDF tuning rules, optimal PIDF and two $\mathcal{H}_\infty$ controllers. The comparison was based on Pareto optimality and experimental tests carried out in a prototype flow system. We showed that for this case the IMC-PIDF controllers are very close to the PIDF Pareto

Table 1. Comparison of different controllers in experiments

Controller	K_p	K_i	K_d	T_f	ISE	$\|S\|_\infty$	$\|T\|_\infty$	$\|KS\|_\infty$	GM	DM
Optimal PIDF (1)	-3.089	-1.62	-186.73	4	160.79	1.00	1.15	50	0.12	2.67
Optimal PIDF (2)	0.150	-0.198	-198.09	4	647.175	1.00	1.09	50	0.086	2.80
IMC PIDF	-11.84	-1.38	-152.65	4	171.45	1.00	1.19	50	0.11	2.49
Chidambaram PIDF	1.69	-0.15	-206.90	4	864.75	1.13	1.09	50	0.084	2.81
Lee PIDF	-41.05	-3.42	-0.08	4	726.88	1.20	1.62	50	0.17	1.70
$\mathcal{H}_\infty$ Loop Shaping	–	–	–	–	184.98	1.10	1.12	50	0.10	2.48
$\mathcal{H}_\infty$ Mixed Sensitivity	–	–	–	–	330.25	1.00	1.18	50	0.15	3.00

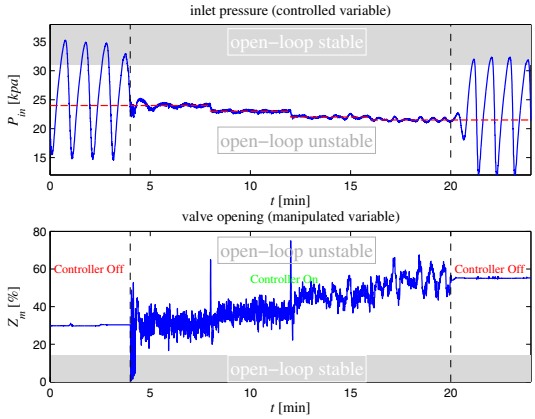

Fig. 15. Experimental result of mixed-sensitivity $\mathcal{H}_\infty$

optimal surface for a large range of the tuning parameter. Better results can be achieved by the $\mathcal{H}_\infty$ loop-shaping approach, where we employ the IMC-PIDF controller to obtain the initially shaped plant. However, this method results in higher order controllers which may not be desired by the practitioner. The $\mathcal{H}_\infty$ mixed-sensitivity design is more involved as it requires tuning of many weights simultaneously. However, we could not achieve better results than that of a PIDF controller for this case and further investigation is needed.

REFERENCES

Åström, K. and Hägglund, T. (2006). *Advanced PID Control*. ISA.

Biegler, L.T. (2010). *Nonlinear programming: Concepts, algorithms, and applications to chemical processes*. SIAM.

Di Meglio, F., Kaasa, G.O., Petit, N., and Alstad, V. (2010). Model-based control of slugging flow: an experimental case study. In *American Control Conference*, 2995–3002. Baltimore, USA.

Doyle, J., Glover, K., Khargonekar, P., and Francis, B. (1989). State-space solutions to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems. *IEEE Transactions on Automatic Control*, 34(8), 831–847.

Gill, P.E., Murray, W., and Sunders, M.A. (2005). SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. *SIAM Journal on Optimization*, 47(1), 99–131.

Glover, K. and Doyle, J.C. (1988). State-space formulae for all stabilizing controllers that satisfy an $\mathcal{H}_\infty$-norm bound and relations to relations to risk sensitivity. *Systems and Control Letters*, 11(3), 167–172.

Glover, K. and McFarlane, D. (1989). Robust stabilization of normalized coprime factor plant descriptions with h_∞-bounded uncertainty. *IEEE Transactions on Automatic Control*, 34(8), 821–830.

Godhavn, J.M., Fard, M.P., and Fuchs, P.H. (2005). New slug control strategies, tuning rules and experimental results. *Journal of Process Control*, 15, 547–557.

Grimholt, C. and Skogestad, S. (2012). Optimal PI-control and verification of the SIMC tuning rules. In *IFAC Conference on Advances in PID Control*. Brescia, Italy.

Havre, K., Stornes, K., and Stray, H. (2000). Taming slug flow in pipelines. *ABB Review*, 4, 55–63.

Jahanshahi, E. and Skogestad, S. (2011). Simplified dynamical models for control of severe slugging in multiphase risers. In *18th IFAC World Congress*, 1634–1639. Milan, Italy.

Jahanshahi, E. and Skogestad, S. (2013a). Closed-loop model identification and pid/pi tuning for robust anti-slug control. In *10th IFAC International Symposium on Dynamics and Control of Process Systems*. Mumbai, India.

Jahanshahi, E. and Skogestad, S. (2013b). Comparison between nonlinear modelbased controllers and gain-scheduling internal model control based on identified model. In *52nd IEEE Conference on Decision and Control*. Florence, Italy.

Lee, Y., Park, S., and Lee, M. (2006). Consider the generalized imc-pid method for pid controller tuning of time-delay processes. *Hybrocarbon Processing*, 6, 87–91.

Morari, M. and Zafiriou, E. (1989). *Robust Process Control*. Prentice Hall, Englewood Cliffs, New Jersey.

Rao, A.S. and Chidambaram, M. (2006). Control of unstable processes with two rhp poles, a zero and time delay. *Asia-Pacific Journal of Chemical Engineering*, 1(1-2), 63–69.

Skogestad, S. and Postlethwaite, I. (2005). *Multivariable Feedback Control: Analysis and Design*. Wiley & Sons, Chichester, West Sussex, UK.

Appendix D

Optimization of oil field production under gas coning conditions using the optimal closed-loop estimator

2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production,
May 27-29, 2015, Florianópolis, Brazil

Optimization of Oil Field Production Under Gas Coning Conditions Using the Optimal Closed-Loop Estimator

Chriss Grimholt **Sigurd Skogestad***

Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway
** e-mail: skoge@ntnu.no*

Abstract: In an oil field that has reached its maximum gas handling capacity, and where wells are producing under gas coning conditions such that the gas-oil ratio (GOR) depends on the wells production rate, oil production is maximized if the marginal GOR (mGOR) is equal for all wells. The GOR and the mGOR are not readily available measurements, but they can be predicted using detailed reservoir models. In this paper we propose predicting the mGOR for each well from measurements like pressure and valve position by using the static linear *optimal closed-loop estimator*. The estimator can be generated for each well individually, and it is not necessary to consider the production network as a whole. The *optimal closed-loop estimator* is intended for use in combination with feedback. Based on a simple two-well case study, we show that the method is effective and results in close-to-optimal production.

Keywords: Production optimization, static estimators, feedback

1. INTRODUCTION

Optimal production from an oil field, for example maximum oil rates, involves finding an optimal combination of individual well rates. Usually, wells produce oil, gas and water. The gas-oil ratio (GOR) becomes an important factor in the later stages of a field's production if the maximum gas handling capacity of the facility has been reached. In such a case, assuming no other constraints are in effect, it is beneficial to produce from wells with low GOR. The optimal production problem can be formulated as

$$\max \ \Sigma\, m_{o,i}, \tag{1}$$

subject to the constraint

$$\Sigma\, m_{g,i} = m_g^{max}. \tag{2}$$

That is, we want to maximize the total oil production at the maximum gas handling capacity. In this paper, we have chosen to use mass as the basis for the calculations, and we define the gas-oil ratio (GOR) and marginal GOR (mGOR) as

$$\text{GOR} = (m_g/m_o) \quad \text{and} \quad \text{mGOR} = (\partial m_g/\partial m_o). \tag{3}$$

The GOR can depend on the production rate. This is the case for wells producing under gas coning conditions. For optimal production, assuming that the wells are independent, it is required that

$$\text{mGOR}_{well\,i} = \text{mGOR}_{well\,j}. \tag{4}$$

is satisfied for all wells. A simple proof can be found in (Downs and Skogestad, 2011, p. 107). The optimality criterion in (4) is well known, and Urbanczyk and Wattenbarger (1994) used the mGOR in an iterative procedure to obtain optimal well rates for gas coning wells. However, they did not mention that the criterion in (4) only holds for independent wells.

At present, there exist reservoir simulators capable of predicting the GOR for different well rates. Gunnerud and Foss (2010) used such simulators to generate well performance curves, and presented an efficient real time optimizer (RTO) for solving large scale well allocation problems. However, execution times still range from minutes to tens of minutes depending on the size of the problem. If there is a sudden change (disturbance) in for example GOR or reservoir pressure between each execution of the RTO, the production will be sub-optimal.

In this paper, we propose to use reservoir simulators to make simple linear static estimators. The goal is to predict the mGOR for each well using available process measurements, e.g. pressure and valve position. Next, we propose to use simple feedback controllers, like proportional-integral (PI) controllers, to adjust the well rates such that the predicted mGOR are equal for all wells. Because we use the estimate in a feedback loop, we use the optimal closed-loop estimator presented in Ghadrdan et al. (2013). One of the benefits of this estimator is that it ensures that the gain of the estimator, from measurements to prediction, is not corrupted by measurement noise, which is essential for control purposes.

By using feedback, we can continuously reject disturbances and thus minimize the deviation from optimal operation. This control structure can also be used in combination with RTO. In such a scheme, the RTO finds optimal allocation of well rates based on detailed models and adjusts the setpoints for the feedback layer. Between each

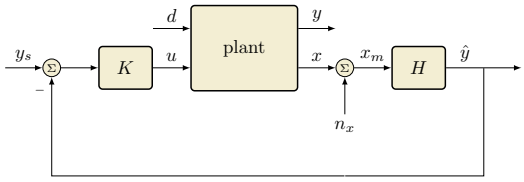

Fig. 1. Control of the predicted control variable $\hat{y}$ by K such that $\hat{y} = y_s$ by adjusting the plant input u. The estimator is used in closed-loop.

execution of the RTO, the feedback layer will attenuate disturbances and keep the production optimal.

The structure of the paper is as follows: The optimal closed-loop estimator will be presented, followed by the basis for and assumptions behind the case study model, and ending with a simple case study demonstrating the method.

2. THE OPTIMAL CLOSED-LOOP ESTIMATOR

In this section we give a brief overview of the closed-loop estimator and the assumptions behind it. For the complete derivation, please refer to Ghadrdan et al. (2013). Consider the system given in Figure 1, and a linear static estimator H on the form,

$$\hat{y} = Hx_m. \tag{5}$$

Here, $\hat{y}$ is the predicted output (controlled variable) and x_m is the measurements with measurement noise. We assume that the measurements x and the outputs y, which we want to estimate, can be expressed as linear static models;

$$x = G_x u + G_x^d d, \tag{6}$$
$$y = G_y u + G_y^d d. \tag{7}$$

The measurements x_m, which are influenced by noise n_x, are expressed as

$$x_m = x + n_x. \tag{8}$$

It is also assumed that $\dim(y) = \dim(u)$.

In closed-loop operation, the manipulated variables u are adjusted to keep the estimated controlled variables $\hat{y}$ at its setpoint y_s,

$$\hat{y} = y_s.$$

In this case, the optimal closed-loop estimator H can be found by solving the optimization problem (Ghadrdan et al., 2013)

$$H = \arg\min_H \|H\,(FW_d \;\; W_{n_x})\|_F \qquad s.t. \quad HG_x = G_y. \tag{9}$$

Here, F is the optimal sensitivity,

$$F = \left(\tfrac{\partial x}{\partial d}\right)_{y=y_s} = \left(G_x^d - G_x G_y^{-1} G_y^d\right), \tag{10}$$

and W_d and W_{n_x} are diagonal scaling matrices, representing the expected disturbance and noise. If these are worst case magnitudes, the estimator will minimize the worst case prediction error (Halvorsen et al., 2003). If they are average normally distributed deviation, the estimator will minimize the expected prediction error (Kariwala et al., 2008).

The gain of some estimators tends to approach zero when the measurement noise approaches infinity. This becomes a problem when using feedback control. Here the manipulated variable u is used to adjust $\hat{y}$. If the gain of the estimator is zero, the required u to reach setpoint approaches infinity. The closed-loop estimator avoids this problem with the constraint $HG_x = G_y$. This ensures that the gain is unaffected even in the presence of large measurement noise.

The optimization problem (2) can be written as

$$\min_H \left\|H\widetilde{F}\right\|_F \qquad \text{s.t.} \quad HG_x = G_y,$$

where $\widetilde{F} = (FW_d \;\; W_{n_x})$. Under the assumption that $\widetilde{F}\widetilde{F}^T$ is of full rank, the optimal closed-loop estimator H has the following analytical solution (Alstad et al., 2009),

$$H^T = \left(\widetilde{F}\widetilde{F}^T\right)^{-1} G_x \left(G_x^T \left(\widetilde{F}\widetilde{F}^T\right)^{-1} G_x\right)^{-1} G_y^T. \tag{11}$$

3. A SIMPLE MODEL

A simple steady-state pressure drop model has been developed for the well allocation problem (Figure 2). The model consists of three parts: model for reservoir inflow, model for pressure drop in a vertical pipe, and model for flow across a valve. The multiphase fluid of oil, water and gas is treated as a one phase pseudo fluid. These three parts can be combined to create a network of wells, manifolds, and clusters. This is a significantly simplified model, and is only intended as a demonstration.

3.1 Reservoir inflow model

The inflow relations for oil and water are assumed to follow the quadratic deliverability equation proposed by Fetkovich (1973):

$$\dot{m}_o = k_o\left(p_r^2 - p_{wf}^2\right), \tag{12}$$
$$\dot{m}_w = k_w\left(p_r^2 - p_{wf}^2\right). \tag{13}$$

The flow of gas is given by the gas-oil ratio

$$\dot{m}_g = \mathrm{GOR} \times \dot{m}_o. \tag{14}$$

To represent gas coning conditions, we have assumed the GOR to have the following relation to pressure,

$$\mathrm{GOR} = \frac{k_g}{k_o}\left(p_r - p_{wf}\right)^2. \tag{15}$$

This implies a rapid increase in GOR with increasing production.

3.2 One phase pseudo fluid

To simplify our model, we approximate the multiphase fluid (liquid and gas) as a one-phase pseudo fluid. Neglecting mixing volumes, the density of the pseudo fluid is approximated by its volumetric average

$$\rho_{mix} = v_g \rho_g^{ig} + v_o \rho_o + v_w \rho_w. \tag{16}$$

where ρ is the density and v is the volume fraction of the respective phase. In terms of mass flows, the overall density becomes

$$\rho_{mix} = \frac{\dot{m}_o + \dot{m}_g + \dot{m}_w}{\dot{m}_g/\rho_g^{ig} + \dot{m}_o/\rho_o + \dot{m}_w/\rho_w}. \tag{17}$$

We assume that oil and water are incompressible, and that the gas behaves as an ideal gas,

$$\rho_g^{ig} = \frac{p\, M_g}{R\, T}, \tag{18}$$

where M_g is the molar weight of the gas, R is the ideal gas constant and T the temperature.

3.3 Pressure drop through a vertical pipe

We estimate the pressure drop for multiphase flow in a vertical pipe using the stationary mechanical energy balance. Assuming no slip between the phases and neglecting friction, work and kinetic energy, the energy balance becomes

$$\mathrm{d}p = \rho_{mix}\, g\, \mathrm{d}h. \tag{19}$$

Integrating (19) between the limits $(p_1,\, h_1)$ and $(p_2,\, h_2)$ gives

$$\alpha\, \ln(p_2/p_1) + \beta\, (p_2 - p_1) = \dot{m}_{tot}\, g\, \Delta h, \tag{20}$$

where $\Delta h = h_2 - h_1$,

$$\alpha = \dot{m}_g\, R\, T/M_g, \quad \text{and} \quad \beta = \dot{m}_o/\rho_o + \dot{m}_w/\rho_w.$$

The equation cannot be solved directly for the pressure p_2. By using a serial expansion of the natural logarithm,

$$\ln\left(p_2/p_1\right) = \ln\left(p_1 + \Delta p/p_1\right) = \ln\left(1 + \Delta p/p_1\right) \approx \Delta p/p_1, \tag{21}$$

the pressure drop over the pipeline can be expressed as a function of the pipe length inlet pressure and height Δh,

$$\Delta p = p_2 - p_1 = \frac{\dot{m}_{tot}\, g\, p_1\, \Delta h}{\alpha + p_1\, \beta}. \tag{22}$$

3.4 Pressure drop across a valve

The mass flow across a valve is assumed given by a standard valve equation,

$$\dot{m} = f(z)\, C_d\, A \sqrt{\rho\left(p_2 - p_1\right)}, \tag{23}$$

where C_d is the valve constant, A is the cross section area, and p_1 and p_2 are the pressures on each side of the valve. $f(z)$ is the valve characteristics, with the valve opening z ranging between 0 when fully closed and 1 when fully open. For simplicity, we assume linear valve characteristics

$$f(z) = z, \quad \text{where} \quad z \in [0,\, 1]. \tag{24}$$

We assume a one-phase pseudo fluid and the density in (23) is calculated as the average density for the two sides of the valve;

$$\rho = \tfrac{1}{2}\left(\rho_{mix,1} + \rho_{mix,2}\right). \tag{25}$$

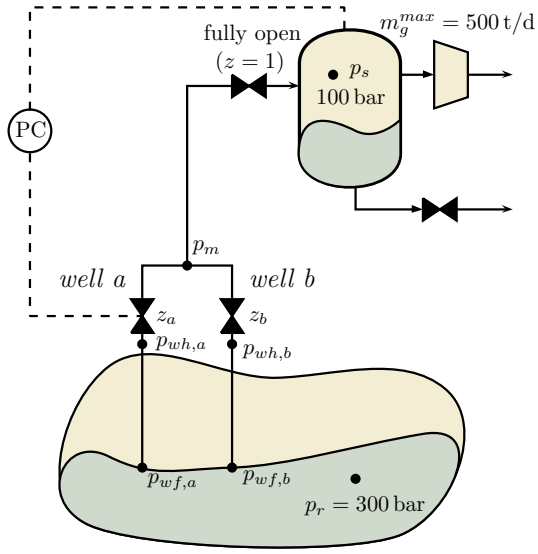

Fig. 2. Sketch of the two-well case study with all relevant nomenclature. Pressure control is shown varying *well a*, but also other options are studied.

4. CASE STUDY

We consider a two-well case (*well a* and *well b*) with a separator as shown in Figure 2. We assume the separator is operated at its maximum pressure (100 bar), which together with a given maximum compressor work, give a maximum gas handling capacity of 500 t/d. The flow characteristics as a function of the well flow pressure p_{wf} for the two wells are shown in Figures 3 and 4. Notice the sharp increase in gas production with higher production (lower flow pressure p_{wf}). For simplicity, only the well valves can be manipulated, and the top valve is fixed to fully open ($z = 1$). For nominal operation, the optimal production is given in Table 1.

For each well the predicted output (controlled variable) is

$$\hat{y} = \mathrm{mGOR} = H x_m,$$

and for each well, the manipulated variable, measurements, and disturbances are as follows,

$$u = z, \quad x = \begin{pmatrix} p_{wf} \\ p_{wh} \\ p_m \\ z \end{pmatrix}, \quad \text{and} \quad d = \begin{pmatrix} k_g \\ p_r \\ p_m \end{pmatrix}. \tag{26}$$

The disturbance k_g represents a shift in the GOR, and p_m represents an upstream disturbance e.g. a change in gas handling capacity.

4.1 The method

To obtain a simple method, we evaluate each well individually. This can be done by assuming that the manifold pressure p_m is a disturbance and independent of the other flow to the manifold. Because we control the separator pressure

Table 1. Optimal well allocation under nominal operation (no disturbances)

		well a	*well b*	*total*
z	—	0.5177	0.8505	—
p_{wf}	bar	214.6	204.5	—
p_{wh}	bar	190.9	173.8	—
p_m	bar	—	—	151.1
m_o	t/d	289.9	265.0	555.0
m_g	t/d	263.7	236.3	500.0
m_w	t/d	145.0	496.3	641.3

p_s, this is not quite true. The manifold pressure p_m will depend on the flow from the other well, and the optimality criterion in (4) does not hold; However, the effect of this interaction is thought to be quite small. The more wells connected to the manifold, the less influence the individual well will have on the manifold pressure. For an infinite number of wells, the manifold pressure will be independent of the individual well flows and the optimality criterion (4) holds. The wells can always be made independent of each other by controlling the manifold pressure instead of the separator pressure.

In addition, inputs and disturbances are assumed to be independent of each other in the derivation of the closed-loop estimator. This is clearly not the case for the input z and the upstream disturbance p_m. Nevertheless, we have

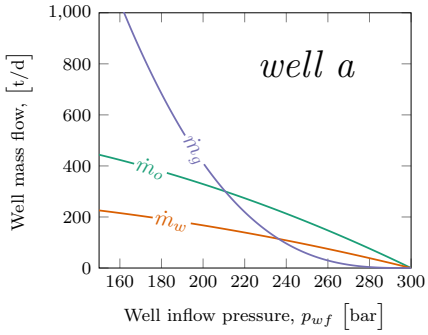

Fig. 3. Well inflow characteristics for *well a*

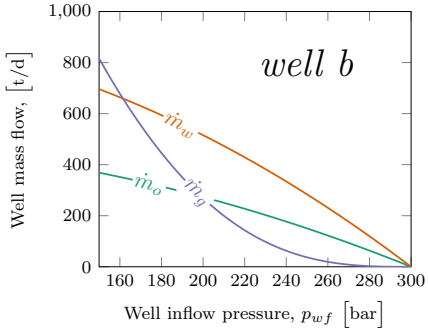

Fig. 4. Well inflow characteristics for *well b*

neglected this interaction in order to evaluate each well individually. We could have used the separator pressure p_s to represent the upstream disturbances. However, when finding the estimator we would have to consider the whole well network.

4.2 The closed-loop estimator

The linear static model for each well was approximated by subjecting the respective well to a 1% positive change in the manipulated variable u, and disturbances d. The mGOR was approximated by linear approximation $\Delta m_g/\Delta m_o$ from the nominal to the new steady state. The expected disturbance is assumed to be $\pm 10\%$, and the measurement noise is assumed to be ± 0.1 bar for pressures and ± 0.01 for valve position.

This gives the following optimal closed-loop estimators for *well a* and *well b*,

$$H_{well\,a} = (-1.7900 \;\; 2.0128 \;\; 0.3735 \;\; 35.7994)\,, \tag{27}$$
$$H_{well\,b} = (-1.0521 \;\; 1.1297 \;\; 0.4626 \;\; 24.3815)\,. \tag{28}$$

For this two-wells case, we have two manipulated variables and one operational constraint (maximum separator pressure). One of the manipulated variables must control the pressure, leaving only one free manipulated variable. This gives three different control structures:

Open-loop a
well a valve is fixed (z_a fixed) and *well b* valve controls separator pressure p_s.

Open-loop b
well b valve is fixed (z_b fixed) and *well a* valve controls separator pressure p_s (Figure 2).

Closed-loop mGOR
one well controls the optimality condition, the other controls the pressure p_s.

In general, for closed-loop mGOR with n wells and n valves, one manipulated variable would be used to control pressure. The remaining $n-1$ valves would be used to ensure equal mGOR for all wells.

We compared the different control structures when subjecting the system to sets of disturbances. Sub-optimality of a structure is quantified in terms of loss. We define loss as the difference between oil production with a given control structure J and the optimal oil-production $J^\star$, for a given disturbance. Mathematically this becomes

$$loss(d) = J(d) - J^\star(d). \tag{29}$$

It is clearly seen from the substantially lower oil-loss that the closed-loop estimator performs better than the open-loop strategies for combined disturbances (Table 2).

For individual disturbances, the open-loop policies had a smaller loss if the disturbance is in the other well. In some cases they also performed better than the closed-loop mGOR policy based on (4). This may seem surprising; one would expect the open-loop strategies to always have worse performance as there is no correction for disturbances. At a closer inspection, the worst control strategy is to fix the valve position on wells that are subjected to disturbances; For example, fixing the valve on *well a* when there is a disturbance down hole in *well a*. Naturally, the best

Table 2. Comparison of control strategies with combinations of disturbances affecting the system.

Disturbance		*Optimal*	*Open-loop a*	*Open-loop b*	*Closed-loop mGOR*
well a: 10% inc. k_g *well b*: 10% inc. k_g	z_a	0.4872	0.5177	0.4482	0.4778
	z_b	0.7682	0.7080	0.8505	0.7873
	oil-loss	—	0.3030 t/d	0.4995 t/d	**0.0284 t/d**
well a: 10% inc. k_g *well b*: 1% dec. p_r	z_a	0.4868	0.5177	0.5077	0.5001
	z_b	0.9063	0.8250	0.8505	0.8703
	oil-loss	—	0.3112 t/d	0.1426 t/d	**0.0582 t/d**
well a: 10% inc. k_g 5% dec. m_g^{max}	z_a	0.4576	0.5177	0.4373	0.4615
	z_b	0.8018	0.6703	0.8505	0.7927
	oil-loss	—	1.3579 t/d	0.1554 t/d	**0.0057 t/d**
well a: 10% inc. k_g *well b*: 5% inc. k_g 5% dec. m_g^{max}	z_a	0.4589	0.5177	0.4176	0.4555
	z_b	0.7582	0.6395	0.8505	0.7656
	oil-loss	—	1.3009 t/d	0.6467 t/d	**0.0045 t/d**

Table 3. Comparison of control strategies with different isolated disturbances affecting the system.

Disturbance		*Optimal*	*Open-loop a*	*Open-loop b*	*Closed-loop mGOR*
well a: 10% inc. k_g	z_a	0.4846	0.5177	0.4882	0.4913
	z_b	0.8593	0.7817	0.8505	0.8430
	oil-loss	—	0.3562 t/d	**0.0042 t/d**	0.0146 t/d
well b: 10% inc. k_g	z_a	0.5202	0.5177	0.4744	0.5031
	z_b	0.7604	0.7652	0.8505	0.7931
	oil-loss	—	**0.0018 t/d**	0.6033 t/d	0.0835 t/d
well a: 1% dec. p_r	z_a	0.5342	0.5177	0.5366	0.5318
	z_b	0.8557	0.8916	0.8505	0.8608
	oil-loss	—	0.0686 t/d	0.0015 t/d	**0.0014 t/d**
well b: 1% dec. p_r	z_a	0.5199	0.5177	0.5388	0.5272
	z_b	0.8970	0.9027	0.8505	0.8789
	oil-loss	—	**0.0014 t/d**	0.1001 t/d	0.0148 t/d
10% dec. m_g^{max}	z_a	0.4597	0.5177	0.4110	0.4565
	z_b	0.7422	0.6269	0.8505	0.7491
	oil-loss	—	1.2854 t/d	0.9176 t/d	**0.0040 t/d**

Table 4. Comparison of closed-loop estimators with reduced number of measurements.

Disturbance		$x^T = (p_{wf}\ p_{wh}\ p_m\ z)$	$x^T = (p_{wh}\ p_m\ z)$	$x^T = (p_{wh}\ z)$
well a 10% inc. k_g *well b* 10% inc. k_g	z_a	0.4778	0.4788	0.4786
	z_b	0.7873	0.7853	0.7858
	oil-loss	0.0284 t/d	**0.0227 t/d**	0.0241 t/d
well a 10% inc. k_g *well b* 1% dec. p_r	z_a	0.5001	0.5066	0.5058
	z_b	0.8703	0.8533	0.8554
	oil-loss	**0.0582 t/d**	0.1282 t/d	0.1183 t/d
well a 10% inc. k_g 5% dec. m_g^{max}	z_a	0.4615	0.4678	0.4672
	z_b	0.7927	0.7782	0.7797
	oil-loss	**0.0057 t/d**	0.0391 t/d	0.0344 t/d
well a 10% inc. k_g *well b* 5% inc. k_g 5% dec. m_g^{max}	z_a	0.4555	0.4591	0.4585
	z_b	0.7656	0.7579	0.7591
	oil-loss	0.0045 t/d	**0.0000 t/d**	0.0001 t/d

solution appears when there is control action on the well that is subjected to disturbance. The closed-loop strategy is always close to the optimal solution.

4.3 The effect of number of measurements

We have so far used 4 measurements for each well, see (26). Because p_m is the same for both wells, closed-loop mGOR uses 7 independent measurements.

To investigate the effect of fewer available measurements, closed-loop estimators were made for the two following cases for each well,

$$x_3 = \begin{pmatrix} p_{wh} \\ p_m \\ z \end{pmatrix} \quad \text{and} \quad x_2 = \begin{pmatrix} p_{wh} \\ z \end{pmatrix}. \tag{30}$$

The resulting closed-loop estimators for three available measurements for each well (5 measurements overall) are

$$H_{well\,a,3} = (0.1949\ 0.1709\ 69.7064)\,,$$
$$H_{well\,b,3} = (0.2419\ 0.1230\ 43.0890)\,.$$

In the case of two available measurements for each well (4 measurements overall), the closed-loop estimator are

$$H_{well\,a,2} = (0.2117\ 70.3486)\,,$$
$$H_{well\,b,2} = (0.2719\ 43.7314)\,.$$

The performance of the estimators with a reduced number of measurements is surprisingly good in some cases (Table 4). For the three measurement estimator, one of the disturbance scenarios actually results in zero loss, outperforming the four measurement estimator. This is a coincidence due to the particular numerical values, and shows that one must be careful about drawing conclusions. Nevertheless, the good result indicates that the method is still applicable when few measurements are available. However, if the measurements contain noise, the estimators using more measurements would most likely have better performance.

5. CONCLUSION

In this paper, we have used the optimal closed-loop estimator to predict the mGOR for different wells using pressure and valve-opening measurements. By adjusting the respective well rates, these mGOR estimates have been controlled such that they are equal for all wells. We have shown that this method gives close-to-optimal production despite disturbances in GOR, reservoir pressure and gas handling capacity. One important advantage of this method is that the estimators can be generated for each well independently of the other wells in the field. The method is also well suited for use with simple feedback controllers, like a PI controller, for efficiently rejecting disturbances.

ACKNOWLEDGEMENTS

We would like to thank Rannei Solbak Simonsen for the work done on the case study model during her 5th year project at NTNU.

REFERENCES

Alstad, V., Skogestad, S., and Hori, E.S. (2009). Optimal measurement combinations as controlled variables. *Journal of Process Control*, 19(1), 138–148.

Downs, J.J. and Skogestad, S. (2011). An industrial and academic perspective on plantwide control. *Annual Reviews in Control*, 35(1), 99 – 110. doi: http://dx.doi.org/10.1016/j.arcontrol.2011.03.006.

Fetkovich, M. (1973). The isochronal testing of oil wells. In *Fall Meeting of the Society of Petroleum Engineers of AIME*, 137–142. Society of Petroleum Engineers. doi: 10.2118/4529-MS.

Ghadrdan, M., Grimholt, C., and Skogestad, S. (2013). A new class of model-based static estimators. *Industrial & Engineering Chemistry Research*, 52(35), 12451–12462.

Gunnerud, V. and Foss, B. (2010). Oil production optimization-A piecewise linear model, solved with two decomposition strategies. *Computers and Chemical Engineering*, 34(11), 1803–1812.

Halvorsen, I.J., Skogestad, S., Morud, J.C., and Alstad, V. (2003). Optimal selection of controlled variables. *Industrial & Engineering Chemistry Research*, 42(14), 3273–3284.

Kariwala, V., Cao, Y., and Janardhanan, S. (2008). Local self-optimizing control with average loss minimization. *Industrial & Engineering Chemistry Research*, 47(4), 1150–1158.

Urbanczyk, C.H. and Wattenbarger, R. (1994). Optimization of Well Rates Under Gas Coning Conditions. *SPE Advanced Technology Series*, 2(02), 61–68.

Appendix A. CASE STUDY MODEL PARAMETERS

Table A.1. Parameters for the case study model

		well a	*well b*
p_r	bar	300	300
k_o	$\mathrm{t/bar^2}$	6.576×10^{-3}	5.462×10^{-3}
k_g	$\mathrm{t/bar^4}$	8.239×10^{-7}	5.373×10^{-7}
k_w	$\mathrm{t/bar^2}$	3.344×10^{-3}	1.031×10^{-2}
h	m	1000	1000
T	K	373	373
M_g	kg/kmol	16.04	16.04
ρ_o	$\mathrm{kg/m^3}$	800	800
ρ_w	$\mathrm{kg/m^3}$	1000	1000
R	J/(kmol K)	8314	8314
C_d	$(\mathrm{kg/m\,bar\,d^2})^{0.5}$	84600	84600
A	$\mathrm{m^2}$	5.66×10^{-4}	5.66×10^{-4}
		riser model	
h	m	1500	
C_d	$(\mathrm{kg/m\,bar\,d^2})^{0.5}$	84600	
A	$\mathrm{m^2}$	0.0011	

Bibliography

E.J. Adam, H.A. Latchman, and O.D. Crisalle. Robustness of the Smith predictor with respect to uncertainty in the time-delay parameter. In *American Control Conference, 2000. Proceedings of the 2000*, volume 2, pages 1452–1457 vol.2, 2000.

S. Alcántara, C. Pedret, and R. Vilanova. On the model matching approach to PID design: analytical perspective for robust servo/regulator tradeoff tuning. *Journal of Process Control*, 20(5): 596–608, 2010.

S. Alcántara, R. Vilanova, and C. Pedret. PID control in terms of robustness/performance and servo/regulator trade-offs: A unifying approach to balanced autotuning. *Journal of Process Control*, 23(4): 527–542, 2013.

V.M. Alfaro, R. Vilanova, V. Méndez, and J. Lafuente. Performance/robustness tradeoff analysis of PI/PID servo and regulatory control systems. In *2010 IEEE International Conference on Industrial Technology*, pages 111–116, March 2010.

Karl Johan Åström and Tore Hägglund. *Automatic Tuning of PID Controllers*. Instrument Society of America, 1988. ISBN 978-1-55617-081-2.

Karl Johan Åström and Tore Hägglund. *Advanced PID control*. International Society of Automation, 2006. ISBN 978-1-55617-942-6.

Karl Johan Åström, Chang Chieh Hang, Per Persson, and Weng Khuen Ho. Towards intelligent PID control. *Automatica*, 28:1–9, 1992.

Karl Johan Åström, Hélène Panagopoulos, and Tore Hägglund. *PID Controllers - Theory, Design, and Tuning*. Instrument Society of America, 2 edition, 1995. ISBN 978-1-55617-516-9.

Karl Johan Åström, Hélène Panagopoulos, and Tore Hägglund. Design of PI controllers based on non-convex optimization. *Automatica*, 34(5):585–601, 1998.

Jens G. Balchen. *A performance index for feedback control systems based on the Fourier transform of the control deviation*. Number MA1 in Acta Polytechnica Scandinavica: Mathematics and Computing Machinery Series No.1. Norges tekniske vitenskapsakademi, Trondheim, Norway, 1958. Acta P 247/1958.

Josefin Berner. *Automatic Tuning of PID Controllers based on Asymmetric Relay Feedback*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Lund University, 2015.

W.L. Bialkowski. Control of the pulp and paper making process. In William S. Levine, editor, *The Control Handbook*, chapter 72, pages 1219–1243. CRC Press and IEEE Press, 1996.

Lorenz T. Biegler. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2010. ISBN 978-0-89871-702-0.

Stephen P. Boyd and Craig H. Barratt. *Linear controller design: limits of performance*. Prentice Hall, Englewood Cliffs, NJ, USA, 1991. ISBN 0-13-538687-X.

E.B. Dahlin. Designing and tuning digital controllers. *Instruments and Control systems*, 41(6):77–83, 1968.

David Di Ruscio. On tuning PI controllers for integrating plus time delay systems. *Modeling, Identification and Control*, 31(4):145–164, 2010.

Martin S. Foss. Validation of the SIMC PID tuning rules. Technical report, Norwegian University of Science and Technology, 2012. 5th year specialization project in process system engineering.

Olof Garpinger and Tore Hägglund. A software tool for robust PID design. In *Proc. 17th IFAC World Congress, Seoul, Korea*, 2008.

Olof Garpinger, Tore Hägglund, and Karl Johan Åström. Performance and robustness trade-offs in PID control. *Journal of Process Control*, 24(5):568–577, 2014.

Chriss Grimholt and Sigurd Skogestad. Optimal PI-control and verification of the SIMC tuning rule. In *IFAC conference on Advances in PID control (PID'12)*. The International Federation of Automatic Control, March 2012a.

Chriss Grimholt and Sigurd Skogestad. The SIMC method for smooth pid controller tuning. In Ramon Vilanova and Antonio Visioli, editors, *PID control in the third Millennium – Lessons Learned and new approaches*. Springer, 2012b.

Chriss Grimholt and Sigurd Skogestad. Optimal PID-control on first order plus time delay systems & verification of the SIMC rules.

In *10th IFAC International Symposium on Dynamics and Control of Process Systems*, 2013.

Chriss Grimholt and Sigurd Skogestad. Improved optimization-based design of PID controllers using exact gradients. In *12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering*, volume 37, pages 1751–1757, 2015a.

Chriss Grimholt and Sigurd Skogestad. Optimization of oil field production under gas coning conditions using the optimal closed-loop estimator. In *2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*, pages 39–44, May 2015b.

Chriss Grimholt and Sigurd Skogestad. Optimal PID control of double integrating processes. In *11th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems*, pages 127–132, NTNU, Trondheim, Norway, 6 2016.

Chriss Grimholt and Sigurd Skogestad. Optimization of fixed order controllers using exact gradients. *Journal of Process Control*, 71: 130–138, 2018a.

Chriss Grimholt and Sigurd Skogestad. Optimal PI and PID control of first-order plus delay processes and evaluation of the original and improved SIMC rules. *Journal of Process Control*, 70, 2018b.

Chriss Grimholt and Sigurd. Skogestad. Should we forget the smith predictor? In *3rd IFAC conference on Advances in PID control, Ghent, Belgium, 9-11 May 2018.*, 2018c.

Roman Gudin and Leonid Mirkin. On the delay margin of dead-time compensators. *International Journal of Control*, 80(8):1316–1332, 2007.

Tore Hägglund. An industrial dead-time compensating PI controller. *Control Engineering Practice*, 4(6):749–756, 1996.

Albert C. Hall. *The analysis and synthesis of linear servomechanisms*. Technology Press Massachusetts Institute of Technology, 1943.

Galal Ali Hassaan. Controller tuning for disturbance rejection associated with delayed double integrating processes: Part i: PD-PI controller. *International Journal of Computer Techniques*, 2(3):110–115, 2015.

Martin Hast, Karl Johan Åström, Bo Bernhardsson, and Stephen Boyd. Pid design by convex-concave optimization. In *Proceedings European Control Conference*, pages 4460–4465, 2013.

Finn Haugen. Comparing PI tuning methods in a real benchmark temperature control system. *Modeling, Identification and Control*, 31 (3):79–91, 2010.

P. Hazebroek and B.L. Van der Waerden. The optimum tuning of regulators. *Trans. ASME*, 72:317–322, 1950.

Axel Lødemel Holene. Performance and robustness of smith predictor control and comparison with PID control. Master's thesis, Norwegian University of Science and Technology, Trondheim, 2013.

Mikulas Huba. Performance measures, performance limits and optimal PI control for the IPDT plant. *Journal of Process Control*, 23 (4):500–515, 2013.

Ari Ingimundarson and Tore Hägglund. Performance comparison between PID and dead-time compensating controllers. *Journal of Process Control*, 12(8):887–895, 2002.

E. Jahanshahi, V. De Oliveira, C. Grimholt, and Skogestad S. A comparison between internal model control, optimal PIDF and robust controllers for unstable flow in risers. In *19th World Congress*. The International Federation of Automatic Control, 2014.

Birgitta Kristiansson and Bengt Lennartson. Robust PI and PID controllers including Smith predictor structure. In *American Control Conference, 2001. Proceedings of the 2001*, volume 3, pages 2197–2202, 2001.

Birgitta Kristiansson and Bengt Lennartson. Robust and optimal tuning of PI and PID controllers. *IEE Proceedings-Control Theory and Applications*, 149(1):17–25, 2002.

Birgitta Kristiansson and Bengt Lennartson. Evaluation and simple tuning of PID controllers with high-frequency robustness. *Journal of Process Control*, 16(2):91–102, 2006.

Per-Ola Larsson and Tore Hagglund. Robustness margins separating process dynamics uncertainties. In *European Control Conference (ECC), Budapest*, pages 543–548. IEEE, 2009.

Per-Ola Larsson and Tore Hägglund. Comparison between robust PID and predictive PI controllers with constrained control signal noise sensitivity. In *IFAC Conf. Advances in PID Control*, volume 12, pages 175–180, 2012.

Jietae Lee, Wonhui Cho, and Thomas F. Edgar. Simple analytic PID controller tuning rules revisited. *Industrial & Engineering Chemistry Research*, 53(13):5038–5047, 2014.

Tao Liu, Xing He, D.Y. Gu, and W.D. Zhang. Analytical decoupling control design for dynamic plants with time delay and double

integrators. *IEE Proceedings-Control Theory and Applications*, 151(6): 745–753, 2004.

Petter Lundström. *Studies on Robust Multivariable Distillation Control*. PhD thesis, Norwegian University of Science and Technology, July 1994.

Manfred Morari and Evanghelos Zafiriou. *Robust process control*. Prentice Hall, NJ, USA, 1989. ISBN 978-0-13782-153-2.

J.A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

G.C. Newton, L.A. Gould, and J.F. Kaiser. *Analytical design of linear feedback controls*. John Wiley & Sons, New York, NY, USA, 1957.

Julio Elias Normey-Rico and Eduardo F. Camacho. *Control of dead-time processes*. Springer Science & Business Media, 2007. ISBN 978-1-84628-828-9.

Aidan O'Dwyer. *Handbook of PI and PID controller tuning rules*. Imperial College Press, 2 edition, 2006. ISBN 978-1-86094-622-6.

Zalman J. Palmor. Stability properties of Smith dead-time compensator controllers. *International Journal of Control*, 32(6):937–949, 1980.

Hélène Panagopoulos, Karl Johan Åström, and Tore Hägglund. Design of PID controllers based on constrained optimisation. *IEE Proceedings-Control Theory and Applications*, 149(1):32–40, 2002.

Igor Podlubny. Fractional-order systems and $PI^{\lambda}D^{\mu}$-controllers. *IEEE Transactions on automatic control*, 44(1):208–214, 1999.

Venkatesh G. Rao and Dennis S. Bernstein. Naive control of the double integrator. *IEEE Control Systems*, 21(5):86–97, 2001.

Jacques Richalet and Donal O'Donovan. *Predictive functional control: principles and industrial applications*. Springer Science & Business Media, 2009. ISBN 978-1-84882-492-8.

Jacques Richalet, A. Rault, J.L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5):413–428, 1978.

Danlel E. Rivera, Manfred Morarl, and Sigurd Skogestad. Internal model control. 4. PID controller design. *Ind. Eng. Chem. Process Des. Dev.*, 25:252–256, 1986.

Tor Steinar Schei. Automatic tuning of PID controllers based on transfer function estimation. *Automatica*, 30(12):1983–1989, 1994.

Vanessa Romero Segovia, Tore Hägglund, and Karl Johan Åström. Design of measurement noise filters for PID control. In *IFAC World Congress 2014*, volume 19, pages 8359–8364, 2014.

M. Shamsuzzoha and Moonyong Lee. PID controller design for integrating processes with time delay. *Korean Journal of Chemical Engineering*, 25(4):637–645, 2008.

M. Shamsuzzoha and Sigurd Skogestad. The setpoint overshoot method: A simple and fast method for closed-loop PID tuning. *Journal of Process Control*, 20:1220–1234, 2010.

F.G. Shinskey. How good are our controllers in absolute performance and robustness? *Measurement and Control*, 23(4):114–121, 1990.

Sigurd Skogestad. Simple analytic rules for model reduction and PID controller tuning. *Journal of process control*, 13(4):291–309, 2003.

Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control – Analysis and Design*. John Wiley & Sons, Ltd, 2 edition, 2005. ISBN 978-0-470-01168-3.

Otto Smith. Closer control of loops with dead time. *Chemical engineering progress*, 53(5):217–219, 1957.

Kristian Soltesz, Chriss Grimholt, and Sigurd Skogestad. Simultaneous design of pid controller and measurement filter by optimization. *IET Control Theory & Applications*, 11(3):341–348, 2017a.

Kristian Soltesz, Chriss Grimholt, and Sigurd Skogestad. Simultaneous design of pid controller and measurement filter by optimization. *IET Control Theory & Applications*, 11(3):341–348, 2017b.

Murray P. Spiegel. *Schaum's outline series: Theory and Problems of Laplace Transforms*. McGraw-Hill Book Company, 1965.

John G. Ziegler and Nathaniel B. Nichols. Optimum settings for automatic controllers. *Trans. ASME*, 64(11):759–768, 1942.