Analysis and Design of Software-Based Optimal PID Controllers

Garpinger, Olof

Analysis and Design of Software-Based Optimal PID Controllers

Olof Garpinger

LUND
UNIVERSITY

Department of Automatic Control

Present, Past, Future. The picture on the upper hand of the front cover shows two parts of a longcase clock made by the Swedish woodcarver and artist Jöns Mårtensson. The parts have been slightly modified by the thesis author.

PhD Thesis
ISRN LUTFD2/TFRT--1105--SE
ISBN 978-91-7623-271-2 (print)
ISBN 978-91-7623-272-9 (web)
ISSN 0280–5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Media-Tryck.
Lund 2015

To Linda-Marie and Wilmer

Abstract

A large process industry can have somewhere between five hundred and five thousand control loops, and PID controllers are used in 90–97% of the cases. It is well-known that only 20–30% of the controllers in the process industry are tuned satisfactorily, but with the methods available today it is considered too time-consuming to optimize each single controller. This thesis presents tools for analysis and design of optimal PID controllers, and suggests when and how to use them efficiently. High performing low-level controllers are also likely to be beneficial for higher-level advanced process control, thus promoting the economy of whole factories.

Controller design is often a trade-off between conflicting criteria, such as load disturbance attenuation, robustness, and noise sensitivity. In this thesis, a MATLAB®-based software tool is used to solve a constrained optimization problem, with respect to all three requirements. This gives tuning of both the PID parameters and a low-pass filter time constant.

A large batch of benchmark models, representative for the process industry, has been used throughout the whole thesis for controller analysis. This includes comparisons between PID controllers derived using either optimization or tuning rules. Trade-off plots are also presented, which explicitly show the relationships between performance, robustness and the PID parameters.

A new procedure for software-based optimal PID design is suggested, which leads to a set of PID, PI, and I controllers. The user can then select the best performing controller with an acceptable control signal activity. It is shown that the resulting controllers are optimal or near optimal with respect to the three above mentioned criteria. The same procedure can also be used to analyze the benefit of the derivative part by comparing optimal PI and PID controllers with the same level of noise sensitivity. The efficiency of the procedure is demonstrated on an industrial friction stir welding machine. For a more wide-spread use of the proposed procedure, it is shown that better modeling techniques are needed, and guidelines for such methods are also included.

Acknowledgments

First, I would like to thank all my colleagues at the Department of Automatic Control, Lund University, for giving me such a warm welcome back after spending a few years in the industry. I am especially grateful to Tore Hägglund and Karl-Johan Åström for asking me to come back and for convincing me that it was a good idea. I have really enjoyed working together with all of you again.

I am very fortunate to have had Tore Hägglund as my supervisor. My work is deeply influenced by his ambition to make research useful in practice, and he has opened my eyes to the beauty of simplicity. From a personal point of view, I truly believe that his wisdom has helped me become a better man.

It is a rare luxury to have worked with someone as talented and full of energy as Karl-Johan Åström. I want to thank you for all your advice over the years, it has granted me a lot of insight into PID control.

During my first years at the department, I shared (countless) offices with Per-Ola Larsson. I am very thankful for the many discussions we have had, and during which I think we converged to more or less the same visions for the future of PID design.

I would like to thank SKB for giving me the opportunity to work on their friction stir welding machine. A special thanks to Lars Cederqvist, for helping me regain my self confidence and for introducing me to FSW. I am grateful to Anders Robertsson for introducing me to Lars. I am also glad that I have had the chance to work with two brilliant control engineers on the SKB project, namely Anton Cervin and Isak Nielsen.

Several of my colleagues have helped me proofread my articles and thesis. Therefore, I would like to thank: Tore Hägglund, Per-Ola Larsson, Karl-Johan Åström, Lars Cederqvist, Anton Cervin, Kristian Soltesz, Martin Hast, and Anders Robertsson.

I am very thankful for all the help Leif Andersson has provided during the work on this thesis. His knowledge in LaTeX is invaluable. I would also

like to thank the rest of our Research Engineers as well as our Administrators for making the department run so smoothly. This includes our previous employees: Eva Schildt, Britt-Marie Mårtensson, Agneta Tuszynski, Lizette Borgeram, and Rolf Braun.

I would like to send my warmest appreciations to both my own parents and my parents-in-law, for helping me and my wife finish our theses through countless hours of babysitting. Finally, I would like to say that I am the most fortunate man to have such a wonderful wife as Linda-Marie, and such lovely son as Wilmer. You mean the world to me.

Contents

Preface

Contents and contributions of the thesis

This thesis consists of six introductory chapters and five papers. This section describes the contents of the introductory chapters and the contributions of each paper.

Chapter 1 – Introduction

The closed-loop system used to formulate a constrained optimization problem for PID controller design is defined in this chapter. The different parts of the PID controller are also defined. Finally, the thesis is put into a process industrial context. Notice that the theory presented in Sections 1.1–1.2 is standard PID material, see e.g. [Åström and Hägglund, 2005].

Chapter 2 – Models and modeling for the process industry

Some commonly used process models and modeling techniques for the process industry are introduced here together with a simple measure for process classification. A large batch of process models representative for the process industry is also presented and motivated.

Chapter 3 – Criteria and trade-offs for PID design

Several commonly used performance, robustness and noise sensitivity criteria are presented, and the specific choices made for this thesis are motivated. The chapter is ended with the formulation of a constrained optimization problem for design of a PID controller with a low-pass measurement noise filter.

Chapter 4 – PID design methods

Several commonly used PID tuning rules are introduced and their popularity is explained. The second section of the chapter presents different optimization-based PID design methods.

Chapter 5 – Friction stir welding

The friction stir welding process is explained here together with a brief description of the specific control application to seal copper canisters for nuclear waste.

Chapter 6 – Thesis contributions

Thesis objectives, contributions and future visions for software-based optimal PID design are presented in this chapter.

Paper I

Garpinger, O., T. Hägglund, and K. J. Åström (2014). "Performance and robustness trade-offs in PID control". *Journal of Process Control* **24**:5, pp. 568–577.

This paper introduces new trade-off plots for PID control that show explicitly how performance, with respect to load disturbances, and robustness depend on the controller parameters. Plots are given for processes with lag-dominated, balanced and delay-dominated dynamics, for both PI and PID control. Both input and output disturbances are considered for PI control, and for PID control noise filtering is briefly analyzed. The plots are also used to show strengths and weaknesses of some common PI tuning rules.

O. Garpinger and K.J. Åström constructed the trade-off plots. The development of the plots has been carried out by all three authors through numerous discussions. The idea to compare different tuning rules came from O. Garpinger. The parametrization of optimal controllers was derived by K.J. Åström. K.J. Åström and O. Garpinger have written most of the paper, with assistance from T. Hägglund.

Paper II

Garpinger, O. and T. Hägglund (2008). "A Software Tool for Robust PID Design". In: *17th IFAC World Congress*. Seoul, South Korea.

The MATLAB®-based software presented in this paper finds optimal PID controllers with respect to minimum integrated absolute error during a unit load disturbance on the process input and H_∞ robustness constraints on the sensitivity and complementary sensitivity functions. The optimization problem is solved using a simplex-based algorithm called the Nelder-Mead method. The resulting controllers are shown to give good control for a large batch of process models that are representative for the process industry. The software can both provide very good controllers in a short amount of time and be useful for PID control analysis.

The MATLAB®-based software was written by O. Garpinger, inspired by the previous work by P. Nordfeldt (see citation in the paper). T. Hägglund contributed with support and experience from PID design. The paper was written by O. Garpinger with assistance from T. Hägglund.

Paper III

Garpinger, O. and T. Hägglund (2014). "Modeling for Optimal PID Design". In: *19th IFAC World Congress*. Cape Town, South Africa.

In this paper, three commonly used PI and PID tuning rules are compared to optimal controller design with respect to performance and robustness. First order time delayed model approximations, derived from step response tests, are used. All four methods are shown to have weaknesses in terms of performance and robustness variation or the lack of a tuning parameter, especially for PID control. It is investigated what process information is desirable for design based on constrained optimization. For PI control it is enough to have information around a single phase angle, while for PID control the desirable information depends on the normalized time delay of the process. Given such improved models it is shown that performance and robustness can be kept close to optimal.

The comparison of different tuning methods was carried out by O. Garpinger. The desired process information for optimal PI and PID design was investigated by O. Garpinger. O. Garpinger also derived the improved models that are shown to work well with the given optimal PID design. T. Hägglund contributed with support and experience from PID design. The paper was written by O. Garpinger with assistance from T. Hägglund.

Paper IV

Garpinger, O. and T. Hägglund (2015). "Software-Based Optimal PID Design with Robustness and Noise Sensitivity Constraints". Submitted to *Journal of Process Control*.

This paper presents a new optimal PID design method that takes performance, robustness and control signal noise sensitivity into account to find both the three PID parameters and a low-pass filter time constant. The tuning method uses a Matlab-based software tool and results in a set of optimal or near-optimal PID, PI and I controllers, which the user can switch between to select the best controller giving a maximum allowed control signal activity. No noise modeling is needed in the final design procedure. A large batch of process models representative for the process industry is used to compare optimal PI and PID controllers with

the same noise sensitivity and robustness measures. This shows which processes have the most to benefit from derivative action.

O. Garpinger has plotted the performance, robustness and noise sensitivity trade-offs that illustrate the relationship between optimal PID, PI and I controllers. O. Garpinger has developed the proposed PID design procedure with support from T. Hägglund. The method for showing the benefit of the derivative part has also been developed by O. Garpinger. The paper was written by O. Garpinger with support from T. Hägglund.

Paper V

Cederqvist, L., O. Garpinger, T. Hägglund, and A. Robertsson (2012). "Cascade control of the friction stir welding process to seal canisters for spent nuclear fuel". *Control Engineering Practice* **20**:1, pp. 35 –48.

The Swedish Nuclear Fuel and Waste Management Company plans to join at least 12,000 lids and bases to the extruded copper tubes containing Sweden's nuclear waste, using friction stir welding. To ensure high quality welds without defects or tool fractures, it is important to control the welding temperature. The process is exposed to both quick torque disturbances and power losses due to changing thermal boundary conditions. A cascade controller with two PI controllers for control of power input and temperature is proposed and applied to the custom-built machine. It is shown that the cascade controller manages very well to keep the temperature within the specified process window of $790 - 910°C$.

The PID design method used in this paper is older than the one presented in Paper IV. The paper has thus been extended with a supplement to show the advantages of the new method when applied to the FSW process.

The experimental set-up and all welds have been carried out by L. Cederqvist. The high-level controller objectives were specified by L. Cederqvist. O. Garpinger developed the cascade control structure with support from L. Cederqvist. Process modeling, controller selection and design was carried out by O. Garpinger. Control strategies during the start-up and parking sequences were developed by L. Cederqvist and O. Garpinger. T. Hägglund and A. Robertsson have contributed with their experience through numerous discussions. The paper has mostly been written by O. Garpinger with support from L. Cederqvist, T. Hägglund and A. Robertsson. The paper supplement is work by O. Garpinger.

Additional peer-reviewed publications

Below is a list of additional peer-reviewed publications by the thesis author that were decided not to be included in the thesis.

Cederqvist, L., O. Garpinger, T. Hägglund, and A. Robertsson (2010). “Cascaded Control of Power Input and Welding Temperature During Sealing of Spent Nuclear Fuel Canisters”. In: *Proc. ASME Dynamic Systems and Control Conference*. Cambridge, Massachusetts.

Garpinger, O., T. Hägglund, and K. J. Åström (2012a). “Criteria and Trade-offs in PID Design”. In: *IFAC Conference on Advances in PID Control*. Brescia, Italy.

Garpinger, O., T. Hägglund, and L. Cederqvist (2012b). “Software for PID Design: Benefits and Pitfalls”. In: *IFAC Conference on Advances in PID Control*. Brescia, Italy.

Nielsen, I., O. Garpinger, and L. Cederqvist (2013). “Simulation based Evaluation of a Nonlinear Model Predictive Controller for Friction Stir Welding of Nuclear Waste Canisters”. In: *2013 European Control Conference*. Zürich, Switzerland.

1

Introduction

1.1 The closed-loop system

The single-input single-output feedback loop in Fig. 1.1 will be used throughout this thesis both for analysis and to set up a constrained optimization problem for the design of proportional integral derivative (PID) controllers. The process, $P(s)$, is manipulated by a controller, $C(s)$, such that the controlled variable, z, is kept as close as possible to a set-point, r, in order to minimize the control error, e. The process is affected by a load disturbance, d, at the process input. The measurements of the controlled variable, y, typically contain noise, n, and are fed through a low-pass filter, $F(s)$, to keep the noise level of the control signal, u, low.

The choice of letting the load disturbance act on the process input is supported by e.g. [Shinskey, 1996, p 5], that claims that this is the usual case in process industrial control. The case of having a process output disturbance is, however, briefly investigated in Paper I.

Assuming regulatory control around a constant set-point, $r = 0$, the

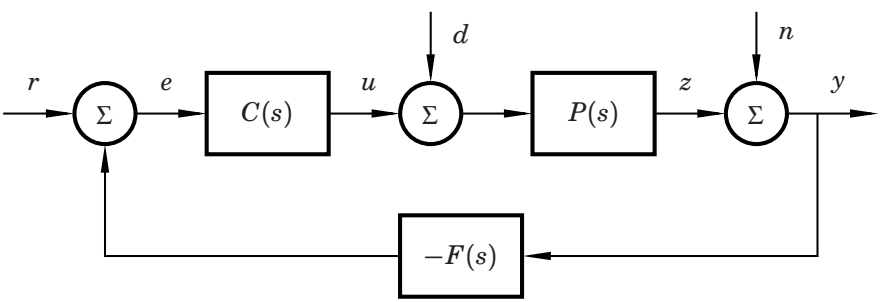

Figure 1.1 A load disturbance, d, measurement noise, n, and set-point, r, act on the closed-loop system with process $P(s)$, controller $C(s)$ and measurement filter $F(s)$.

closed-loop system can be described by three equations

$$Z(s) = \frac{P(s)}{1+P(s)C(s)F(s)}D(s) - \frac{P(s)C(s)F(s)}{1+P(s)C(s)F(s)}N(s), \quad (1.1)$$

$$Y(s) = \frac{P(s)}{1+P(s)C(s)F(s)}D(s) + \frac{1}{1+P(s)C(s)F(s)}N(s), \quad (1.2)$$

$$U(s) = -\frac{P(s)C(s)F(s)}{1+P(s)C(s)F(s)}D(s) - \frac{C(s)F(s)}{1+P(s)C(s)F(s)}N(s), \quad (1.3)$$

with frequency-domain signals in capital letters. It is well-known that the four transfer functions

$$S(s) = \frac{1}{1+P(s)C(s)F(s)}, \quad (1.4)$$

$$T(s) = \frac{P(s)C(s)F(s)}{1+P(s)C(s)F(s)}, \quad (1.5)$$

$$S_p(s) = \frac{P(s)}{1+P(s)C(s)F(s)}, \quad (1.6)$$

$$S_c(s) = \frac{C(s)F(s)}{1+P(s)C(s)F(s)}, \quad (1.7)$$

are sufficient to describe the closed-loop system in Fig. 1.1. $S(s)$ is called the sensitivity function and $T(s)$ the complementary sensitivity function.

1.2 The PID controller

The main objectives of this thesis are to suggest methods for both design and analysis of the PID controller. The three PID controller parts will be described in this section, together with some possible PID controller forms and measurement filters.

Proportional part

The proportional part (P-part) of the control signal is proportional to the control error,

$$u_p(t) = Ke_p(t) + u_0, \quad e_p(t) = br(t) - y(t), \quad (1.8)$$

such that it reacts to present deviations from the set-point. The proportional gain, K, is the parameter normally associated with the P-part, but it is sometimes replaced by a parameter called the proportional band, see

e.g. [Shinskey, 1996]. A P controller alone cannot guarantee zero static control errors, since the control signal becomes zero for $e_p(t) = 0$. The bias term u_0 is used to reduce this effect in controllers that lack an integral part. The magnitude of the static error depends on K. The speed and noise sensitivity of the closed-loop system will typically increase with an increasing K at the same time as the robustness decreases. The P-part is sensitive to noise since K is multiplied directly with the measurements, $y(t)$, unless filtered first. Abrupt changes in the set-point can be smoothed out in the control signal by choosing a set-point weight $0 \leq b < 1$.

Integral part

The integral part (I-part) integrates past values of the control error,

$$u_i(t) = \frac{K}{T_i} \int_0^t e(\tau) d\tau = k_i \int_0^t e(\tau) d\tau, \quad e(t) = r(t) - y(t), \tag{1.9}$$

and will thus remove static control errors due to step load disturbances and set-point changes. It introduces the integral time T_i, but also depends on the proportional gain K unless the fraction K/T_i is replaced by an independent parameter, k_i, called the integral gain. Reducing T_i normally leads to a faster, although less robust, closed-loop system. The summation of past control errors makes the I-part insensitive to noise. A drawback of the I-part is that the controller implementation needs to handle so-called integrator wind-up, see e.g. [Åström and Hägglund, 2005, pp 76–77] for more information.

Derivative part

The derivative part (D-part) of the PID controller,

$$u_d(t) = K T_d \frac{de_d(t)}{dt} = k_d \frac{de_d(t)}{dt}, \quad e_d(t) = cr(t) - y(t), \tag{1.10}$$

predicts future behavior of the controlled variable. It introduces the derivative time T_d, but also depends on K unless KT_d is replaced by the derivative gain k_d. Closed-loop robustness will typically increase with an increasing T_d at the same time as the performance decreases. This is a consequence of the damping properties of the D-part. The system as a whole can still obtain better performance, since the proportional and integral gains can be increased to balance robustness, see e.g. Paper I. A major disadvantage of the D-part is that the differentiation of the control error makes it very noise sensitive. It is thus important to use a low-pass filter together with the D-part. The set-point weight c is normally set to zero to avoid large transients in the control signal.

Controller forms and measurement filters

There are several possible controller combinations that can be formed with the three parts (1.8–1.10). The most common ones are P, I, PI, PD and PID control. I, PI and PID controllers will be considered in this thesis, since the majority of process industrial control applications benefit from the I-part.

In the remaining part of this thesis all controllers will be described in the frequency domain rather than in the time domain. The I controller,

$$C_I(s) = \frac{k_i}{s}, \tag{1.11}$$

has only one parameter, k_i. The PI controller is given by

$$C_{PI}(s) = K\left(1 + \frac{1}{sT_i}\right). \tag{1.12}$$

For PID control there are two common forms: the parallel form,

$$C_{PID}(s) = K\left(1 + \frac{1}{sT_i} + sT_d\right), \tag{1.13}$$

which just adds the D-part to the PI controller, and the series form,

$$C'_{PID}(s) = K'\frac{(1 + sT'_i)\,(1 + sT'_d)}{sT'_i}, \tag{1.14}$$

which is convenient for design based on lead-lag compensation, see e.g. [Franklin et al., 2010]. For $T_i \geq 4T_d$, the parallel and series forms are equivalent. The parallel form is, however, more general since it can have complex zeros. [Hägglund and Åström, 2004] among others have previously shown that this is often preferred. For this reason, the parallel form will be the main focus in this thesis, while the series form will only be used for comparison of different PID tuning methods.

The low-pass filter is an important component of the PID controller since the derivative part is very noise sensitive. There are several ways in which filtering can be implemented together with a PID controller, but it is common practice to use filters of order one either on the derivative part alone,

$$C_{PIDF}(s) = K\left(1 + \frac{1}{sT_i} + \frac{sT_d}{s(T_d/N) + 1}\right), \tag{1.15}$$

where N is usually a number between 5 and 10, or on the whole measurement signal,

$$F(s) = \frac{1}{sT_f + 1}. \tag{1.16}$$

See e.g. [Isaksson and Graebe, 2002; Kristiansson and Lennartson, 2002; Šekara and Mataušek, 2009; Sadeghpour et al., 2012] for studies on these filter forms. An advantage with the measurement signal filter is that one can design controllers for the combination $P(s)F(s)$. This is the approach used in this thesis, but a second-order filter

$$F_{PID}(s) = \frac{1}{(sT_f)^2/2 + sT_f + 1}, \tag{1.17}$$

has been used with PID control in order to guarantee amplitude roll-off for high frequencies. F_{PID} has two complex poles with relative damping $\zeta = 1/\sqrt{2}$, which is the smallest damping ratio for which there is no amplitude increase caused by the filter. Some other studies [Larsson and Hägglund, 2011; Romero Segovia et al., 2013; Micić and Mataušek, 2014] have also explored higher-order low-pass filters for PID control. A first-order filter

$$F_{PI}(s) = \frac{1}{sT_f + 1}, \tag{1.18}$$

will be used here for PI control, also for the sake of high-frequency roll-off. The filter time constant T_f is the only parameter that needs to be set in both F_{PI} and F_{PID}. They would have been more general with two or more parameters, but these forms were chosen to keep the amount of controller parameters low. [Larsson and Hägglund, 2011] showed that the filters F_{PI} and F_{PID} are well suited for the closed-loop system in Fig. 1.1 and that filters of lower and higher order are not likely to give any performance benefits for equivalent noise sensitivity when using white Gaussian noise. I control has natural roll-off, so $F_I(s) = 1$.

1.3 Process industrial context

Even though the research presented in this thesis can be used on a wide range of processes (see e.g. Paper V), its main target is regulatory control in the process industry. This section will, therefore, provide some background on the process industrial control situation and show why the PID controller fits so well into its framework.

According to e.g. [Yamamoto and Hashimoto, 1991; Bialkowski, 1993; Ender, 2001] the two main objectives of process control are to ensure a safe and stable process around the set-point and to minimize the variation of the control error. This has to be accomplished in spite of constantly changing environment, equipment and raw materials [Forsman, 2005]. Disturbances are typically not accessible before they have already influenced the process under control.

Large process industries typically have somewhere between five hundred and five thousand control loops [Yamamoto and Hashimoto, 1991; Bialkowski, 1993; Desborough and Miller, 2002]. The PID control algorithm is used to control almost all of these loops with relative numbers normally as high as 90–97% [Yamamoto and Hashimoto, 1991; Bialkowski, 1993; Ender, 1993; Desborough and Miller, 2002; Kano and Ogawa, 2010; Kuzu, 2012]. This predominance is largely a result of the following PID controller properties:

1. Easy to understand [Desborough and Miller, 2002; Isaksson, 2012]
2. Works well for a majority of processes [Desborough and Miller, 2002; Isaksson, 2012]
3. Pre-programmed in all control systems [Desborough and Miller, 2002]
4. Easy to tune (PI control) [Isaksson, 2012]
5. Tradition

Furthermore, a recent study [Piechottka and Hagenmeyer, 2014] points out that two major obstacles for a more frequent use of advanced process control are the lack of qualified personnel and a quantification of its benefits.

Several surveys of the current process control status points out that only 20–30% of all controllers operate satisfactorily [Bialkowski, 1993; Ender, 1993; Desborough and Miller, 2002]. Moreover, 30% of the loops run in manual mode [Ender, 1993; Desborough and Miller, 2002], while another 30% of the loops increase variability over manual control [Bialkowski, 1993; Ender, 1993]. [Ender, 2001] also points out that

> *"Findings indicate the typical regulatory control system contributes to as much as 50% of the non-uniformity of the final product."*

While all of these studies and reports are more than 10 years old, there are few, if any, recent reports suggesting that the situation would have changed dramatically since then.

There are several reasons why process industrial control loops are not performing satisfactorily. For example:

1. The sheer number of controllers, i.e. lack of tuning time [Kuzu, 2012]
2. The low level of control knowledge in industry [Bialkowski, 1993]

3. Equipment problems, like control valve stiction [Desborough and Miller, 2002]

Manual model-free tuning of controllers is still the most commonly used PID design method in industry. More experienced control engineers tend to rather use simple modeling methods, like bump tests, and model-based tuning rules to determine the controller parameters. Two of the most commonly used PID tuning rules in the process industry are the Internal model control (IMC) and Lambda methods [Ang et al., 2005; Kuzu, 2012], see Section 4.1. Some key aspects behind the success of these methods are that they are simple, fast and intuitive. Important properties given the first two reasons for poor control tuning stated above.

Note that almost all PID controllers in the process industry have the D-part turned off, so that they are actually PI controllers [Bialkowski, 1993]. However, process control experts like Shinskey, Isaksson and Graebe, and Gonzales [Shinskey, 1996; Isaksson and Graebe, 2002; Gonzales, 2012] agree that the derivative part can add considerable value in many control applications. [Ender, 2001; Isaksson and Graebe, 2002; Isaksson, 2012] list several reasons why the D-part is still seldom used:

- PI control is often sufficient.
- The many ways in which the PID controller can be implemented must be matched with the parameters given from the PID design method.
- The D-part can lead to excessive control signal activity, i.e. high noise sensitivity.
- There is a lack of simple four-parameter controller design methods for both the noise filter and PID parameters.

Excessive control signal activity may lead to wear and tear of control valves or other final control elements [Buckbee, 2002; Seborg et al., 2004; Jelali and Huang, 2010]. In Paper III it is also pointed out that:

- High performing PID control requires better models than PI control.

Note that even the simplest dynamic process models are rarely available for most process industrial control loops [Desborough and Miller, 2002]. In conclusion, it is not so surprising that the D-part is not used more often. On the other hand, almost all academic research in the field of PID tuning includes the D-part and this thesis is no exception. For academic work to be accepted in the process industry, it is thus important to consider all above mentioned reasons for preferring PI control over PID.

Advanced process control methods like model predictive control (MPC) is typically applied on a higher hierarchical plant level than PID controllers, such that its control signals become the set-points of the PID loops [Ender, 2001; Desborough and Miller, 2002; Kuzu, 2012]. Better performing PID controllers are therefore important also from a plant-wide optimization point of view.

In summary, there are still many opportunities and reasons for improving the performance of process industrial PID control loops. It is safe to say that the most frequently used PID tuning rules are mainly focused on the robustness of the closed-loop system. This thesis will show that design methods that can handle performance, robustness and noise sensitivity all at the same time have an advantage over methods based on tuning rules.

2

Models and modeling for the process industry

Short modeling time is just as important to the process industry as fast controller design and the goal is to find simple models that provide just enough process information for a given tuning method. This is probably the reason why more advanced system identification methods are typically not used in the industry.

2.1 Process models and process classification

First-order time-delayed (FOTD) models

$$P_m(s) = \frac{K_p}{sT+1}e^{-sL}, \tag{2.1}$$

are commonly used for controller design in the process industry. K_p gives the static gain relation between the process input and output. The speed of the process is captured by the time constant T, where a low value indicates a fast process response. Lastly, L contains the time delay between an input change and the corresponding output reaction.

Second-order time-delayed (SOTD) process models without zeros

$$P_m(s) = \frac{K_p}{(sT_1+1)(sT_2+1)}e^{-sL}, \tag{2.2}$$

will also be considered in this thesis for the purpose of controller tuning. This model holds two time constants, T_1 and T_2, that describe the shape of the process response to input changes. The special case $T_1 = T_2$ is also investigated in this thesis. In Paper III it is shown that the two classes of models (2.1–2.2) are sufficient to keep closed-loop robustness

and performance close to optimal PID design on a large batch of processes representative for the process industry.

Processes can be classified based on the normalized time delay τ, see e.g. [Åström and Hägglund, 2005]. For FOTD models,

$$\tau = \frac{L}{L+T}, \tag{2.3}$$

and for SOTD models,

$$\tau = \frac{L}{L+T_1+T_2}, \tag{2.4}$$

such that its value ranges from 0 to 1. In this thesis, process dynamics will be defined as lag-dominated if $0 \leq \tau \leq 0.2$, balanced if $0.2 < \tau < 0.7$, or delay-dominated if $0.7 \leq \tau \leq 1.0$.

2.2 Modeling methods

Many tuning rules are based on the FOTD model (2.1) and typically have two ingredients: a method to determine the model parameters and a method to determine controller parameters from the model parameters. Both ingredients influence the behavior of the closed-loop system and it is therefore important to be aware of both aspects. A common way to determine K_p, T and L is based on an open-loop step response of the process. K_p is the steady state gain. The apparent time delay L is the t-coordinate of the intersection of the steepest tangent with the time axis, and $L+T$ is the time when the step response has reached 63% of its steady state value. This method is called the 63%-rule in this thesis.

The SIMC tuning rules (see Section 4.1) are instead based on the reduction of an accurate process model [Skogestad, 2003]. K_p is the steady state gain. The apparent time constant T is the largest time constant of the process plus half of the greatest neglected time constant. The apparent time delay L is the sum of the true time delay, half of the largest neglected time constant and all other time constants. The rule is called the half-rule because the biggest neglected time constant is distributed evenly to the apparent delay and the apparent lag. A nice feature of the half-rule is that it can also be used to obtain SOTD models (2.2). A drawback is that it requires an accurate process model.

A relay test is made in closed loop where the control signal switches amplitude whenever the process output crosses a certain hysteresis threshold. This method is less sensitive to disturbances than the step test and keeps the process closer to its set-point during the modeling experiment. The relay typically only gives information about one frequency

point in the process spectrum and it is seldom used to find transfer function models. There are, however, modifications of the relay experiment that give more process information, see e.g. [Berner et al., 2014].

Other possible modeling methods, using both step response and relay tests are collected in a recent review article [Liu et al., 2013]. These methods have, however, not been evaluated in this thesis.

2.3 Benchmark models for the process industry

A test batch of 134 benchmark process models, representative for the process industry, was introduced in [Hägglund and Åström, 2004] and can be found in Fig. 2.1. This batch will be used in this thesis to analyze methods for both modeling and controller design. The benchmark models will be approximated with FOTD models (2.1) in order to classify them based on the normalized time delay (2.3).

A storage into which mass or energy flow in and out is often called a capacity, see e.g. [Shinskey, 1996, p 22]. A capacity can be represented by one or several first-order lags

$$\frac{K_p}{sT+1}, \tag{2.5}$$

if the flows depend on the level of mass or energy in the capacity. The heat flow through an insulating material can e.g. be described in this way. If the flows instead are independent of the capacity level, energy or mass will accumulate and the capacity can then be described by an integrator

$$\frac{K_v}{s}, \tag{2.6}$$

where K_v gives a measure of the accumulation speed. A buffer tank is often modeled like this. A process that involves transportation of material, like e.g. a conveyor belt or a static mixer, will include a dead time L

$$e^{-sL}. \tag{2.7}$$

Processes that contain several of the elements in Eqs. (2.5–2.7) can be modeled with P_{1-7} in Fig. 2.1. Processes with temperature control often include heat transport through several materials and thus become second or higher-order lags. Two or more similar tanks in a row, with liquid flow between them, result in processes like P_2 and P_4. P_4 can for example model level control in a distillation tank as pointed out in [Shinskey, 1996, p 50]. The processes in P_5 model high-order processes with several different time constants, e.g. a mix of process, sensor and actuator dynamics. The

$$P_1(s) = \frac{e^{-s}}{1+sT},$$
$$T = 0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 1,$$
$$1.3, 1.5, 2, 4, 6, 8, 10, 20, 50, 100, 200, 500, 1000$$

$$P_2(s) = \frac{e^{-s}}{(1+sT)^2},$$
$$T = 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 1,$$
$$1.3, 1.5, 2, 4, 6, 8, 10, 20, 50, 100, 200, 500$$

$$P_3(s) = \frac{1}{(s+1)(1+sT)^2},$$
$$T = 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 2, 5, 10$$

$$P_4(s) = \frac{1}{(s+1)^n},$$
$$n = 3,\ 4,\ 5,\ 6,\ 7,\ 8$$

$$P_5(s) = \frac{1}{(1+s)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)},$$
$$\alpha = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$$

$$P_6(s) = \frac{1}{s(1+sT_1)} e^{-sL_1},$$
$$L_1 = 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 0.9, 1.0, \quad T_1 + L_1 = 1$$

$$P_7(s) = \frac{1}{(1+sT)(1+sT_1)} e^{-sL_1}, \qquad T_1 + L_1 = 1,$$
$$T = 1, 2, 5, 10 \qquad L_1 = 0.01, 0.02, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0$$

$$P_8(s) = \frac{1-\alpha s}{(s+1)^3},$$
$$\alpha = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1$$

$$P_9(s) = \frac{1}{(s+1)((sT)^2 + 1.4sT + 1)},$$
$$T = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0$$

Figure 2.1 Test batch of 134 processes representative for the process industry.

integrating processes in P_6 could e.g. model the concentration in a batch with sensor lag and transportation delay. Right half plane zeros, like those in P_8, could be the result of e.g. boiler water level control or concentration control in a series of two continuous stirred-tank reactors [Marlin, 1995; Desborough and Miller, 2002]. A mechanical process or a recirculation loop are examples of processes with complex poles, like P_9.

According to [Shinskey, 1996, p 99] many processes have static gains in the range $K_p = 1 - 10$. It is reasonable to believe, however, that K_v could be a much smaller number e.g. for large tanks. In this thesis both K_p and K_v have been set to 1 for the batch. It seems reasonable to believe that these two gains will vary randomly from process to process, i.e. similar to the characteristics of the measurement noise. This is relevant information since the noise sensitivity measure used in this thesis depends on these gains, see Section 3.3.

3

Criteria and trade-offs for PID design

A rational way to design a controller is to derive a process model and a collection of requirements. Constrained optimization can then be applied to make a trade between often conflicting requirements. Tuning of PID controllers for the process industry is seldom done this way since the effort that can be devoted to a single loop is severely limited.

Requirements typically include specifications on load disturbance attenuation, robustness to process uncertainty, noise sensitivity and set-point tracking. Load disturbance attenuation is a primary concern in process control where steady-state regulation is a key issue, see [Shinskey, 1996], while set-point tracking is a major concern in motion control. Set-point tracking can, however, be treated separately by using a control architecture having two degrees of freedom, see e.g. [Åström and Hägglund, 2005] for more information. Therefore, the focus of this thesis will be on load disturbance attenuation.

3.1 Performance

Load disturbance attenuation will be characterized here by the integrated absolute error (IAE)

$$\text{IAE} = \int_0^\infty |e(t)|dt, \tag{3.1}$$

for a unit step load disturbance on the process input. The IAE-value is thus equal to the total area under the response, as shown in Fig. 3.1. [Shinskey, 1996, p 17] points out that the IAE is a valuable performance measure since it can be related to the economic cost of adding either too

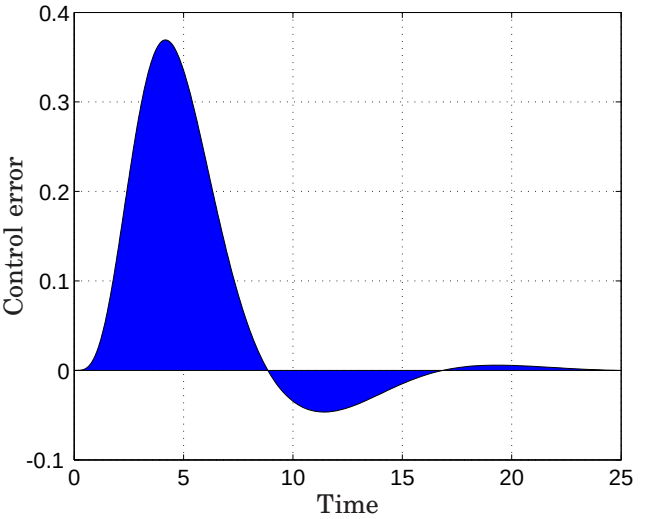

Figure 3.1 The IAE-value is equal to the shaded area of the load disturbance response.

much or too little of a valuable ingredient. Other integral criteria for control performance such as the integrated error (IE)

$$\text{IE} = \int_0^\infty e(t)dt, \tag{3.2}$$

the integrated square error (ISE)

$$\text{ISE} = \int_0^\infty e^2(t)dt, \tag{3.3}$$

and the integrated time-weighted absolute error (ITAE)

$$\text{ITAE} = \int_0^\infty t|e(t)|dt, \tag{3.4}$$

have also been suggested by e.g. [McMillan, 1983; Marlin, 1995].

When performance is considered, it is important to remember that not all control loops need optimal performance. The process in Paper V is a good example of this since the acceptable range for the temperature control is rather wide.

3.2 Robustness

As stated in Section 1.3, it is very important to ensure that all plant processes are safe and stable around the set-point. For this reason, the primary goal of many PID design methods is to secure the robustness of the closed-loop system, i.e. to keep good margins to the point of instability.

The phase margin and the gain margin are classical robustness measures that are still used today [Sanchis et al., 2010; Romero et al., 2011]. See [Åström and Hägglund, 2005, pp 104–105] for definitions.

According to e.g. [Åström and Hägglund, 2005, pp 112–116] and [Zhou and Doyle, 1998, pp 142–143] robustness can be captured by the sensitivity function, $S(s)$, and the complementary sensitivity function, $T(s)$, see Eqs. (1.4–1.5). The maximum values of these functions

$$|S(i\omega)| \leq M_s, \quad |T(i\omega)| \leq M_t, \quad \forall \omega \in \mathbb{R}^+, \tag{3.5}$$

i.e. the H_∞-norms, will be used in this thesis as robustness constraints on the closed-loop system. Note that ω is the frequency in [rad/s] and $\mathbb{R}^+$ denotes the set of all non-negative real numbers. As shown in [Åström and Hägglund, 2005, pp 116–117], this corresponds to the open-loop Nyquist curve of $P(i\omega)C(i\omega)F(i\omega)$ not entering the M_s- or M_t-circles shown in Fig. 3.2. The circles expand with decreasing values of M_s and M_t, resulting in greater closed-loop robustness. Values of M_s and M_t ranging between 1.2 and 2.0 are said to give reasonable robustness [Åström and Hägglund, 2005, p 127] and correspond to gain margins between 6 and 2, as well as phase margins between 49° and 29°. The measure

$$M_{st} = \max(|S(i\omega)|, |T(i\omega)|), \quad \forall \omega \in \mathbb{R}^+ \tag{3.6}$$

will also be used here to quantify the robustness of the closed-loop system given some controller $C(s)$.

The so called M-circle is the smallest circle that encircles the M_s and M_t-circles and it is sometimes used as an alternative robustness criteria, see e.g. [Hägglund and Åström, 2004]. A few other robustness measures can be found in e.g. [Alfaro, 2007; Larsson and Hägglund, 2009; Hansen, 2012].

3.3 Noise sensitivity

Large control signal activity generated by measurement noise could cause undesirable actuator wear and tear. The impact of the transfer function from measurement noise to control action, $S_c(s)$, depends on many factors, with the controller parameters and low-pass filter being particularly

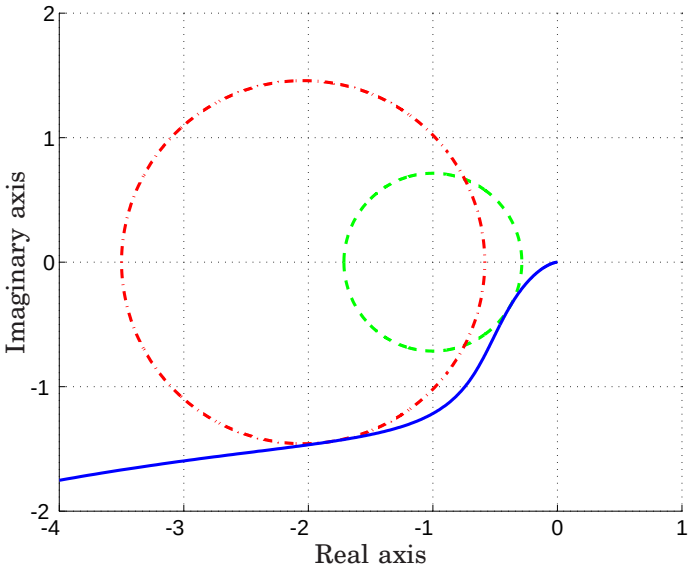

Figure 3.2 The robustness constraints in (3.5) are fulfilled if the open-loop Nyquist curve (solid) does not enter the M_s-circle (dashed) or the M_t-circle (dash-dotted).

important. As with both performance and robustness, there are several possible measures of the closed-loop noise sensitivity. [Garpinger, 2009] and Paper V use the ratio between control signal and measurement noise variance

$$\frac{\sigma_u^2}{\sigma_n^2} \leq V_k, \tag{3.7}$$

to constrain the impact of the measurement noise. σ_u denotes the standard deviation of the control signal due to noise and σ_n is the standard deviation of the noise itself. A drawback with this measure is that one needs information about the noise profile, e.g. the spectral density, in order to calculate it. For this reason, the more recent Paper IV instead uses the H_2 norm of $S_c(s)$

$$\|S_c(s)\|_2 \leq \kappa_u, \tag{3.8}$$

to constrain the noise sensitivity. This condition is also used by e.g. [Larsson and Hägglund, 2011]. Assuming continuous-time white Gaussian measurement noise with unit spectral density, $\|S_c(s)\|_2$ can be derived using

the integral formula

$$\|S_c(s)\|_2 = \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} |S_c(i\omega)|^2 d\omega}. \tag{3.9}$$

This equation will mainly be used for the purpose of controller analysis in this thesis. For control of real processes where the noise characteristics is typically different, it makes more sense to use σ_u as the measure of noise sensitivity. However, one of the biggest advantages with the PID design procedure proposed in Paper IV, is that there is no need to derive σ_u explicitly for real processes.

Two other options would be to use either the H_∞-constraint

$$\|S_c(s)\|_\infty \leq c_u, \tag{3.10}$$

see e.g. [Kristiansson and Lennartson, 2006; Micić and Mataušek, 2014], or the total variation (TV) of the control signal

$$\text{TV} = \sum_{i=1}^{\infty} |u_{i+1} - u_i|, \tag{3.11}$$

where $|u_{i+1} - u_i|$ denotes the control signal change between two consecutive samples, see e.g. [Skogestad, 2003].

3.4 Constrained optimization of PID controllers

PID control studies seldom treat more than one or two tuning criteria for controller design. Taking all three criteria of performance, robustness and noise sensitivity into consideration, one can formulate several different constrained optimization problems. The one that will be the main focus of this thesis is

$$\begin{aligned} \underset{K,T_i,T_d,T_f \in \mathbb{R}^+}{\text{minimize}} \quad & \int_0^{\infty} |e(t)| dt, \\ \text{subject to} \quad & |S(i\omega)| \leq M_s, \\ & |T(i\omega)| \leq M_t, \quad \forall \omega \in \mathbb{R}^+, \\ & \|S_c(s)\|_2 \leq \kappa_u, \end{aligned} \tag{3.12}$$

with the cost function and constraints as defined in Eqs. (3.1), (3.5) and (3.8). This optimization problem can be used to design any of the controllers $C(s)$ given in Eqs. (1.11–1.13) as well as the filters in Eqs. (1.17–1.18). However, the non-convexity of this optimization problem makes it difficult to solve directly.

4

PID design methods

In this chapter we distinguish between PID design based on tuning rules and optimization. Tuning rules are a set of formulas from which one can determine the controller parameters, and they typically depend on the parameters of some specific process model, e.g. FOTD or SOTD. In optimization-based methods, the controller parameters are instead derived from the solution to an optimization problem like (3.12), also given some model of the process. The aim of tuning rules is thus to find universal relations between model and controller parameters, while optimization-based design treats each process model individually.

4.1 Tuning rules

A good PID tuning method should both be fast and easy to carry out for the large number of control loops in a factory. This has led to the great popularity of formula-based tuning rules. A measure of this popularity is given by the book [O'Dwyer, 2009], in which there are 1,730 PI and PID tuning rules collected. Some of the most commonly used tuning rules are compared in this thesis (see Papers I and III), both with respect to each other and optimal PID design. Another example of a study that compares several different tuning rules is [Lin et al., 2008].

The Ziegler-Nichols methods

During the second world war, [Ziegler and Nichols, 1942] presented two methods for design of P, PI and PID controllers that have received considerable attention. The step response method uses only two parameters, K_v and L. These are determined from a step response where K_v is the steepest slope and L is the intersection of the steepest tangent with the time axis. L is therefore the same as in the 63%-rule, see Section 2.2. The

step response method gives the PI controller parameters

$$K = \frac{0.9}{K_v L}, \quad T_i = 3L. \tag{4.1}$$

The Ziegler-Nichols frequency response method is based on a closed-loop test with a P controller. The proportional gain is increased until the process is on the border to instability. This proportional gain is called the ultimate gain, K_u, and the oscillation period is called the ultimate period, T_u. The frequency response method is tuned to give quarter amplitude damping of load disturbance responses and is given by

$$K = 0.4K_u, \quad T_i = 0.8T_u. \tag{4.2}$$

The Ziegler-Nichols methods are included in this thesis only for historical reasons. Several studies have concluded that the tuning methods are not suited for industrial practice, but they are still quite frequently used. For example, [Bialkowski, 1993] states that

> *"Most trade schools deal with the loop tuning issue by teaching the Ziegler Nichols quarter-amplitude-decay method as the only reasonable fallback position. Unfortunately, for the pulp and paper industry, this method is very oscillatory and is one of the reasons for the destabilization of paper uniformity."*

Internal model control

IMC is based on the configuration in Fig. 4.1 where $P(s)$ is the process, $P_m(s)$ is the process model and $C_{IMC}(s)$ is the internal model controller. The process model is separated into two parts, one minimum phase part $P_-(s)$ and a non-minimum phase part $P_+(s)$ such that $P_m(s) = P_-(s)P_+(s)$. The IMC controller is then defined as $C_{IMC}(s) = P_-^{-1}(s)f(s)$ e.g. with

$$f(s) = \frac{1}{(T_{cl}s + 1)^r}, \tag{4.3}$$

where the integer r is set such that the IMC controller is proper. Assuming that the process model is perfect and minimum phase, i.e. $P(s) = P_m(s) = P_-(s)$, the closed-loop system from $R(s)$ to $Y(s)$ becomes $T(s) = f(s)$. T_{cl} can thus be viewed as a tuning parameter adjusting the speed of the closed-loop system. The IMC configuration in Fig. 4.1 can be transformed to the closed-loop system in Fig. 1.1 by the formula

$$C(s) = \frac{C_{IMC}(s)}{1 - P_m(s)C_{IMC}(s)} = \frac{P_-^{-1}(s)}{f^{-1}(s) - P_+(s)}. \tag{4.4}$$

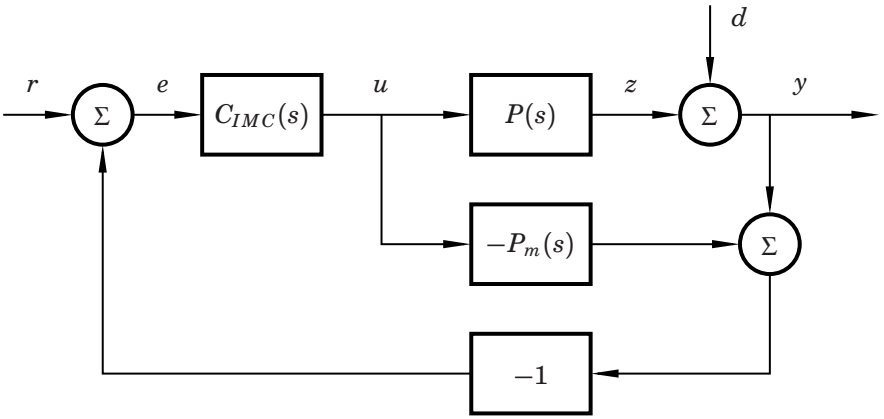

Figure 4.1 The IMC configuration.

[Rivera et al., 1986] points out that most process models used in industrial applications lead to IMC controllers on the PID form. Based on different forms of process models, a large number of IMC PID tuning rules are presented in this paper [Rivera et al., 1986, Table 1]. These formulas are, however, known to give poor performance to e.g. input load disturbances on lag dominated processes since the slow process pole is still present in this response. Some modifications to IMC-based PID are presented in e.g. [Lee et al., 1998; Shamsuzzoha and Lee, 2007; Vilanova, 2008]. In this thesis, however, two other IMC-related methods are used for PID design comparison, namely Lambda tuning and SIMC.

Lambda tuning

Lambda tuning originates from early computer control [Dahlin, 1968; Higham, 1968], was rediscovered in connection with the development of IMC PID [Rivera et al., 1986], and is today widely adopted in the industry, see e.g. [Sell, 1995; Bialkowski, 1996; Forsman, 2005]. Modeling is typically based on step responses, and an FOTD model can be obtained using for example the 63%-rule. The desired closed-loop time constant to a set-point change, T_{cl}, is used as a tuning parameter. This parameter admits a compromise between performance and robustness where low values on T_{cl} give aggressive control, while higher values give smooth control. In the original paper [Dahlin, 1968], $1/T_{cl}$ was called λ, which explains the name of the tuning method. For PI control the tuning formulas are

$$K = \frac{T}{K_p(T_{cl} + L)}, \quad T_i = T, \tag{4.5}$$

and for series form PID they are

$$K' = \frac{T}{K_p(T_{cl} + L)}, \quad T_i' = T, \quad T_d' = \frac{L}{2}, \tag{4.6}$$

given the notations from (1.12), (1.14) and (2.1). These two tuning formulas can be derived directly from rules D and F in [Rivera et al., 1986, Table 1] using either a first-order Taylor expansion (PI) or a first-order Padé approximation (PID) of the time delay. The Lambda tuning rules inherit the problem with sluggish load disturbance responses from IMC tuning for processes with lag-dominated dynamics. There are several suggestions for choosing T_{cl}, some of which can be found in [Sell, 1995; Bialkowski, 1996; Janvier and Bialkowski, 2005]. [Janvier and Bialkowski, 2005] also gives instructions for Lambda tuning modifications of integrating and lag-dominated processes. In this thesis, however, T_{cl} will be chosen proportional to the process time constant T.

Skogestad's methods

[Skogestad, 2003] introduced modifications of the Lambda tuning method called SIMC, that improves performance especially for lag-dominant processes. The FOTD model is obtained by model reduction of a high-order process model using the half-rule. The proportional gain K is chosen as in (4.5) and the integral time as

$$T_i' = \min(T, 4(T_{cl} + L)). \tag{4.7}$$

This choice avoids cancellation of a slow process pole by a controller zero when $T > 4(T_{cl} + L)$. Furthermore, Skogestad recommends PID control mainly for processes with dominant second-order dynamics, which can be modeled using (2.2). In this case, T is replaced by the largest time constant T_1 in (4.5) and (4.7). The derivative time is set to

$$T_d' = T_2. \tag{4.8}$$

Notice that the PID series form is used. Unlike the original Lambda tuning, the desired closed-loop time constant T_{cl} is chosen as a factor of the apparent time delay L. Skogestad recommends $T_{cl} = L$ as the primary choice which typically gives a sensitivity close to $M_s = 1.6$. Less aggressive tuning is obtained for larger values of T_{cl} and can be found through selection of an upper limit on the controller gain, see [Skogestad, 2006].

A modified version of SIMC PI control, here called SIMC+, was introduced in [Skogestad and Grimholt, 2012] to improve performance for delay-dominated systems where the original SIMC rule typically results

in controllers with too low proportional gain. The PI parameters were instead chosen as

$$K = \frac{T + L/3}{K_p(T_{cl} + L)}, \quad T_i = \min(T + L/3, 4(T_{cl} + L)). \tag{4.9}$$

For delay-dominated and balanced processes the integral gain is the same as for SIMC, while the proportional gain K and integral time T_i are larger than for SIMC. Recently [Grimholt and Skogestad, 2013], made yet another extension to the SIMC rules that also designs PID controllers for FOTD processes. This new study is, however, not considered in this thesis.

The AMIGO method

The approximate M constrained integral gain optimization (AMIGO) tuning rules were introduced in [Hägglund and Åström, 2004] and further developed in [Åström and Hägglund, 2005]. The formulas were derived using the MIGO PID design method (see Section 4.2) on the test batch in Fig. 2.1 to find optimal PI and PID controllers. Curve fitting was then employed to find tuning rules for PI and PID control with respect to FOTD parameters derived with the 63%-rule. The AMIGO PI tuning rule is

$$\begin{aligned} K &= \frac{0.15}{K_p} + \left(0.35 - \frac{LT}{(L+T)^2}\right)\frac{T}{K_p L}, \\ T_i &= 0.35L + \frac{13LT^2}{T^2 + 12LT + 7L^2}, \end{aligned} \tag{4.10}$$

and the PID rule for the parallel form is

$$\begin{aligned} K &= \frac{1}{K_p}\left(0.2 + 0.45\frac{T}{L}\right), \\ T_i &= \frac{0.4L + 0.8T}{L + 0.1T}L, \\ T_d &= \frac{0.5LT}{0.3L + T}. \end{aligned} \tag{4.11}$$

Finally, the AMIGO rule for integrating processes is

$$K = \frac{0.45}{K_v L}, \quad T_i = 8L, \quad T_d = 0.5L, \tag{4.12}$$

where K_v is the integral gain. [Padula and Visioli, 2011] presents PID tuning rules that are similar to the AMIGO rules, but are instead based on IAE minimization with respect to maximum sensitivity on FOTD processes alone. This design method can also be used to find fractional-order PID controllers.

Four parameter tuning rules

The number of tuning rules that also take the noise sensitivity problem into consideration is still small. The methods presented in [Kristiansson and Lennartson, 2002; Kristiansson and Lennartson, 2006] are an exception. They use an H_∞-based optimization approach to find several tuning rules for both the PID parameters and the noise filter. The filter time constant is chosen such that the closed-loop performance and robustness are barely affected.

In [Romero Segovia et al., 2014a; Romero Segovia et al., 2014b] second-order measurement filters are designed in combination with three common tuning rules for PID control. The authors relate trade-offs for performance, robustness and noise sensitivity to find tuning rules such that the filter has little influence on the original, unfiltered, control performance and robustness. A similar method was presented in [Leva and Maggio, 2011] where ideal PID parameters, given by an arbitrary design method, are converted into a PID with derivative filter (1.15). The filter is chosen in relation to closed-loop cut-off frequency or high-frequency gain such that the effect on nominal performance and robustness is limited.

4.2 Optimization-based methods

Over the past two decades, optimization-based PID design methods have started to gain attention. There are, however, still rather few studies in comparison to those dealing with tuning rules. Some of the optimization-based methods that are important in relation to the proposed design method are summarized in this section. Some other PID optimization methods of interest are proposed by e.g. [Liu and Daley, 1999; Syrcos and Kookos, 2005; Toscano, 2005; Alfaro and Vilanova, 2013].

MIGO design and similar methods

M constrained integral gain optimization (MIGO) tuning of PI and PID controllers was presented in [Åström et al., 1998; Panagopoulos et al., 2002]. The designed controllers are the result of an IE minimization subject to the M_s robustness constraint given in (3.5). The PID control optimization problem is also complemented by additional constraints on the shape of the loop transfer function $L(s) = P(s)C(s)$ to give the load disturbance response better damping.

The series form PID design method presented in [Sanchis et al., 2010; Romero et al., 2011] also considers IE minimization, but instead with respect to classical amplitude and phase margins. They also impose the limitation that the two controller zeros have to be equivalent. The more recent methods by [Hast et al., 2013; Boyd et al., 2014] present a more

general framework of convex-concave optimization which can be solved very quickly, but it is still based on IE minimization with robustness constraints. [Grimholt and Skogestad, 2015], on the other hand, presents a fast algorithm for minimization of input and output disturbance IAE with respect to the robustness constraints in (3.5). This new method is thus very similar to the next one presented here.

SWORD

A MATLAB®-based software tool for robust PI and PID design is presented in Paper II. This tool extends the one presented in [Nordfeldt and Hägglund, 2006], and solves a modified version of Eq. (3.12) without the noise sensitivity constraint. The optimization problem is

$$\begin{aligned} \underset{K,T_i,T_d\in\mathbb{R}^+}{\text{minimize}} \quad & \int_0^\infty |e(t)|dt, \\ \text{subject to} \quad & |S(i\omega)| \leq M_s, \\ & |T(i\omega)| \leq M_t, \quad \forall\omega \in \mathbb{R}^+, \\ & |S(i\omega^s)| = M_s, \quad \text{and/or} \\ & |T(i\omega^t)| = M_t, \end{aligned} \tag{4.13}$$

where ω^s are frequencies for which the open-loop frequency response is tangent to the M_s-circle and vice versa for ω^t and the M_t-circle. Either ω^s or ω^t could be an empty set, but not simultaneously. As a direct result of these additional equality constraints, at least one of the inequality constraints will be active, thus forcing the solution to have contact with at least one of the M_s- and M_t-circles. This makes the optimization problem easier to solve than the problem with only inequality constraints. The drawback is that there could be a controller with greater robustness that yields better performance. Several studies by [Garpinger, 2009; Grimholt and Skogestad, 2012; Garpinger et al., 2014] do, however, indicate that the solution to the optimization problem (4.13) is, in most cases, optimal also without equality constraints just as long as the robustness is reasonable with M_s and M_t roughly below 1.8. The low-pass filter time constant, T_f, is fixed in this optimization problem such that the controllers are designed for the process and filter combination $P(s)F(s)$. This design method is called SWORD (Software-based optimal robust design) and it needs a stable linear process model as well as specified values of M_s, M_t and T_f to work. A more detailed description of the program is given in Paper II. Also note that the tool can easily be modified to deal with e.g. discrete-time PID control, other performance measures and different low-pass filter configurations. The software tool is freely available at:

http://www.control.lth.se/project/PID

Other similar design methods can be found in [Oviedo et al., 2006; Harmse et al., 2009; Sadeghpour et al., 2012], which present software tools that can be readily used for constrained optimization of the PID parameters. This collection of methods will here be referred to as Software-based optimal design methods. Notice that MATLAB® has a PID optimization tool of its own [MathWorks®, 2015].

Four parameter optimization methods

The optimization-based methods presented so far, except [Panagopoulos et al., 2002], have in common that performance and robustness are their main focus. They exploit constrained optimization, but they do not explore the noise sensitivity problem. Studies by [Šekara and Mataušek, 2009; Larsson and Hägglund, 2011; Micić and Mataušek, 2014] show how constrained optimization can be used to design PID controllers and noise filters of different orders. The methods do, however, only describe how the optimization can be carried out, and they do not present any software tools for solving their respective optimization problems.

The method for design of PID controllers and noise filters proposed in Paper IV is based on the method from [Garpinger, 2009]. The original idea uses the MATLAB® toolbox presented in the previous subsection to solve a constrained optimization problem for the design of robust PID controllers. The noise filter time constant is fixed each time a new controller is designed and the closed-loop noise sensitivity is then constrained through repeated PID design, similar to the method presented in [Panagopoulos et al., 2002]. With this approach there is no need to keep the performance close to the unfiltered, nominal, case. An increase of the filter time constant does not affect the robustness as it would in the earlier mentioned methods by [Romero Segovia et al., 2014a; Romero Segovia et al., 2014b; Kristiansson and Lennartson, 2002; Kristiansson and Lennartson, 2006]. Noise sensitivity is measured by the variance gain from measurement noise to control signal (see Eq. 3.7), and Simulink® simulations are used to calculate this measure by the use of process noise data.

5

Friction stir welding

The author of this thesis has had the opportunity to work on the temperature control of an industrial friction stir welding (FSW) process. While the given machine is not part of any process industry, it is still a good example of a process where PID control is sufficient. One of the measurement signals used in the final cascade controller also includes a lot of noise, which makes the process suitable for analysis of the proposed control design technique. The first controller tuning made for the FSW process is described in Paper V and uses the design method proposed in the author's licentiate thesis [Garpinger, 2009]. A supplement has been added to Paper V in this thesis, to show that the design procedure proposed in Paper IV is better to use.

5.1 Background

FSW is a thermo-mechanical solid-state process that was invented in 1991 at The Welding Institute (TWI) [Thomas et al., 1991]. A rotating non-consumable tool, consisting of a tapered probe and shoulder, is plunged into the weld metal and traversed along the joint line, see Fig. 5.1. Frictional heat is generated between the tool and the weld metal, causing the metal to soften, normally without reaching the melting point, and allowing the tool to traverse the joint line. The three most common input parameters are (see Fig. 5.1)

1. Tool rotation rate
2. Welding speed along the joint
3. Axial force

Temperature control could be an important part of the FSW process if it has non-uniform thermal boundary conditions or if it is used on a material

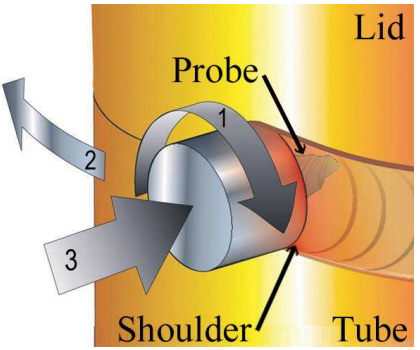

Figure 5.1 Illustration of the friction stir welding process.

with a smaller allowed temperature range, the so-called process window. If the welding temperature gets too high for a longer period of time, there is a risk for probe fracture. Similarly, too low temperatures may result in discontinuities in the weld. A previous study by [Cederqvist et al., 2008] has shown that the rotation rate is the best suited control signal for weld temperature control and is used here.

As of today, most friction stir welds are made on plates with rather uniform thermal boundary conditions. For this reason, there are currently only a small, although growing, amount of FSW applications using temperature control. [Fehrenbacher et al., 2008] have approached closed-loop control of the welding temperature by manipulation of the welding speed and more recently by adjusting the tool rotation rate [Fehrenbacher et al., 2010]. Another study [Mayfield and Sorensen, 2010] suggests a cascaded control strategy similar to that previously presented in [Cederqvist et al., 2009] and in Paper V. Recently, there have been two more studies on temperature control, see [De Backer et al., 2014; Ross, 2014].

The need for reliable control of the welding temperature should increase as FSW is used on metals like titanium and steels. In addition, more complex geometries of welding objects will result in non-uniform thermal boundary conditions and thus also a need for temperature control.

5.2 FSW of thick copper canisters for nuclear waste

The Swedish Nuclear Fuel and Waste Management Company (SKB) plans to join at least 12,000 lids and bases to the extruded copper tubes containing Sweden's nuclear waste, using FSW. The canisters produced (5 m height, 1 m diameter) are a major component of the Swedish system

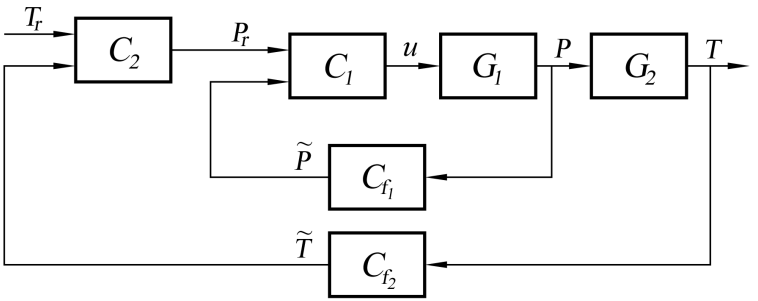

Figure 5.2 The cascaded control structure proposed for the FSW process.

for managing and disposing nuclear waste. They will be stored in the Swedish bedrock and must remain intact for 100,000 years. A corrosion barrier of 5 cm thick copper and a cast iron insert are used to meet this requirement.

The friction stir welding machine currently used to seal the copper canisters measures the welding temperature [°C] and the torque [Nm] required to maintain the tool rotation rate. Another important variable is the power input [kW] which is proportional to the tool rotation rate multiplied with the torque. [Cederqvist et al., 2008] showed that the power input is well correlated to the welding temperature.

Of all process outputs, the welding temperature is the most crucial one to control. It is, therefore, very important to keep the temperature within the process window, which is roughly between 790 and 910°C for FSW on the copper canisters, when measured inside the probe. Several aspects of the welds make the temperature challenging to control. Depending on the position of the tool, a full weld cycle can be divided into five separate sequences each with challenges of different nature. For example, the start-up typically contain a lot of fast and high-magnitude torque disturbances. Other disturbances are caused either by the tool moving in and out of preheated areas or by greater heat conduction at the joint line compared to the lid. Each of these disturbances has to be counteracted to make sure the temperature stays within the process window.

5.3 PID design for an FSW process

A cascaded control strategy seems ideal for the characteristics of this specific FSW application with its fast torque disturbances, and slower temperature counterparts. It was, therefore, decided to use the controller structure displayed in Fig. 5.2. The process has been divided into two subsystems. Process G_1 holds the dynamics from the tool rotation rate,

u (rpm), to the power input, P (kW). Step response modeling resulted in the process model

$$G_1(s) = 0.12 \cdot \frac{4.6^2}{s^2 + 2 \cdot 0.8 \cdot 4.6s + 4.6^2}. \tag{5.1}$$

This system mainly holds the servo characteristics of the spindle motor that drives the probe. The outer process, G_2, on the other hand, describes how the power input is related to the probe temperature, T (°C). This process model,

$$G_2(s) = \frac{11.6}{(7s+1)^2} e^{-5s}, \tag{5.2}$$

was also derived using step response modeling as described in Paper V.

The process was originally controlled by two PI controllers, C_1 and C_2, which were sufficient to keep the temperature within the process window, see Paper V. The design procedure was based on the one proposed in [Garpinger, 2009], but no low-pass filters, C_{f_1} or C_{f_2}, were used in the control solution. Since Paper V was written, the PID design procedure has been updated, see Paper IV. In the Supplement to Paper V it is shown how an inner I controller and outer PID controller, with low-pass filter $C_{f_2}(s)$, can be derived using the updated design procedure in order to improve the temperature control at the joint line.

6

Thesis contributions

6.1 Thesis objectives

In industry, PID control is by far the most common control strategy and more advanced control techniques are often rejected due to lack of time and personnel with the required knowledge. In an effort to present PID tuning methods for the industry, most PID researchers strive to develop tuning rules which can easily be used by practitioners. This has led to several methods that are good at finding robust PI controllers. However, these methods often fail to design controllers with respect to optimal performance as well as noise sensitivity.

The objectives of this thesis are to propose new tools for design and analysis of PID controllers. These should show advantages of tuning based on constrained optimization as well as how and when to use it. This is a standpoint that recognizes both the academic efforts on constrained optimization and the industry in which the PID algorithm is likely to remain for many years to come.

6.2 Contributions

The low order of the PID controller makes it easier to analyze, with respect to the different controller criteria, than more complex control strategies. This property is exploited in Paper I, where new trade-off plots are presented. Performance and robustness level curves are drawn as functions of the PID parameters. This is a tool that can provide tuning insight and intuition for practitioners as well as students. It is also used to show weaknesses and strengths of different PI tuning rules.

The trade-off plots are also used to show the set of optimal controllers for different robustness values. This set can be derived directly by the MATLAB®-based software design method which was briefly mentioned in Section 4.2 (i.e. SWORD). It is thoroughly described in Paper II. As far

as the author of this thesis knows, this is the only freely available PID design tool that minimizes IAE with respect to robustness constraints. The usefulness of the software is demonstrated in Papers III–V, both for controller analysis and PID design on real processes. So far, the program has mainly been used for academic purposes, but similar software-based optimal design methods should be commercially viable in the future. However, as shown in Paper III, it is important to find a modeling method that works well together with the optimal PID design. FOTD models based on the 63%-rule (see Section 2.2) are e.g. not good enough to use together with SWORD neither for PI nor PID control and several common tuning rules share this weakness, at least in the PID case. It is shown that a moderate amount of process knowledge, concentrated around single phase angles, is enough to provide both PI and PID control close to optimal. Knowing that the closed-loop robustness variation can be decreased compared to other tuning methods, it is also pointed out that M_s and M_t can be increased to improve performance even further.

The design method presented in Paper IV shows how SWORD can be used to find optimal or near-optimal PID controllers also with respect to control signal noise sensitivity. The procedure uses the low-pass filter time constant T_f and the integral gain k_i as tuning variables to create a set of PID, PI and I controllers. For example, Bode magnitude plots of a set of 17 PID, 7 PI, and 7 I controllers, derived for the balanced process,

$$P(s) = \frac{1}{(s+1)^4}, \tag{6.1}$$

with $M_s = M_t = 1.4$, are shown in in Fig. 6.1. Such a set of controllers can also be used to slow down the closed-loop system if it is deemed too fast. In Paper IV, performance and noise sensitivity trade-off curves are plotted for different robustness levels and give insight into design with respect to all three criteria. The article also presents a new method for quantifying the benefit of the derivative part, by comparing optimal PI and PID controllers with the same robustness and noise sensitivity constraints. Besides these major contributions, which the thesis author believe are novel, the paper ends with a comparison of tuning rules and software-based optimal design. The main conclusions from this comparison are that even though tuning rules are a good tool for present use in industry, software-based optimal tuning has a greater research potential.

In Paper V, the SWORD method is used to design two PI controllers in a cascade-loop for control of a friction stir welding process. As far as the thesis author knows, this is the first time that a cascade controller has been systematically designed and applied to the temperature control problem of an FSW process. Since the PID design procedure used in Paper V is older than the one presented in Paper IV, a supplement

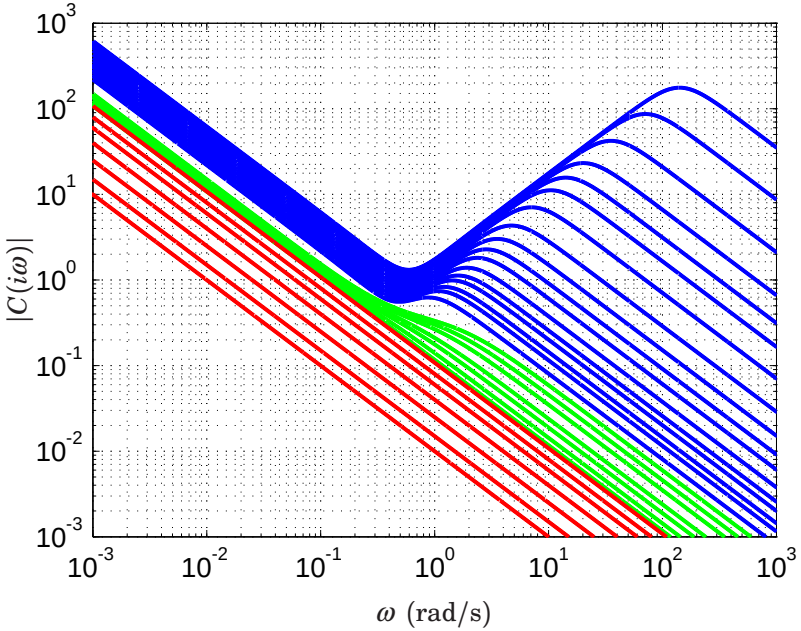

Figure 6.1 Bode magnitude plots of a set of PID (blue), PI (green), and I controllers (red), with $M_s = M_t = 1.4$, derived for the balanced process (6.1).

has been added just after the main paper to show the improvements of the updated method. The results also give some nice insights about the noise content of the control signal showing that PI controllers should be used together with a low-pass filter, at least in this application. Besides the design of PID controllers, Paper V also contributes with strategies for control during the start and end of the welds, a task that presents much greater challenges than the control during the joint line sequence. One should also remember that for this specific FSW process, the choices of cascade control and strategies during start-up are much more crucial for the overall temperature control than the choices of parameters for the individual PID controllers. For the sake of the thesis, however, one could instead view the FSW process as part of e.g. a chemical factory. If all controllers of the factory were optimized like this single process, it would be reasonable to expect synergy effects benefiting the factory as a whole. Unfortunately, the level of such benefits is rather difficult to predict and likely part of the reason why optimal PID control has not already been developed in the industry.

6.3 Visions and future work

In this thesis, it is suggested that software-based optimal PID design methods are the future. The future, however, is not here yet, and the question is what is missing to fulfill the prediction. The most important key still missing is a modeling method that is tailor made for optimal PID control. Some desired properties of such a method have already been suggested. It should ideally be as fast as possible and therefore provide just the process information that is absolutely necessary for the control design. Closed-loop identification experiments, like relay tests, are preferred, since they are less sensitive to disturbances than their open-loop counterparts. A reasonable approach would be to first develop something that works for PI control and then move on to more general alternatives.

The software behind the SWORD method is just one suggestion of how such a tool could work, and some improvements are necessary before the proposed design procedure from Paper IV can be employed in industry. For example, it would be desirable to obtain faster solutions to the optimization problem. That way it would be possible to quickly determine a whole set of PID controllers, both with respect to noise sensitivity and robustness. A possible way to speed up the solver is to simplify the optimization problem even further. What matters is that the software reliably provides PID controllers and low-pass filters that are decently close to optimal, with at least one robustness constraint (e.g. M_s).

Experienced control engineers are probably able to use optimization-based design techniques as stand-alone methods. But in order to attract a broader audience and to speed up controller tuning, they should ideally be part of future autotuners together with the desired modeling method. One suggestion is to let modeling experiments be run by a PLC (Programmable Logic Controller) or DCS (Distributed Control System), while system identification and optimal PID design could be handled by an external unit, like a laptop or tablet computer. The autotuning could then be scheduled to run either periodically by an operator or when changes have been made to the process. This routine should preferably be reliable enough to use without the need of careful supervision.

There should be plenty of possibilities to add new interesting features to future versions of the SWORD software. For example, it would be desirable to add optimization of other PID and filter forms, like the series form (1.14) and the derivative filter (1.15). It should also be possible to change the cost function to e.g. minimize ISE (3.3) or ITAE (3.4), both with respect to input and output disturbances. In industrial applications it is important to have a strategy for set-point changes, and the final software should thus include some options to design the reference weights or a reference filter. For educational purposes, it seems like a good idea

to combine the optimal PID control design software with functions from pedagogic software tools, like the one presented in [Guzman et al., 2008].

Finally, it is important that software-based optimal PID design methods are tested on more industrial applications and that their functions are discussed together with experienced control engineers. This will help facilitate the acceptance of such methods in the industry.

Bibliography

Alfaro, V. M. (2007). "PID controllers' fragility". *ISA transactions* **46**:4, pp. 555–559.

Alfaro, V. M. and R. Vilanova (2013). "Performance and Robustness Considerations for Tuning of Proportional Integral/Proportional Integral Derivative Controllers with Two Input Filters". *Industrial & Engineering Chemistry Research* **52**:51, pp. 18287–18302.

Ang, K. H., G. Chong, and Y Li (2005). "PID Control System Analysis, Design and Technology". *IEEE Transactions on Control Systems Technology* **13**:4, pp. 559–576.

Åström, K. J. and T. Hägglund (2005). *Advanced PID Control*. ISA – The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC.

Åström, K. J., H. Panagopoulos, and T. Hägglund (1998). "Design of PI Controllers based on Non-Convex Optimization". *Automatica* **34**:5, pp. 585–601.

Berner, J., K. J. Åström, and T. Hägglund (2014). "Towards a New Generation of Relay Autotuners". In: *19th IFAC World Congress*. Cape Town, South Africa.

Bialkowski, W. L. (1993). "Dreams versus reality: A view from both sides of the gap". *Pulp & Paper Canada* **94**:11, pp. 19–27.

Bialkowski, W. L. (1996). "Control of the Pulp and Paper Making Process". In: Levine, W. S. (Ed.). *The Control Handbook*. CRC Press, Inc, Boca Raton, Florida. Chap. 72, pp. 1219–1242.

Boyd, S., M. Hast, and K. J. Åström (2014). "MIMO PID Tuning via Iterated LMI Restriction". *Unpublished manuscript*. URL: `https://web.stanford.edu/~boyd/papers/mimo_pid_tuning.html`.

Buckbee, G. (2002). "Poor Controller Tuning Drives Up Valve Costs". *Control Magazine*, pp. 67–77.

Cederqvist, L., C. D. Sorensen, A. P. Reynolds, and T. Öberg (2008). "Improved process stability during friction stir welding of 5 cm thick copper canisters through shoulder geometry and parameter studies". *Science and Technology in Welding and Joining* **46**:2, pp. 178–184.

Cederqvist, L., R. Johansson, A. Robertsson, and G. Bolmsjö (2009). "Faster Temperature Response and Repeatable Power Input to aid Automatic Control of Friction Stir Welded Copper Canisters". In: *Friction Stir Welding and Processing V*. San Francisco, USA.

Dahlin, E. B. (1968). "Designing and tuning digital controllers". *Instruments and Control Systems* **41**:6, pp. 77–83.

De Backer, J., G. Bolmsjö, and A.-K. Christiansson (2014). "Temperature control of robotic friction stir welding using the thermoelectric effect". *The International Journal of Advanced Manufacturing Technology* **70**:1-4, pp. 375–383.

Desborough, L. and R. Miller (2002). "Increasing Customer Value of Industrial Control Performance Monitoring-Honeywell's Experience". In: *AIChE Symposium Series, Number 326 (Volume 98)*, pp. 169–189.

Ender, D. B. (1993). "Process Control Performance: Not as Good as you Think". *Control Engineering* **40**:10, pp. 180–190.

Ender, D. B. (2001). "Regulatory Control is the Foundation for Advanced Process Control". *Control Engineering*.

Fehrenbacher, A., F. E. Pfefferkorn, M. R. Zinn, N. J. Ferrier, and N. A. Duffie (2008). "Closed-loop Control of Temperature in Friction Stir Welding". In: *7th International Friction Stir Welding Symposium*. Awaji Island, Japan.

Fehrenbacher, A., N. A. Duffie, N. J. Ferrier, M. R. Zinn, and F. E. Pfefferkorn (2010). "Temperature measurement and closed-loop control in friction stir welding". In: *8th International Friction Stir Welding Symposium*. Timmendorfer Strand, Germany.

Forsman, K. (2005). *Reglerteknik för processindustrin*. Studentlitteratur, Lund, Sweden.

Franklin, G. F., J. D. Powell, and A. Emami-Naeini (2010). *Feedback Control of Dynamic Systems*. 6th Edition. Pearson, Upper Saddle River, NJ.

Garpinger, O. (2009). *Design of Robust PID Controllers with Constrained Control Signal Activity*. Licentiate Thesis ISRN LUTFD2/TFRT--3245--SE. Department of Automatic Control, Lund University, Sweden.

Garpinger, O., T. Hägglund, and K. J. Åström (2014). "Performance and robustness trade-offs in PID control". *Journal of Process Control* **24**:5, pp. 568–577.

Gonzales, R. (2012). “Future Perspectives of PID Control - Panel Session”. In: *IFAC Conference on Advances in PID Control*. Brescia, Italy.

Grimholt, C. and S. Skogestad (2012). “Optimal PI-Control and Verification of the SIMC Tuning Rule”. In: *IFAC Conference on Advances in PID Control*. Brescia, Italy.

Grimholt, C. and S. Skogestad (2013). “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*. Mumbai, India.

Grimholt, C. and S. Skogestad (2015). “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*. Copenhagen, Denmark.

Guzman, J. L., K. Astrom, S. Dormido, T Hagglund, M. Berenguel, and Y. Piguet (2008). “Interactive learning modules for PID control [Lecture Notes]”. *Control Systems, IEEE* **28**:5, pp. 118–134.

Hägglund, T. and K. J. Åström (2004). “Revisiting the Ziegler-Nichols step response method for PID control”. *Journal of Process Control* **14**:6, pp. 635–650.

Hansen, P. (2012). “Parametric Robustness”. In: *IFAC Conference on Advances in PID Control*. Brescia, Italy.

Harmse, M., R. Hughes, R. Dittmar, H. Singh, and S. Gill (2009). “Robust optimization-based multi-loop PID controller tuning: A new tool and an industrial example”. In: *7th IFAC International Symposium on Advanced Control of Chemical Processes, ADCHEM’09*. Istanbul, Turkey.

Hast, M., K. J. Åström, B. Bernhardsson, and S. P. Boyd (2013). “PID design by convex-concave procedure”. In: *2013 European Control Conference*. Zürich, Switzerland.

Higham, J. D. (1968). “‘Single-term’ control of first- and second-order processes with dead time”. *Control* **12**, pp. 136–140.

Isaksson, A. (2012). “Future Perspectives of PID Control - Panel Session”. In: *IFAC Conference on Advances in PID Control*. Brescia, Italy.

Isaksson, A. and S. Graebe (2002). “Derivative filter is an integral part of PID design”. *Control Theory and Applications, IEE Proceedings* **149**:1, pp. 41–45.

Janvier, B. and B. Bialkowski (2005). “On lambda tuning – the how and why”. In: *PAPTAC 91st Annual Meeting*. Montreal, Canada.

Jelali, M. and B. Huang (2010). *Detection and Diagnosis of Stiction in Control Loops: State of the Art and Advanced Methods*. Springer London, London.

Kano, M. and M. Ogawa (2010). "The state of the art in chemical process control in japan: Good practice and questionnaire survey". *Journal of Process Control* **20**:9, pp. 969–982.

Kristiansson, B. and B. Lennartson (2002). "Robust and optimal tuning of PI and PID controllers". *Control Theory and Applications, IEE Proceedings D* **149**:1, pp. 17–25.

Kristiansson, B. and B. Lennartson (2006). "Evaluation and simple tuning of PID controllers with high-frequency robustness". *Journal of Process Control* **16** (2), pp. 91–102.

Kuzu, E. (2012). "Future Perspectives of PID Control - Panel Session". In: *IFAC Conference on Advances in PID Control*. Brescia, Italy.

Larsson, P.-O. and T. Hägglund (2009). "Robustness Margins Separating Process Dynamics Uncertainties". In: *2009 European Control Conference*. Budapest, Hungary.

Larsson, P.-O. and T. Hägglund (2011). "Control Signal Constraints and Filter Order Selection for PI and PID Controllers". In: *2011 American Control Conference*. San Francisco, California, USA.

Lee, Y., S. Park, M. Lee, and C. Brosilow (1998). "PID Controller Tuning for Desired Closed-Loop Responses for SI/SO Systems". *AIChE Journal* **44**:1, pp. 106–115.

Leva, A. and M. Maggio (2011). "A systematic way to extend ideal PID tuning rules to the real structure". *Journal of Process Control* **21**:1, pp. 130–136.

Lin, M., S. Lakshminarayanan, and G. Rangaiah (2008). "A Comparative Study of Recent/Popular PID Tuning Rules for Stable, First-Order Plus Dead Time, Single-Input Single-Output Processes". *Industrial & Engineering Chemistry Research* **47**:2, pp. 344–368.

Liu, G. and S. Daley (1999). "Optimal-tuning PID controller design in the frequency domain with application to a rotary hydraulic system". *Control Engineering Practice* **7**:7, pp. 821–830.

Liu, T., Q.-G. Wang, and H.-P. Huang (2013). "A tutorial review on process identification from step or relay feedback test". *Journal of Process Control* **23**:10, pp. 1597 –1623.

Marlin, T. (1995). *Process Control – Designing Processes and Control Systems for Dynamic Performance*. Chemical Engineering. McGraw-Hill, New York, USA.

MathWorks® (2015). *Design Optimization-Based PID Controller for Linearized Simulink Model (GUI)*. Accessed: 2015-04-01. URL: http://se.mathworks.com/help/sldo/gs/design-an-optimization-based-pid-controller-for-a-linearized-simulink-model.html.

Mayfield, D. W. and C. D. Sorensen (2010). "An improved temperature control algorithm for friction stir processing". In: *8th International Friction Stir Welding Symposium*. Timmendorfer Strand, Germany.

McMillan, G. (1983). *Tuning and Control Loop Performance*. ISA, Research Triangle Park, NC.

Micić, A. D. and M. R. Mataušek (2014). "Optimization of PID controller with higher-order noise filter". *Journal of Process Control* **24**:5, pp. 694–700.

Nordfeldt, P. and T. Hägglund (2006). "Decoupler and PID controller design of TITO systems". *Journal of Process Control* **16**:9, pp. 923–936.

O'Dwyer, A. (2009). *Handbook of PI and PID Controller Tuning Rules*. 3rd Edition. Imperial College Press, London.

Oviedo, J. J. E., T. Boelen, and P. van Overschee (2006). "Robust Advanced PID Control (RaPID) - PID Tuning Based on Engineering Specifications". *IEEE Control Systems Magazine* **26**, pp. 15–19.

Padula, F. and A. Visioli (2011). "Tuning rules for optimal PID and fractional-order PID controllers". *Journal of Process Control* **21**:1, pp. 69 –81.

Panagopoulos, H., K. J. Åström, and T. Hägglund (2002). "Design of PID controllers based on constrained optimisation". *IEE Proceedings - Control Theory & Applications* **149**:1, pp. 32–40.

Piechottka, U. and V. Hagenmeyer (2014). "A discussion of the actual status of process control in theory and practice: a personal view from German process industry". *Automatisierungstechnik* **62**:2, pp. 67–77.

Rivera, D. E., M. Morari, and S. Skogestad (1986). "Internal Model Control. 4. PID Controller Design". *Ind. Eng. Chem. Process Des. Dev.* **25**, pp. 252–265.

Romero, J. A., R. Sanchis, and P. Balaguer (2011). "PI and PID autotuning procedure based on simplified single parameter optimization". *Journal of Process Control* **21**:6, pp. 840–851.

Romero Segovia, V., T. Hägglund, and K. J. Åström (2013). "Noise filtering in PI and PID Control". In: *2013 American Control Conference*. Washington, D.C., USA.

Romero Segovia, V., T. Hägglund, and K. J. Åström (2014a). "Measurement noise filtering for PID controllers". *Journal of Process Control* **24**:4, pp. 299–313.

Romero Segovia, V., T. Hägglund, and K. J. Åström (2014b). "Measurement noise filtering for common PID tuning rules". *Control Engineering Practice* **32**, pp. 43–63.

Ross, K. (2014). "Temperature Control in Friction Stir Welding for Industrial and Research Applications". In: *10th International Symposium on Friction Stir Welding*. Beijing, China.

Sadeghpour, M., V. de Oliveira, and A. Karimi (2012). "A toolbox for robust PID controller tuning using convex optimization". In: *IFAC Conference on Advances in PID Control*. Brescia, Italy.

Sanchis, R., J. A. Romero, and P. Balaguer (2010). "Tuning of PID controllers based on simplified single parameter optimisation". *International Journal of Control* **83**:9, pp. 1785–1798.

Seborg, D. E., T. F. Edgar, and D. A. Mellichamp (2004). *Process dynamics and control*. 2. ed. Wiley, Hoboken, N.J.

Sell, N. J., (Ed.) (1995). *Process Control Fundamentals for the Pulp & Paper Industry*. Tappi Press, Technology Park, Atlanta, GA.

Shamsuzzoha, M and M. Lee (2007). "IMC-PID Controller Design for Improved Disturbance Rejection of Time-Delayed Processes". *Industrial & Engineering Chemistry Research* **46**:7, pp. 2077–2091.

Shinskey, F. G. (1996). *Process-Control Systems. Application, Design, and Tuning*. 4th. McGraw-Hill, New York, NY.

Skogestad, S. (2003). "Simple analytic rules for model reduction and PID controller tuning". *Journal of Process Control* **13**:4, pp. 291–309.

Skogestad, S. and C. Grimholt (2012). "The SIMC Method for Smooth PID Controller Tuning". In: Vilanova, R. et al. (Eds.). *PID Control in the Third Millenium*. Springer-Verlag, London. Chap. 5, pp. 147–175.

Skogestad, S. (2006). "Tuning for Smooth PID Control with Acceptable Disturbance Rejection". *Industrial & Engineering Chemistry Research* **45**:23, pp. 7817–7822.

Syrcos, G. and I. K. Kookos (2005). "PID controller tuning using mathematical programming". *Chemical Engineering and Processing* **44**:1, pp. 41–49.

Thomas, W. M., E. D. Nicholas, J. C. Needham, M. G. Murch, P. Temple-Smith, and C. J. Dawes (1991). *Friction stir butt welding*. International Patent Application No. PCT/GB92/02203.

Toscano, R (2005). "A simple robust PI/PID controller design via numerical optimization approach". *Journal of Process Control* **15**:1, pp. 81–88.

Vilanova, R. (2008). "IMC based Robust PID design: Tuning guidelines and automatic tuning". *Journal of Process Control* **18**:1, pp. 61–70.

Šekara, T. B. and M. R. Mataušek (2009). "Optimization of PID controller based on maximization of the proportional gain under constraints on robustness and sensitivity to measurement noise". *IEEE Transactions on Automatic Control* **54**:1, pp. 184–189.

Yamamoto, S. and I. Hashimoto (1991). "Present status and future needs: the view from Japanese industry". In: *Chemical Process Control-CPCIV*. Padre Island, Texas, USA.

Zhou, K. and J. Doyle (1998). *Essentials of Robust Control*. Prentice Hall International, Upper Saddle River, NJ.

Ziegler, J. G. and N. B. Nichols (1942). "Optimum Settings for Automatic Controllers". *Trans. ASME* **64**:11, pp. 759–768.

Paper I

Performance and robustness trade-offs in PID control

Olof Garpinger Tore Hägglund Karl Johan Åström

Abstract

Control design is a rich problem which requires consideration of many issues such as load disturbance attenuation, set-point tracking, robustness with respect to process variations and model uncertainty, and effects of measurement noise. The purpose of this paper is to provide insight into the trade-offs between performance and robustness explicitly. This is accomplished by introducing plots that show the trade-offs for PI and PID control. These also provide valuable understanding of design compromises used for common PI design methods.

1. Introduction

A rational way to design a controller is to derive a process model and a collection of requirements. Constrained optimization can then be applied to make a trade-off between often conflicting requirements. Tuning of PID controllers is typically not done in this way since the large number of PID loops encountered limits the effort that can be devoted to a single loop. Methods for design of PID controllers have instead focused on development of simple tuning rules based on process models characterized by few parameters, like the first-order time-delayed (FOTD) model.

Requirements typically include specifications on load disturbance attenuation, robustness to process uncertainty, measurement noise and set-point tracking. Load disturbance attenuation is a primary concern in process control where steady-state regulation is a key issue, see [Shinskey, 1996], while set-point tracking is a major concern in motion control. Set-point tracking can, however, be treated separately by using a control architecture having two degrees of freedom, which is simply done by set-point weighting in PID control, see e.g. [Åström and Hägglund, 2005] for more information. The set-point response will thus not be treated in this paper.

The trade-off between performance and robustness is a key issue in control design. In this paper we introduce new trade-off plots for PID control that show explicitly how performance and robustness depend on the controller parameters. The plots give both insight and tuning intuition, useful for practitioners as well as students. For example they show that IE and IAE are similar in some parameter regions and different in other. The trade-off plots also show the effect of the controller parameters that in some regions the integral gain controls performance while proportional gain controls robustness.

The plots are also useful to understand the properties of different tuning methods. We use them to compare some popular tuning methods: Lambda tuning, Skogestad's SIMC tuning and the AMIGO tuning rules. The trade-off plots show clearly the strengths and the weaknesses of the different methods and they also indicate how they can be improved.

The trade-offs depend on the nature of the process dynamics. In PID tuning it has been common to separate processes with and without integral action. We show that a more refined classification can be made based on the normalized time delay, which is in the range between 0 and 1. Trade-off plots are presented for three illustrative types of processes; processes that are lag-dominant, processes that are delay-dominant, and processes with a balance between lag and delay. The insight provided by these plots clearly demonstrates that there are significant differences in the trade-offs for these processes. By obtaining e.g. a first-order time-delayed model of the process, one can then choose the corresponding plot

for tuning analysis.

Since the PID controller has three parameters, three-dimensional trade-off plots are required to get full insight. To avoid this complication, two-dimensional projections are presented with the derivative gain kept constant. The paper ends with a discussion of how to incorporate the requirements on measurement noise attenuation using filtering in the trade-off plots.

2. Controllers and design criteria

A PI controller can be parametrized either in terms of proportional gain k_p and integral gain k_i or in terms of k_p and integral time T_i. A PID controller includes either the derivative gain k_d or derivative time T_d. We will consider ideal PI and PID controllers

$$\begin{aligned} C_{PI}(s) &= k_p + \frac{k_i}{s} = k_p\left(1 + \frac{1}{sT_i}\right), \\ C_{PID}(s) &= k_p + \frac{k_i}{s} + k_d s = k_p\left(1 + \frac{1}{sT_i} + sT_d\right) \end{aligned} \tag{1}$$

with first- or second-order filtering of the measured signal

$$G_f(s) = \frac{1}{1 + sT_f}, \quad G_f(s) = \frac{1}{1 + sT_f + s^2T_f^2/2}. \tag{2}$$

The transfer function of the controller and filter combination becomes

$$C(s) = C_{PI}(s)G_f(s), \qquad C(s) = C_{PID}(s)G_f(s) \tag{3}$$

An advantage with filtering the measured signal instead of the common procedure of filtering only the derivative term is that one can account for filtering by designing ideal controllers for the combination PG_f of process dynamics P and the filter G_f. Computer control can be dealt with by adding the transfer function of the sample and hold.

The PID controller in (1) is on parallel form. The series form is another parametrization which is convenient for design based on lead-lag compensation, see e.g. [Franklin et al., 2010].

Control performance can be characterized by the integrated error and the integrated absolute error

$$\text{IE} = \int_0^\infty e(t)dt, \qquad \text{IAE} = \int_0^\infty |e(t)|dt, \tag{4}$$

where e is the control error due to a unit step load disturbance. The load disturbance may enter at many different places, we will investigate the

extreme cases when they enter at the process input or at the process output, respectively. For a system with error feedback the error response to a disturbance at the process output is equivalent to the set-point response.

The integrated error for a unit step disturbance at the process input is the inverse of the controller integral gain, $\mathrm{IE}_{id} = 1/k_i$, see e.g. [Åström and Hägglund, 2005, p 129]. For a unit step output disturbance it is instead $\mathrm{IE}_{od} = 1/(K_p k_i)$, where K_p is the static gain of the process. When the closed-loop system is well damped, IE and IAE are approximately the same. The criteria IE and IAE are widely used in process control applications but other criteria such as the integrated square error, ISE, and the integrated time-weighted absolute error, ITAE, have also been suggested by e.g. [McMillan, 1983; Marlin, 1995].

Robustness to process uncertainty can be captured by the sensitivity functions

$$S(i\omega) = \frac{1}{1 + G_l(i\omega)}, \quad T(i\omega) = \frac{G_l(i\omega)}{1 + G_l(i\omega)}, \tag{5}$$

where $G_l(s) = P(s)C(s)$ is the loop transfer function. We will use

$$M_{st} = \max_{\omega}(|S(i\omega)|, |T(i\omega)|), \quad \forall \omega \in \mathbb{R}^+ \tag{6}$$

as our robustness measure. Reasonable values of M_{st} are in a range between 1.2 and 2.0, which correspond to gain margins between 6 and 2 and phase margins between 49° and 29°.

The control actions generated by measurement noise should not be large. The transfer function from measurement noise to control action is

$$G_{un}(s) = \frac{C(s)}{1 + G_l(s)}. \tag{7}$$

The impact of this transfer function depends on many factors, the controller parameters and the low-pass filter being particularly important. To determine the magnitude of the fluctuations of the control signal we must also know the characteristics of the measurement noise, for example its power spectral density. Such detailed information is rarely available and simpler measures must then be used. The high frequency gain of the transfer function (7) is essentially determined by k_p and T_f for PI control and k_d and T_f for PID control. More detailed discussions of the effect of measurement noise are given in e.g. [Isaksson and Graebe, 2002; Kristiansson and Lennartson, 2006; Garpinger, 2009; Larsson and Hägglund, 2011; Romero Segovia et al., 2013].

Summarizing, our criteria for performance, robustness and noise injection are IAE, M_{st} and some characterization of G_{un}. Under certain conditions IAE can be approximated by $1/k_i$ and criteria for noise injection by k_p (PI control) and k_d/T_f (PID control).

3. The trade-off plot

Fig. 1 shows trade-off plots for PI control of processes with the transfer functions

$$P_1(s) = \frac{1}{(s+1)(0.1s+1)(0.01s+1)(0.001s+1)}, \tag{8}$$

$$P_2(s) = \frac{1}{(s+1)^4}, \tag{9}$$

$$P_3(s) = \frac{1}{(1+0.05s)^2} e^{-s}. \tag{10}$$

The plots show the level curves for performance (IAE) during input disturbances and robustness (M_{st}) as functions of proportional (k_p) and integral gain (k_i).

In process control, process models are often approximated by the FOTD model

$$P(s) = \frac{K_p}{1+sT} e^{-sL}. \tag{11}$$

Two methods for FOTD modeling are presented in Section 4.1. Processes can be characterized based on the normalized time delay $\tau = L/(L+T)$, which ranges from 0 to 1. A process is lag-dominated if τ is small, delay-dominated if τ is large, and balanced if τ is around 0.5. There are no sharp boundaries between these three categories, but reasonable limits are $0 \leq \tau \leq 0.2$ for lag-dominated, $0.2 < \tau < 0.7$ for balanced, and $0.7 \leq \tau \leq 1.0$ for delay-dominated processes. Process P_1 is lag-dominated ($\tau = 0.067$), P_2 is balanced ($\tau = 0.33$), and P_3 is delay-dominated ($\tau = 0.92$).

The shaded areas represent controller parameters that give unstable closed-loop systems. The solid blue contours are the level curves of IAE, whose values are related to $1/k_i$. For example, an IAE level along the line $k_i = 0.1$ corresponds to IAE $= 1/0.1 = 10$. Therefore, the reader must beware the non-linear scale of the IAE levels. Level curves of IE would have been horizontal lines in the plots since IE $= 1/k_i$. Thus, level curves for IE would be identical to IAE in the lower central parts of the graph where the IAE curves are horizontal. The IAE is greater than the IE outside these regions where the controller parameters give closed-loop systems with oscillatory time response to load disturbances.

The solid red, parabola-shaped, contours are loci for constant M_{st}. High robustness (low values of M_{st}) are obtained for small values on the controller gains. The M_{st} values are indicated by numbers in the plots. Controller parameters that minimize IE for a given robustness correspond to the maxima of the robustness curves since they give the greatest k_i.

Controller parameters that minimize IAE for a given robustness are indicated by a green dash-dotted line in the graphs. Parameters that give

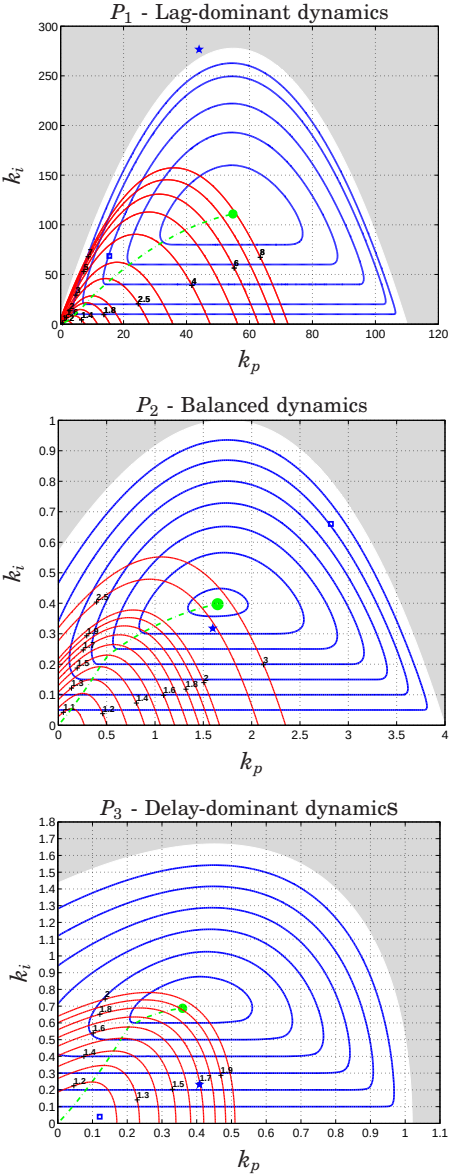

Figure 1. Trade-off plots for PI control of the processes P_1, P_2 and P_3 with the transfer functions (8)-(10). Level curves of IAE for a step disturbance at the process input are shown in blue and loci of constant M_{st} in red. IAE-values are equal to $1/k_i$ in the region where the levels are horizontal. The dash-dotted line gives the locus of controller gains that minimize IAE.

the smallest IAE are indicated by green dots. The trade-off plots show the compromise between performance and robustness directly.

Controller parameters obtained by the Ziegler-Nichols step method are indicated with squares (□) while those given by the Ziegler-Nichols frequency method are marked with stars (⋆). These and other tuning methods will be presented in more detail in Section 4.

3.1 Controller parametrization

PI controllers can be parametrized in terms of k_p, k_i or $k_p, T_i = k_p/k_i$. The trade-off plot gives an interesting insight into the different parametrizations.

Figure 1 shows that the integrated absolute error is close to $1/k_i$ except for low and high values of proportional gain. There is, however, a difference between processes with lag-dominated and delay-dominated dynamics. For delay-dominated processes robustness is reasonable even for relatively low values of k_p, but processes with lag-dominated dynamics have very poor robustness for low k_p. For fixed integral gain, robustness increases when the proportional gain is increased from $k_p = 0$, it reaches a maximum and will then decrease as the proportional gain increases. There is a region for low values of k_p where the level curves for the performance have vertical tangents. Any changes of k_p within this region will thus have a great influence on the closed-loop performance.

Now consider controllers parametrized in k_p and $T_i = k_p/k_i$. A constant integral time T_i corresponds to a line through the origin in Fig. 1 with the slope $1/T_i$. The figure shows that there is a choice of T_i that gives good balance between robustness and performance. A conservative choice is the value of T_i for the controller that gives the absolute minimum of IAE.

If T_i is fixed, k_p can be used as a parameter that controls both robustness and performance. Increasing k_p increases performance and decreases robustness. This is not the case for the $k_p - k_i$ parametrization since a change in k_p does not influence k_i. It is thus safer to use k_p and T_i as parameters for PI hand-tuning even though k_p and k_i give trade-off plots that are easier to interpret.

We will now discuss the features of the different processes in Fig. 1 in more detail.

3.2 PI control of processes with different dynamics

Table 1 summarizes controller parameters, IAE, and M_{st} that give the absolute minimum of IAE for each process P_1, P_2, and P_3. The maximum values of M_{st} for which IE = IAE are also specified in the table. The trade-off plot for process P_1, which has lag-dominated dynamics, is shown

Table 1. Controller parameters k_p^{opt} and k_i^{opt}, that gives the absolute IAE minimum, IAE^{opt}, with robustness M_{st}^{opt} for the processes P_1, P_2 and P_3. M_{st}^{IE} denotes the largest value of M_{st} for which IE = IAE.

	k_p^{opt}	k_i^{opt}	IAE^{opt}	M_{st}^{opt}	M_{st}^{IE}
P_1	54.7	110.9	0.0102	7.68	1.95
P_2	1.64	0.4	2.8	2.75	1.5
P_3	0.36	0.69	1.49	1.9	1.6

in the top graph of Fig. 1. The unconstrained minimum is obtained for $M_{st} = 7.68$, which corresponds to a closed loop with very poor robustness. Minimization of IE and IAE give the same results if the maximum sensitivity is less than $M_{st} = 1.95$, which is a larger region than for the other two processes. For lag-dominant processes, the gains are so large that it is necessary to take measurement noise into account. Assume for example that the robustness requirement is $M_{st} = 2$. The controller that minimizes IE has $k_p = 10$. Measurement noise of 1% of the signal span then results in control signal variations of about 10% of the signal span. The large control actions can be reduced by filtering or by requiring lower controller gain k_p. A natural way is to specify the largest proportional gain k_p that is acceptable from the view point of measurement noise and to pick the integral gain k_i from the dash-dotted line. The Ziegler-Nichols step response method gives a controller with low IAE, but poor robustness $M_{st} > 3$. The Ziegler-Nichols frequency method gives an unstable closed loop system.

The trade-off plot for process P_2, which has balanced dynamics, is shown in the middle graph in Fig. 1. The absolute minimum of IAE is 2.8, which also corresponds to poor closed-loop robustness with $M_{st} = 2.75$. IAE is decreased from 10 to 5.2 when M_{st} is increased from 1.2 to 1.4, indicating that there is an incentive to do frequent tuning or adaptation. Figure 1 shows that minimization of IE and IAE give the same results for M_{st} lower than 1.5. For larger values of M_{st}, minimization of IAE gives higher values of k_p and lower values of k_i. IE optimization will then start inducing worse oscillations than IAE tuning. Ziegler-Nichols step method results in both very poor robustness and bad performance, while the frequency response method gives controller parameters that are close to the parameters that minimize IAE.

The trade-off plot for the delay-dominated process P_3 is shown in the lower graph in Fig. 1. The green dot shows that the absolute minimum IAE = 1.49 occurs when the maximum sensitivity is $M_{st} = 1.9$, which corresponds to reasonable robustness. Minimization of IAE and IE give

the same controller parameters if the robustness is restricted to $M_{st} < 1.6$. The figure shows that there is significant freedom in choosing controller gains. For example, if $k_i = 0.2$ then proportional gains between 0 and 0.26 give M_{st} smaller than 1.4. Measurement noise is of little concern for this process since the controller gains are so small. Ziegler-Nichols step rule gives a controller with very low gains and thus performance is bad. Ziegler-Nichols frequency method on the other hand chooses a controller with high proportional gain, but low integral gain. This results in decent robustness, but the same performance can be achieved for a much smaller value on the maximum sensitivity.

3.3 Output disturbance

So far, we have only investigated load disturbances that enter at the process input. To assess what happens when disturbances enter the process differently we will explore the other extreme case, when the disturbances enter at the process output. This case is identical to tuning for the set-point response.

Figure 2 shows trade-off plots for PI control of P_1, P_2 and P_3 where IAE is minimized with a robustness constraint for a unit step disturbance at the process output. The grey, dash-dotted, curves show the optimal parameters, for comparison we have also plotted the locus of the parameters that optimized performance for a step disturbance at the input, in green. Notice that there is practically no difference between the case of input and output disturbance for the delay-dominated process but a large difference for lag-dominated systems. The same result was reported by e.g. [Alcántara et al., 2013]. Minimization for an output disturbance gives higher proportional gains and lower integral gains if the robustness requirement is the same.

The plots are similar to the corresponding plots for an input disturbance in Fig. 1, but there are larger differences between controllers that minimize IE and IAE, particularly for processes with lag-dominated dynamics. Notice that the absolute minima for output disturbance control give closed-loop systems with reasonable robustness, $M_{st} = 1.92$ for P_1, $M_{st} = 1.83$ for P_2 and $M_{st} = 1.90$ for P_3.

4. Tuning methods

In Section 3 we showed that the trade-off plots can give good qualitative insight into the relations between controller gains, process dynamics and design criteria. In Section 5 we will show that the trade-off plots can give guidance to the properties of different tuning methods as well. Before

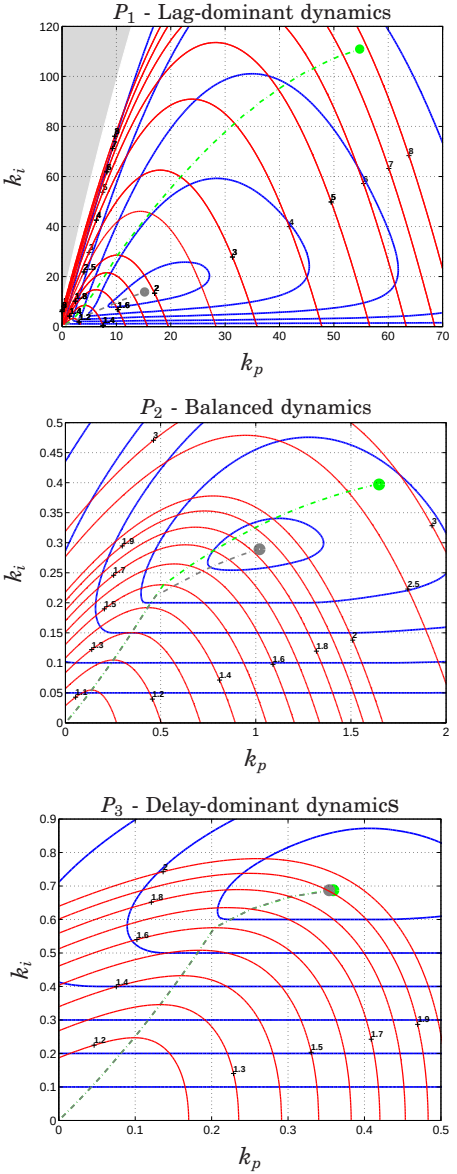

Figure 2. Trade-off plots for PI control of the processes P_1 (8), P_2 (9) and P_3 (10). Level curves of IAE for a step disturbance at the process output are shown in blue, and the loci of constant M_{st} in red. IAE-values are equal to $1/k_i$ in the region where the levels are horizontal. The grey, dash-dotted line shows the loci of optimal controller parameters. The green dash-dotted line shows the optimal parameters for an input disturbance.

discussing this in detail we will briefly summarize some common modeling and tuning methods for PI controllers.

4.1 Modeling

Many tuning rules are based on the FOTD model (11) and typically have two ingredients: a method to determine the model parameters, K_p, L and T, and a method to determine controller parameters from the model parameters. Both ingredients influence the behavior of the closed-loop system and it is therefore important to be aware of both aspects.

A common way to determine K_p, T and L is based on an open-loop step response of the process. The gain K_p is the steady state gain. The apparent time delay L is the t-coordinate of the intersection of the steepest tangent with the time axis, and $L+T$ is the time when the step response has reached 63% of its steady state value. We call this method the 63%-rule.

The Ziegler-Nichols step response method for PI and PID design uses only two parameters, K_v and L. These are determined from a step response where K_v is the steepest slope and L is the intersection of the steepest tangent with the time axis. L is therefore the same as in the 63%-rule.

The Ziegler-Nichols frequency response method is based on a closed-loop test with a P controller. The proportional gain is increased until the process is on the border to instability. This proportional gain is called the ultimate gain, K_u, and the oscillation period is called the ultimate period, T_u.

Skogestad's design methods are based on reduction of an accurate process model [Skogestad, 2003]. The gain K_p is the static gain. The apparent time constant T is the largest time constant of the process plus half of the largest neglected time constant. The apparent time delay L is the sum of the true time delay, half of the largest neglected time constant and all other time constants. The rule is called the half-rule because the largest neglected time constant is distributed evenly to the apparent delay and the apparent lag. A nice feature of Skogestad's method is that it can also be used to obtain models of second order. A drawback is that it requires an accurate process model.

FOTD approximations of the processes (8)–(10) by the 63%-rule and the half-rule are given in Table 2. Notice that there are significant differences in the parameters for process P_2 which has balanced dynamics. When comparing tuning methods it is important to be aware of the method used to obtain the model parameters. The Z-N-rules only use two process parameters and cannot distinguish between processes having lag-dominated or delay-dominated dynamics.

Table 2. Apparent time delay L, apparent time constant T and normalized time delay τ for the processes P_1 (8), P_2 (9) and P_3 (10) obtained by the 63%-rule and Skogestad's half-rule (1/2).

	P_1		P_2		P_3	
	63%	1/2	63%	1/2	63%	1/2
L	0.075	0.061	1.42	2.50	1.014	1.025
T	1.041	1.050	2.93	1.50	0.093	0.075
τ	0.067	0.055	0.33	0.63	0.92	0.93

4.2 The Ziegler-Nichols methods

The Ziegler-Nichols methods are included only for historical reasons. Robustness to process uncertainty and measurement noise were not considered. The step response method give the controller parameters

$$k_p = \frac{0.9}{K_v L}, \qquad k_i = \frac{0.3}{K_v L^2}, \qquad T_i = 3L. \tag{12}$$

The frequency response method is tuned to give quarter amplitude damping of load disturbance responses and is given by

$$k_p = 0.4K_u, \qquad k_i = \frac{0.5K_u}{T_u}, \qquad T_i = 0.8T_u. \tag{13}$$

As we have seen in Fig. 1, the Ziegler-Nichols methods result in poor control for all three processes P_1, P_2 and P_3. We should revere the Ziegler-Nichols rules for the original ideas, but they are not useful for practical PID tuning and there are better methods to teach in e.g. basic control courses.

4.3 Lambda tuning

Lambda tuning originated in early computer control [Dahlin, 1968; Higham, 1968], was rediscovered in connection with the development of internal model control [Rivera et al., 1986] and is today widely adopted in the industry, see e.g. [Sell, 1995; Bialkowski, 1996]. Modeling is typically based on measured step responses and the 63%-rule is used to obtain an FOTD model. The desired closed-loop time constant T_{cl} is used as a tuning parameter. In the original work by Dahlin, T_{cl} was called λ, which explains the name of the tuning method. This parameter admits a compromise between performance and robustness. For PI control the tuning formulas are

$$k_p = \frac{T}{K_p(T_{cl} + L)}, \quad k_i = \frac{1}{K_p(T_{cl} + L)}, \quad T_i = T \tag{14}$$

The process time constant is canceled by the controller zero which gives sluggish load disturbance responses for processes with lag-dominated dynamics. There are several suggestions for choosing T_{cl}, some can be found in [Sell, 1995; Bialkowski, 1996]. We will mark the original choices of $T_{cl} = T, 2T$ or $3T$ in the trade-off plots, but the whole set of possible controllers will also be plotted as a line through these three choices. $T_{cl} = T$ corresponds to aggressive tuning and $T_{cl} = 3T$ to robust tuning.

4.4 Skogestad's methods

Skogestad [Skogestad, 2003] introduced modifications of the Lambda tuning method called SIMC that improves performance especially for lag-dominant processes. The FOTD model is obtained by model reduction of a high-order process model using the half-rule.

SIMC is closely related to Lambda tuning. Unlike the original Lambda tuning, however, the desired closed-loop time constant T_{cl} is chosen as a factor of the apparent time delay L. Skogestad recommends $T_{cl} = L$ as the primary choice which typically gives a sensitivity close to $M_{st} = 1.6$. Less aggressive tuning is obtained for larger values of T_{cl} such as $2L$ and $3L$. In the trade-off plots we will examine these three choices of T_{cl}. The parameters in the tuning rules were obtained by fitting to controllers that were optimized both for input and output disturbances.

The proportional gain is chosen as k_p in (14) and the integral time as

$$T_i = \min(T, 4(T_{cl} + L)). \tag{15}$$

This choice avoids cancellation of a slow process pole by a controller zero when $T > 4(T_{cl} + L)$.

A modified method, here called SIMC+, was introduced in [Skogestad and Grimholt, 2012] to improve performance for delay-dominated systems where the SIMC rule results in controllers with too low proportional gain. Proportional gain and integral time were instead chosen as

$$k_p = \frac{T + L/3}{K_p(T_{cl} + L)}, \quad T_i = \min(T + L/3, 4(T_{cl} + L)). \tag{16}$$

For delay-dominated and balanced systems, the integral gain is the same as for SIMC, the proportional gain k_p and integral time T_i are larger than for SIMC.

4.5 Constrained optimization - AMIGO

The MIGO tuning method presented in [Panagopoulos et al., 2002] minimizes IE subject to the robustness constraint $M_{st} \leq 1.4$. AMIGO [Åström and Hägglund, 2005] is a design method obtained by applying MIGO to

a large test batch of process models and using simple parameter fitting. The parameters of the model are determined by the 63%-rule. The AMIGO tuning rule is

$$\begin{aligned} k_p &= \frac{0.15}{K_p} + \left(0.35 - \frac{LT}{(L+T)^2}\right)\frac{T}{K_p L} \\ T_i &= 0.35L + \frac{13LT^2}{T^2 + 12LT + 7L^2} \end{aligned} \tag{17}$$

5. Assessment of tuning methods

The trade-off plots will now be used to assess the tuning methods for PI control described in Section 4. The processes P_1 (8), P_2 (9) and P_3 (10) will be used to represent processes with lag-dominated, balanced and delay-dominated dynamics, and we will consider controllers that minimize IAE for unit step load disturbances at the process input. Figure 3 is similar to Fig. 1 but the scales are different to show the parameter ranges of interest for tuning more clearly. The parameters corresponding to the different tuning rules are shown in the diagram.

The Ziegler-Nichols rules fall outside the plots except for the process with delay-dominated dynamics in the case of the step response method, see Fig. 1. Both Lambda tuning, marked λ, and the SIMC rules, marked S and S^+ respectively, use the tuning parameter, T_{cl}. These rules are represented by lines with marks for the choices of the design parameter given in Sections 4.3 and 4.4. The AMIGO rule is marked with a blue cross and the letter A. The black line marked with triangles show the following approximation

$$k_i = \frac{k_p + 0.1K_p k_p^2}{0.3L + 0.7T} \tag{18}$$

of the optimal, dash-dotted line. The approximation is valid for sensitivities $M_{st} \leq 1.6$ and the model parameters L and T are given by the 63%-rule.

5.1 Lag-dominated dynamics

The trade-off plot for the system P_1 with lag-dominated dynamics is shown in the top plot of Fig. 3. Lambda tuning, with $T_{cl} = T, 2T, 3T$ gives closed-loop systems with very low gains. The sensitivities are all less than 1.1, but the performance is poor because the gains are much too low. Lower values on T_{cl} give better performance, but it is still poor compared to the optimal controllers. Comparing the set of controllers obtainable using Lambda tuning with the output disturbance optimal curve in Fig. 2, however, shows that they are almost identical. Both SIMC rules give similar

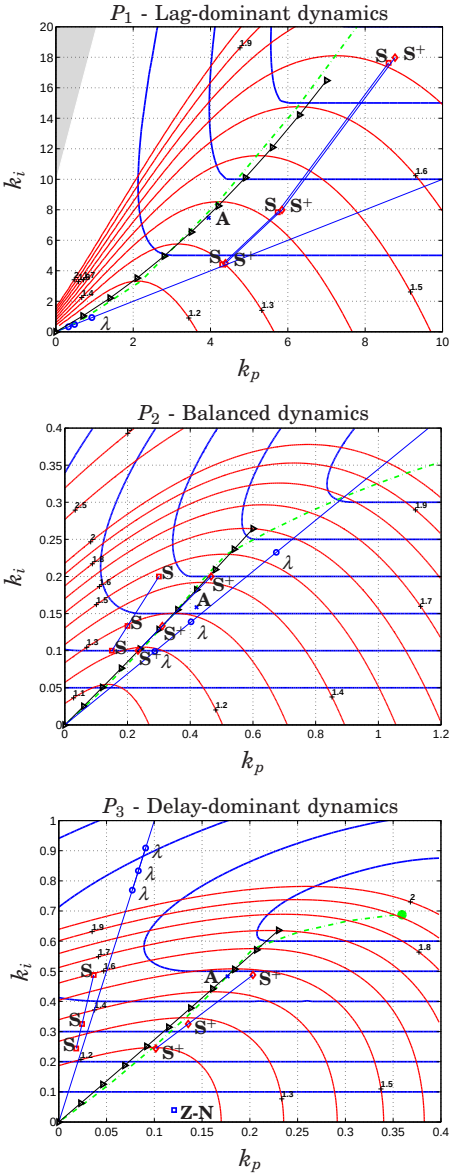

Figure 3. Trade-off plots for PI control of the processes P_1, P_2 and P_3 with controller parameters obtained with the tuning methods: Lambda tuning (λ), SIMC (S), modified SIMC (S^+) and AMIGO (A). IAE-values are equal to $1/k_i$ in the region where the levels are horizontal. The black lines, marked by triangles, show a parametrization of the dash-dotted lines for reasonable robustness given by (18).

performance with the choices $T_{cl} = L, 2L$ and $3L$. The sensitivity for the nominal design is $M_{st} = 1.7$ which is close to the design value 1.6. The controller gains are generally higher than the constrained IAE-optimal controllers. The AMIGO method provides a controller which is close to its design value $M_{st} = 1.4$.

5.2 Balanced dynamics

The trade-off plot for the system P_2 with balanced dynamics is shown in the center plot of Fig. 3. Lambda tuning gives closed-loop systems with good performance and robustness, but the proportional gains are a little too high compared to the optimal controllers. SIMC gives good controllers, but the proportional gains are a little too low. SIMC+ gives close to optimal performance, but the sensitivities are lower than the design values. Notice, however, that by using SIMC and SIMC+ on the FOTD model derived using the 63%-rule, with $T_{cl} = L$, the resulting controllers would have the gains $k_p = 1.04$, $k_i = 0.36$ and $k_p = 1.20$, $k_i = 0.36$ respectively. Both controllers give poor closed-loop robustness with $M_{st} > 2$, showing how important it is to use the recommended modeling procedure together with the design method. AMIGO tuning gives a controller close to that of the Lambda tuning with a robustness slightly higher than the designed value.

5.3 Delay-dominated dynamics

The trade-off plot for the system P_3 with delay-dominated dynamics is shown in the bottom plot of Fig. 3. The original Lambda tuning choices of $T_{cl} = T$, $2T$ and $3T$ give closed-loop systems with very poor robustness. The sensitivities are all larger than 2. Choosing a smaller value of T_{cl} gives better robustness, but the proportional gains are still too low and the integral gains too high compared with the optimal controllers. The SIMC rule gives robustness that agrees well with the design value, but the proportional gains are much too low. SIMC+ have the same integral gains, but larger proportional gains than SIMC. The performance is close to optimal while the sensitivity for the case $T_{cl} = L$ is slightly lower than the design value. The AMIGO rule gives a controller close to optimal with $M_{st} = 1.46$, to compare with the design value 1.4.

5.4 Parametrization of optimal controllers

The green line that corresponds to optimal tuning is close to a straight line through the origin for balanced and delay-dominated systems and it is slightly curved for systems with lag-dominated dynamics. Choosing a T_i that corresponds to that line is a convenient way of describing how performance and robustness varies simply by changing the gain. The

parametrization (18) of optimal controllers provides an even better match and it is possible to relate it to a certain M_{st}-value with the following iteration:

1. Choose a desired robustness value M_{st}^d
2. Determine an FOTD model with the 63%-rule
3. Iterate over k_p to determine k_i and M_{st} with respect to the FOTD model, M_{st}^{FOTD}, for every choice of k_p until $M_{st}^{FOTD} \approx M_{st}^d$

The Swedish pulp and paper industry has a standard for control optimization [SSG, 2004], where they recommend the noise level for mechanic actuators to be at most 0.5% of its total span. k_p can then be set to the minimum value of the k_p that gives M_{st}^d and the k_p that gives a noise level of 0.5%.

6. PID control

The trade-off plot is a simple useful representation for PI control because there are only two controller parameters. The situation is more complicated for PID control because the controller has three parameters and a proper representation will require three-dimensional plots. To avoid this complication we will present two-dimensional projections where the derivative gain is constant. Another complication is that the robustness region has boundaries with edges. In the two-dimensional projection the edges show up as contours with discontinuous derivatives, see [Åström and Hägglund, 2005; Åström et al., 1998]. Since derivative action has practically no benefit for systems with delay-dominated dynamics (see e.g. [Hägglund and Åström, 2004]) we will focus on systems with lag-dominated and balanced dynamics.

6.1 Lag-dominated dynamics

Trade-off plots for the lag-dominated process P_1 with transfer function (8) is shown in Fig. 4, with derivative gains $k_d = 3$, 4.5, and 6. These derivative gain values were chosen with respect to the optimal PID controller when $M_{st} = 1.4$ for which $k_d = 4.59$. Comparing with the corresponding plot for PI control, i.e. the top plot in Fig. 1, we find that the gains are significantly larger. The absolute minima of IAE correspond to controllers with sensitivities above 3.5 in all three cases. Minimizing IAE without a robustness constraint thus gives systems with poor robustness, even though it is better than in the PI case. The level curves for the sensitivities have edges for sensitivities 2 and lower.

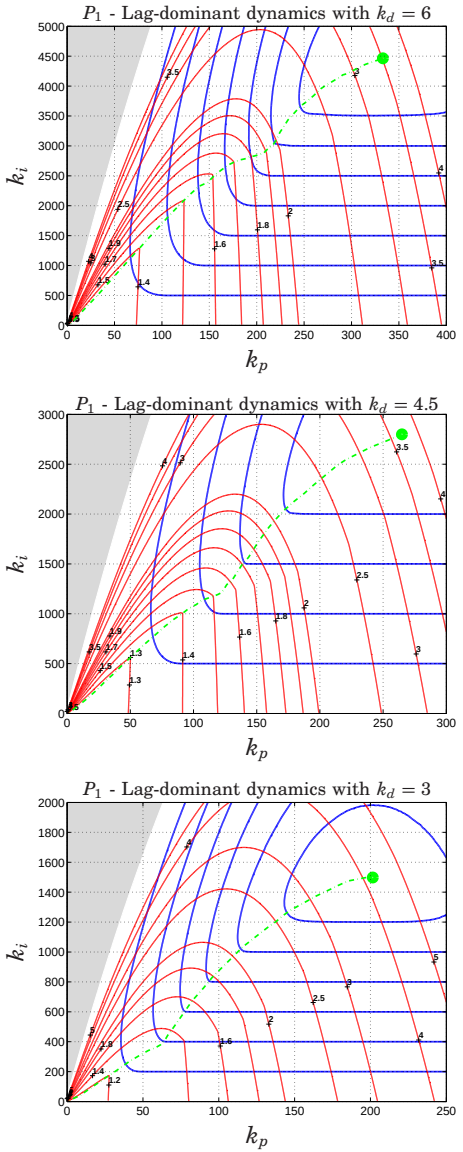

Figure 4. Trade-off plots for PID control of the process $P_1(s) = 1/((s + 1)(0.1s + 1)(0.01s + 1)(0.001s + 1))$, and derivative gains $k_d = 3$ (lower graph), $k_d = 4.5$ (middle graph), and $k_d = 6$ (top graph). The dash-dotted lines are the loci of controller gains that minimize IAE for a given robustness. IAE-values are equal to $1/k_i$ in the region where the levels are horizontal.

If we require that the maximum sensitivity is $M_{st} = 1.4$, Fig. 4 shows that the controllers with $k_d = 3, 4.5$ and 6 have the IAE values 0.0021, 0.0013, and 0.0016. The constrained minimum is $IAE = 0.0013$ for $k_p = 89.48$, $k_i = 1037.5$, and $k_d = 4.59$. The IAE value can be compared with the corresponding value for PI control IAE = 0.1175. Adding derivative action thus improves performance by a factor 90. Since the gains are large it is important to consider the effect of measurement noise and it may therefore be essential to either impose constraints on the proportional and derivative gains or add a noise filter. The dash-dotted lines in the trade-off plots give guidance for detuning.

6.2 Balanced dynamics

Fig. 5 presents trade-off plots for the process P_2 with derivative gains $k_d = 1$, 2, and 3. The derivative gains were chosen with respect to the optimal PID controller when $M_{st} = 1.4$ for which $k_d = 1.78$. Comparing with the corresponding plot for PI control, the center plot in Fig. 1, we find that the plots are similar. The gains are larger with derivative action and the sensitivity curves have peaks with discontinuous derivatives. The absolute minima of IAE without robustness constraint correspond to controllers with poor robustness, the sensitivities are close to 2.5 in all three cases. If we require that the sensitivities are less than 1.4, Fig. 5 shows that the controllers with $k_d = 1, 2$ and 3 have the IAE values 2.5, 2.2 and 3.5. The constrained minimum for $M_{st} = 1.4$ is IAE = 2.14 for $k_p = 1.33$, $k_i = 0.63$, and $k_d = 1.78$. The IAE value can be compared with the corresponding value for PI control, IAE = 4.4, adding derivative actions thus improves performance by a factor 2.

Figure 5 also shows that minimization of IE and IAE do not give the same controllers except if the robustness constraint requires very low sensitivity. The dash-dotted line which corresponds to the constrained minimum has a plateau for large values of k_d, giving it quite a different shape than for PI control. Notice that the dash-dotted line is close to the robustness constraint levels for small values of the sensitivities. This means that the performance can be improved quite a bit almost without costing anything in terms of robustness. This was also the case for process P_1 in Fig. 4.

6.3 Measurement noise filtering

One reason why derivative action is not so commonly used in process industry is that undesired valve motions are created by measurement noise. The effect can be reduced by filtering the measurement signal, but good rules for finding the filter time constant are missing. As pointed

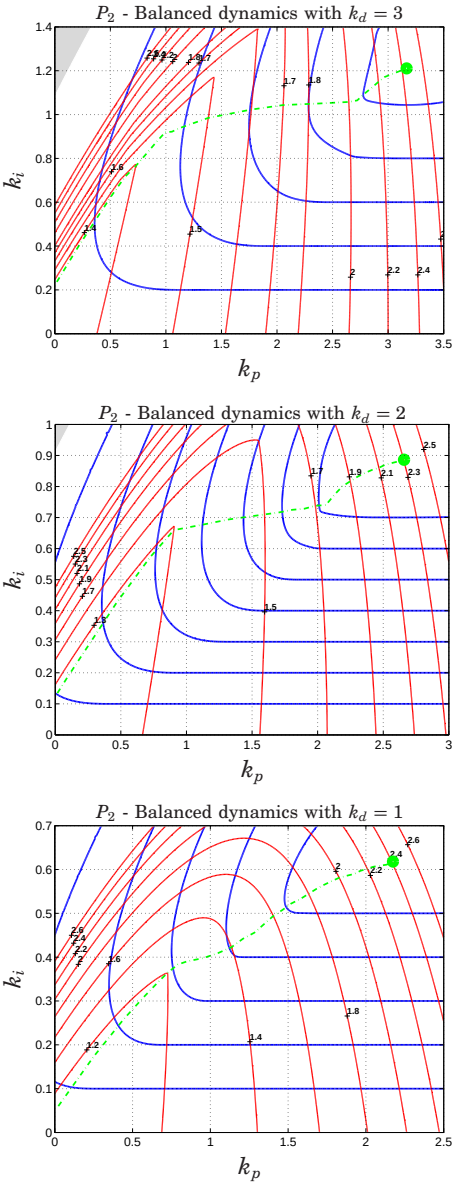

Figure 5. Trade-off plots for PID control of the process $P_2(s) = 1/(s+1)^4$, and derivative gains $k_d = 1$ (lower graph), $k_d = 2$ (middle graph), and $k_d = 3$ (top graph). The dash-dotted lines are the loci of controller gains that minimize IAE for a given robustness. IAE-values are equal to $1/k_i$ in the region where the levels are horizontal.

out by Isaksson and Graebe [Isaksson and Graebe, 2002], the filter time constant should be included in the control design.

To illustrate the effect of measurement noise, we consider the balanced process, P_2, with discrete-time white Gaussian measurement noise with standard deviation $\sigma_n = 1\%$ and a sampling time of 0.01 seconds. The PID controller designed for $M_{st} = 1.4$ in Section 6.2 has the parameters $k_p = 1.33$, $k_i = 0.63$ and $k_d = 1.78$. Using a first-order filter on the derivative part with filter time constant $T_f = T_d/N$, and the common choice $N = 10$, we get $T_f = 0.13$. A Matlab Simulink simulation gives the standard deviation of the control signal $\sigma_u = 14.4\%$. The maximum sensitivity increase to $M_{st} = 1.52$ from 1.4 because of the added dynamics, at the same time as the IAE-value for an input disturbance improves to IAE = 2.07 from 2.14. A greater N-value would give better noise filtering, but at the same time also give worse robustness.

Garpinger's PID tuning method in [Garpinger, 2009] can deal with filtering and at the same keep the robustness constraints satisfied. Assuming that we want to decrease the variation in the control signal from $\sigma_u = 14.4\%$ to $\sigma_u = 0.5\%$, as suggested in Section 5.4, we can proceed as follows. Introduce a second order measurement filter with fixed filter dynamics and perform the controller optimization. Then iterate on T_f until the standard deviation of the control signal satisfies the constraints. Using this procedure, we find that $T_f = 0.38$, $k_p = 0.91$, $k_i = 0.40$ and $k_d = 1.22$, with IAE = 3.18, fulfill both the robustness and the noise criteria. The trade-off plot with $k_d = 1.22$ and $T_f = 0.38$ is shown in Fig. 6. As one could expect, the controller gains are lower than without filtering, however, the level curves look similar to those without filtering. Notice that T_f is significantly larger than the value obtained by the rule of thumb $N = 10$. The IAE-value is 49% higher than with the unfiltered controller. The best PI controller has IAE = 5.2, 63% higher than the filtered PID controller, and $\sigma_u = 0.43\%$. This shows that there is a lot to gain from using derivative action, even when measurement noise is taken care of.

7. Conclusions

A graphical tool which provides insight into the trade-offs between performance and robustness for PID control has been introduced. The trade-off is represented by charts which show the level curves of performance, expressed by integrated absolute error (IAE), and robustness, expressed by the maximum sensitivity (M_{st}), as functions of proportional and integral gain. For PID control the charts are drawn for fixed values of the derivative gain. The trade-off charts are useful to provide intuition about tuning,

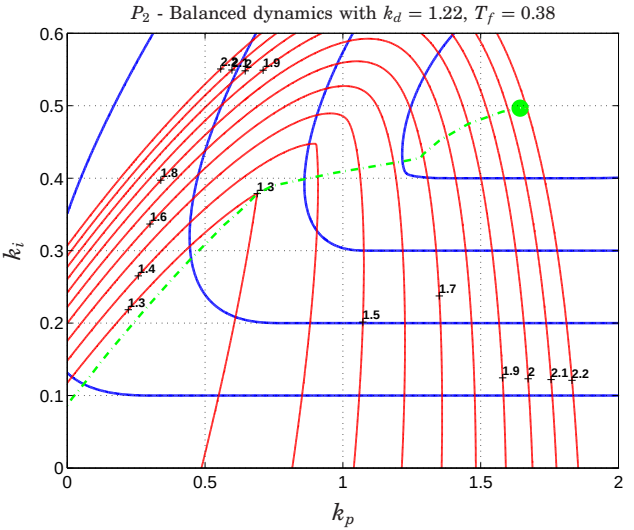

Figure 6. Trade-off plot for PID control of the balanced process $P_2(s) = 1/(s+1)^4$, with derivative gain $k_d = 1.22$ and low-pass filter time constant $T_f = 0.38$. IAE-values are equal to $1/k_i$ in the region where the levels are horizontal.

both for practitioners and students. They can also show the difference between tuning criteria based on input and output disturbances.

The level curves of performance are practically horizontal in large areas of the plots, which indicate that the integral gain is directly related to performance and the proportional gain directly related to robustness. These areas have interesting properties: IAE is a monotone function of k_i and close to IE, M_{st} is a function of k_p with a maximum for each robustness value. For fixed integral time, controllers are represented by straight lines in the plots, and adjustments of the proportional gain then influence both robustness and performance. This fact can be used empirically by exploring a few values of the proportional gain. There is also a parametrization of controllers (18) that corresponds to optimal tuning for input disturbances, which can be related to a certain robustness through iteration over k_p.

In process control it has been a tradition to separate processes with and without integral action. A more graduated separation can be obtained by introducing the normalized time delay τ, which is in the range 0 to 1. For an FOTD model we have $\tau = L/(L+T)$. Analysis of the trade-off charts shows that there is a significant difference between processes with lag-dominant ($\tau \leq 0.2$), balanced $0.2 < \tau < 0.7$, or delay-dominant ($\tau \geq 0.7$) dynamics. By obtaining an FOTD model e.g. with the 63%-rule

one can then calculate τ and choose a corresponding plot to analyze. The benefit of derivative action is marginal for processes with delay-dominated dynamics but significant for processes with lag-dominated dynamics. The proportional gain is typically high for processes with lag-dominated dynamics indicating the necessity to take measurement noise into account even with PI control. With PID control the proportional gains are high even for processes with balanced dynamics indicating the necessity of filtering the measured signal. The trade-off plot with filtering is similar to the unfiltered case, but the gains are smaller.

The trade-off plots were used to compare different tuning methods for PI control. For input disturbances, Lambda tuning only gives good results for processes with balanced dynamics, although it is close to optimal for output disturbances on lag-dominant processes. The weakness of the Lambda method lies in the choice $T_i = T$, which gives a set of controllers with the slope $1/T$. As the plots indicate, this slope deviates significantly from the set of optimal controllers. For delay-dominant processes, Lambda tuning will either give poor robustness or poor performance, depending on the choice of T_{cl}. SIMC gives decent results for at least lag-dominant and balanced processes. SIMC+ is a real improvement since it gives close to optimal tuning also for systems with delay-dominated dynamics. Both SIMC methods depend on having an accurate process model which is reduced using Skogestad's half-rule. The predicted sensitivity of $M_{st} = 1.6$ for Skogestad's rules with $T_{cl} = L$ is close to the truth. The AMIGO method is based on FOTD models, where the parameters are determined using the step response method. It gives controllers with sensitivity close to the specified value $M_{st} = 1.4$.

Acknowledgements

This work was partly funded by the Swedish Foundation for Strategic Research through the PICLU center. The authors are members of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University.

References

Alcántara, S., R. Vilanova, and C. Pedret (2013). "PID control in terms of robustness/performance and servo/regulator trade-offs: A unifying approach to balanced autotuning". *Journal of Process Control* **23**:4, pp. 527–542.

Åström, K. J. and T. Hägglund (2005). *Advanced PID Control*. ISA – The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC.

Åström, K. J., H. Panagopoulos, and T. Hägglund (1998). "Design of PI Controllers based on Non-Convex Optimization". *Automatica* **34**:5, pp. 585–601.

Bialkowski, W. L. (1996). "Control of the Pulp and Paper Making Process". In: Levine, W. S. (Ed.). *The Control Handbook*. CRC Press, Inc, Boca Raton, Florida. Chap. 72, pp. 1219–1242.

Dahlin, E. B. (1968). "Designing and tuning digital controllers". *Instruments and Control Systems* **41**:6, pp. 77–83.

Franklin, G. F., J. D. Powell, and A. Emami-Naeini (2010). *Feedback Control of Dynamic Systems*. 6th Edition. Pearson, Upper Saddle River, NJ.

Garpinger, O. (2009). *Design of Robust PID Controllers with Constrained Control Signal Activity*. Licentiate Thesis ISRN LUTFD2/TFRT--3245--SE. Department of Automatic Control, Lund University, Sweden.

Hägglund, T. and K. J. Åström (2004). "Revisiting the Ziegler-Nichols step response method for PID control". *Journal of Process Control* **14**:6, pp. 635–650.

Higham, J. D. (1968). "'Single-term' control of first- and second-order processes with dead time". *Control* **12**, pp. 136–140.

Isaksson, A. and S. Graebe (2002). "Derivative filter is an integral part of PID design". *Control Theory and Applications, IEE Proceedings* **149**:1, pp. 41–45.

Kristiansson, B. and B. Lennartson (2006). "Evaluation and simple tuning of PID controllers with high-frequency robustness". *Journal of Process Control* **16** (2), pp. 91–102.

Larsson, P.-O. and T. Hägglund (2011). "Control Signal Constraints and Filter Order Selection for PI and PID Controllers". In: *2011 American Control Conference*. San Francisco, California, USA.

Marlin, T. (1995). *Process Control – Designing Processes and Control Systems for Dynamic Performance*. Chemical Engineering. McGraw-Hill, New York, USA.

McMillan, G. (1983). *Tuning and Control Loop Performance*. ISA, Research Triangle Park, NC.

Panagopoulos, H., K. J. Åström, and T. Hägglund (2002). "Design of PID controllers based on constrained optimisation". *IEE Proceedings - Control Theory & Applications* **149**:1, pp. 32–40.

Rivera, D. E., M. Morari, and S. Skogestad (1986). "Internal Model Control. 4. PID Controller Design". *Ind. Eng. Chem. Process Des. Dev.* **25**, pp. 252–265.

Romero Segovia, V., T. Hägglund, and K. J. Åström (2013). "Noise filtering in PI and PID Control". In: *2013 American Control Conference*. Washington, D.C., USA.

Sell, N. J., (Ed.) (1995). *Process Control Fundamentals for the Pulp & Paper Industry*. Tappi Press, Technology Park, Atlanta, GA.

Shinskey, F. G. (1996). *Process-Control Systems. Application, Design, and Tuning*. 4th. McGraw-Hill, New York, NY.

Skogestad, S. (2003). "Simple analytic rules for model reduction and PID controller tuning". *Journal of Process Control* **13**:4, pp. 291–309.

Skogestad, S. and C. Grimholt (2012). "The SIMC Method for Smooth PID Controller Tuning". In: Vilanova, R. et al. (Eds.). *PID Control in the Third Millenium*. Springer-Verlag, London. Chap. 5, pp. 147–175.

SSG (2004). *Regleroptimering (in Swedish)*. 3rd. SSG 5253.

Paper II

A software tool for robust PID design

Olof Garpinger Tore Hägglund

Abstract

This paper presents a fast, interactive and easily modifiable software tool for robust PID design. The MATLAB®-based program is supposed to give people with moderate knowledge on PID control a possibility to learn more and also be a future part of an autotuner. The PID design is made by minimizing the integrated absolute error value during a load disturbance on the process input. The optimization is performed with H_∞-constraints on the sensitivity and complementary sensitivity function, providing a robust closed-loop system. Nelder-Mead optimization is used with the AMIGO method providing an initial controller. The proposed method works well, and is very efficient, on a large batch of systems common in process industry. The design tool is also shown to work on a highly oscillatory process model.

1. Introduction

The PID controller is by far the most common controller in industry today. Even so, a lot of these controllers are poorly tuned. Two of the main reasons for that is lack of knowledge and time among the operators. As a consequence, many controllers are set to default values. In other cases, the derivative part is turned off because it was not used correctly, giving noisy signals. It would therefore be a good idea to educate the operators in the possibilities of the PID controller and to provide them with simple and fast design tools. This paper describes a program that achieves both goals and should be useful for people in the industry as well as for academics.

There are many PID design methods available today and some of the most famous are collected and analysed in [Åström and Hägglund, 2005]. Of these, the MIGO and AMIGO methods (also see [Panagopoulos et al., 2002] and [Hägglund and Åström, 2004]) are probably those most worth mentioning in connection to this paper. They are based on optimization of load disturbance rejection under robustness constraints. A further development of the MIGO method, and largely based on the same method as used in this paper, was presented in [Nordfeldt, 2005]. An advantage with Nordfeldt's method is that it also works for some more advanced process structures. This paper focuses on the software that solves the optimization problem and how it can be used to increase people's understanding of PID control.

The proposed PID control design method is incorporated in several MATLAB® functions. There are many good reasons to have a software based tool for control design and analysis. In [Åström and Hägglund, 2001] it is pointed out that it would be of great value to have software that can give persons with moderate knowledge on PID controllers a possibility to experiment on those and at the same time be able to use the program to build controllers for a real plant, by incorporating it into an autotuning procedure. For simulation experiments and real use purposes, the presented software is able to provide a well working controller with analysis tools in just a few seconds time. The advanced user should also be able to modify the optimization problem to broaden the possibilities. Besides the proposed program, which is intended to be free of charge and downloadable, there are already several commercial software packages able to provide PID design tools using a variety of methods. Many of these are collected in [Li et al., 2006] and another one with very similar features to the proposed is presented in [Oviedo et al., 2006].

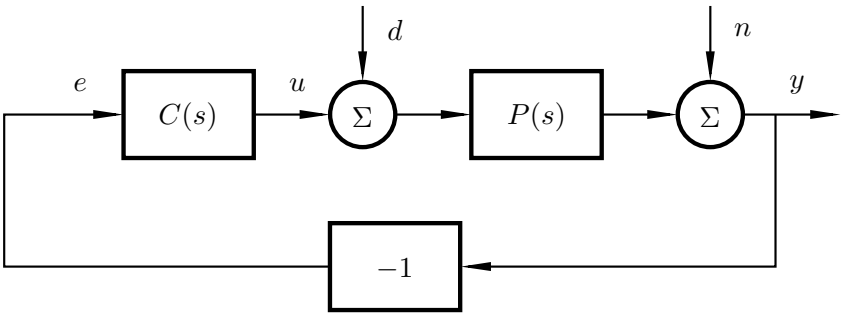

Figure 1. A load disturbance, d, and measurement noise, n, act on the closed-loop system with process $P(s)$ and PID controller $C(s)$.

2. Design criterion

The proposed PID controller design tool is mainly meant to work well for systems common in process industry. The kind of plants encountered there are often stable, monotone and primarily affected by low frequency load disturbances.

In order for the controller design to work well on a process, $P(s)$, it is important to take all system signals into consideration, especially if optimization is used. Figure 1 shows a block diagram of the system that the PID controller, $C(s)$, is designed for. There are two external signals entering the system, namely load disturbance d (mainly low frequency) and measurement noise n (assumed high frequency). Of the closed-loop transfer functions, those of greatest interest for this paper are the complementary sensitivity function $T(s)$ and the sensitivity function $S(s)$, defined as

$$T(s) = \frac{P(s)C(s)}{1 + P(s)C(s)}, \quad S(s) = \frac{1}{1 + P(s)C(s)}.$$

The PID controller is on parallel form with a second order low pass filter

$$C(s) = K\left(1 + \frac{1}{sT_i} + sT_d\right) \cdot \frac{1}{1 + sT_f + (sT_f)^2/2},$$

on the measurement signal. T_f, is chosen to weight the degree of measurement noise rejection.

The objective of the proposed PID design method is to find the PID controller giving the least integrated absolute error (IAE) value when a load disturbance d, modelled as a step, is acting on the closed-loop system.

The optimization is done under the constraints that the closed-loop system is stable and that the open loop Nyquist curve is tangent to one or two prespecified circles in the complex plane without entering either of them (see Fig. 2), thus maximizing the gain. These two circles are called the M_s- and M_t-circles, which sizes and positions are given by

$$M_s = \max_{\omega} |S(i\omega)|, \quad M_t = \max_{\omega} |T(i\omega)|,$$

hence the names. The resulting, non-convex, optimization problem can be written as

$$\min_{K,T_i,T_d \in \mathbb{R}^+} \int_0^\infty |e(t)| dt = \text{IAE}_{load} \tag{1}$$

$$\begin{aligned} \text{subject to } & |G_o(i\omega) - C_{M_s}|^2 \geq R_{M_s}^2 \quad \forall \omega \in \mathbb{R}^+, \\ & |G_o(i\omega) - C_{M_t}|^2 \geq R_{M_t}^2 \quad \forall \omega \in \mathbb{R}^+, \\ & |G_o(i\omega^s) - C_{M_s}|^2 = R_{M_s}^2, \\ & |G_o(i\omega^p) - C_{M_t}|^2 = R_{M_t}^2, \end{aligned}$$

where $e(t)$ is the control error, $G_o(i\omega)$ is the open-loop frequency response, ω^s are frequencies for which $G_o(i\omega)$ is tangent to the M_s-circle and vice versa for ω^p on the M_t-circle. Either ω^s or ω^p could be an empty vector, but not at the same time. The radius and centre point of the M_s-circle are denoted by R_{M_s} and C_{M_s} respectively, with corresponding measures for the M_t-circle, R_{M_t} and C_{M_t}. Small M_s- and M_t-values result in large circles. In the software, the maximum allowed M_s- and M_t-values can be prespecified by the user ($M_s = M_t = 1.4$ is default, resulting in 41.8° phase margin). The M_s- and M_t- criterions are known to set the closed-loop robustness towards process variations, disturbances and nonlinearities as described in [Åström and Hägglund, 2005]. MIGO on the other hand uses a simplified robustness criterion called the M-circle, defined as the smallest circle that can be drawn around both the M_s- and M_t-circle.

3. Algorithm overview

The main goal of the new design algorithm was to develop a fast, interactive and easily modifiable software tool for robust PID design.

A non-convex optimization problem like (1) may have many local minima. It is therefore hard to guarantee that the solution obtained always is the global solution. It is also difficult to draw any general analytical conclusions as the problem is far from trivial. The method of gridding does

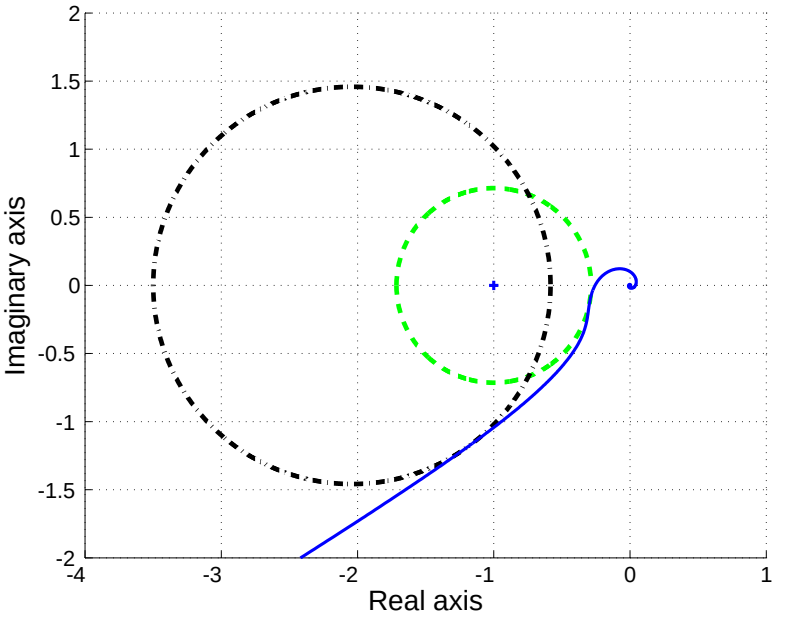

Figure 2. The M_s-circle (dashed), M_t-circle (dash-dotted) and the open-loop Nyquist curve (solid) when the optimization criterions are fulfilled.

however give a possibility of drawing surface plots of the cost function. These can be used to determine whether or not it is likely that a given solution is in fact the global minimum. This is also the major reason why gridding is an optional optimization method in the proposed design program.

Analysis of many cost function surfaces have shown that if not all, then at least a lot of them only have one minimum. This finding gave the idea to use a faster and more advanced optimization tool than gridding, called the Nelder-Mead (NM) method, [Nelder and Mead, 1965], in order to find the minimum in the T_i-T_d plane, see below.

The new algorithm can be summarized by

1. Given a linear transfer function, initial PID parameters are chosen using the AMIGO method.

2. NM optimization finds the PID controller giving the minimum cost function in the T_i-T_d plane.

a) For each T_i-T_d couple, a proportional gain, K, is found such that the constraints are fulfilled.

b) Simulink® simulations are used to calculate IAE-values in the points through which the NM method proceeds.

An interactive program menu has been added to make it possible for the user to change a number of settings in the algorithm as well as for the presentation of the results. When the program is run in MATLAB®, the menu will come up unless the opposite is stated by the user. New default values for the optimization can also be set as input parameter. This is especially useful for batch runs, when you may want to choose the settings before a number of program runs are started. An experienced user should easily be able to modify the program, to for instance, change the optimization method or at least to change the cost function.

4. Algorithm details

In this section, the optimization algorithm will be explained in further detail.

4.1 The Nelder-Mead method

Nelder-Mead optimization belongs to the subclass of optimization methods called direct search methods. The main theme among these is that they only use function values without creating approximations of the function gradients explicitly. These methods are especially useful if, for instance, the cost to evaluate the function is high and if it is impossible to derive the exact gradient. These statements apply to the optimization problem (1). Whenever the cost function is evaluated, the feasible proportional gains must be calculated and Simulink® simulations run. The simulations are particularly costly if the given PID parameters, at a certain grid point, gives a very sluggish closed loop.

The Nelder-Mead method is a simplex-based method. There are many papers and books which describe in detail how the NM algorithm works (see for instance [Walters et al., 1991] and [Lagarias et al., 1998]). Two of the reasons why the method is popular are that it is easy to both understand and implement. It is only necessary to look at two dimensional NM optimization in this paper as (1) can be viewed as an unconstrained minimization problem in $\mathbb{R}^2$, when K is chosen separately. Two dimensional NM optimization can be interpreted as triangle search progression with variable area. In order to begin the NM optimization, an initial triangle has to be specified. The function to be minimized - lets call it f - is evaluated at all three edges and the points are sorted in the order: B (best,

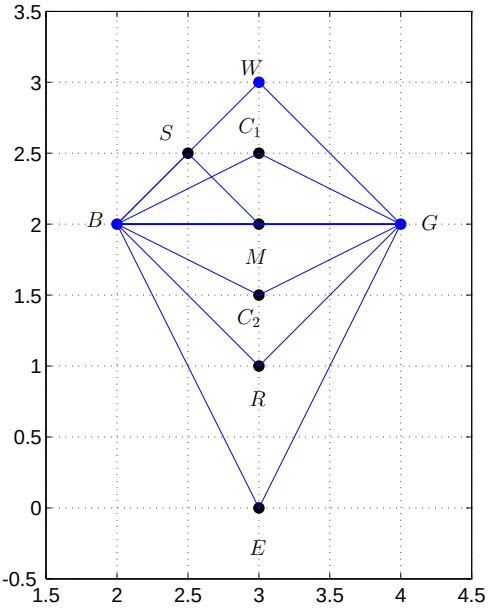

Figure 3. The Nelder-Mead progression in one iteration. The initial simplex is the one with corners in B, G and W. The simplex will change it's shape depending on function evaluations in closely situated points.

lowest function value $f(B)$), W (worst, highest function value $f(W)$) and G (good, function value, $f(G)$, in between the other two). From this point, the algorithm will alter the shape of the triangle to give a new one with less total cost. These steps are well explained in the given sources and will not be presented in further detail here. Figure 3 gives a hint of possible new simplexes. When a new simplex has been determined - again with corners B, G and W - the algorithm will iterate until a termination condition has been fulfilled.

4.2 Initial values

It is preferable to have a good initial guess of where the minimum is located to have fast convergence of the optimization. Another reason is that there is a chance - although not so big when solving (1) - of ending up in a local minimum. The method used to receive an initial controller in the proposed algorithm is called AMIGO, which is a tool for robust PID (and PI) synthesis. To understand AMIGO, it is also important to understand the MIGO method for PID design.

The optimization problem that the MIGO design deals with is very

similar to (1). But instead of minimizing over the IAE-value, it uses the Integrated Error,

$$\mathrm{IE}_{load} = \int_0^\infty e(t)dt,$$

as cost function and the M-circle as robustness constraint, to determine the PID parameters. The IE cost is proportional to $1/k_i = T_i/K$, which reduces the problem to maximizing the k_i-gain over the robustness area.

The AMIGO design is basically a set of formulas yielding K, T_i and T_d. In order to determine these, the MIGO method was run on a large number of systems common in process industry (with $M_s = M_t = 1.4$). Secondly, each and every process in the batch was approximated as a first-order system with time delay (FOTD)

$$G_p(s) = \frac{K_p}{sT+1}e^{-sL}. \tag{2}$$

The PID-parameters were then plotted versus the normalized time delay, $\tau = L/(L+T)$, and parameter fittings were made on these curves resulting in the formulas.

In the proposed PID design method, the system of interest is approximated as a FOTD system, (2), through a step response test and the AMIGO parameters are then determined. Let the index A denote the AMIGO PID parameters. The AMIGO parameters provided are used as one of the corners, (T_d^A, T_i^A), in the initial Nelder-Mead simplex. Taking into account that the evaluation time is usually greater far out in the T_i-T_d plane, the two remaining corners have been set to $(0.4T_d^A, T_i^A)$ and $(T_d^A, 0.4T_i^A)$.

4.3 Determining the proportional gain K

The key idea to find K in every point (T_d,T_i), is to determine all K-values putting the open-loop Nyquist curve on a circle in the complex plane (at every frequency point ω), resulting in a function $K(\omega)$. Since the method is numerical, the frequency span is divided into a finite number of points ω_k, $k = 1, 2, \ldots$. In order to determine $K(\omega)$, let us first assume that the open-loop frequency response, $G_o(i\omega)$, can be written as

$$G_o(i\omega) = K(X(\omega) + iY(\omega)), \tag{3}$$

where $X(\omega)$ and $Y(\omega)$ are the real and imaginary parts of $G_o'(i\omega) = G_o(i\omega)/K$. From the optimization problem (1) we have the constraint

$$|G_o(i\omega) - C|^2 = R^2, \tag{4}$$

for any circle with center in C and radius R. Using (3) and (4), but changing K to $K(\omega)$, will lead to

$$(K(\omega)X(\omega) - C)^2 + (K(\omega)Y(\omega))^2 = R^2 \Rightarrow$$
$$K(\omega)^2 - \frac{2CX(\omega)}{X(\omega)^2 + Y(\omega)^2}K(\omega) + \frac{C^2 - R^2}{X(\omega)^2 + Y(\omega)^2} = 0. \quad (5)$$

The two solutions correspond to the gains for which the open-loop Nyquist curve will cross the front and back side of the circle respectively (see Fig. 4)

$$K_{1,2}(\omega) = \frac{CX(\omega) \pm \sqrt{R^2(X(\omega)^2 + Y(\omega)^2) - C^2Y(\omega)^2)}}{X(\omega)^2 + Y(\omega)^2}. \quad (6)$$

$K_{1,2}(\omega)$ could for instance look like the plots in Fig. 5. For some frequency points, (6) will assume imaginary or negative numbers, which are discarded. In the intervals for which K assumes positive real values, there can be multiple minima and maxima.

There are a few observations needed in order to conclude which K-values will fulfill the constraints in (1).

THEOREM 1
The open-loop Nyquist curve, (3), of an arbitrary controlled process, will be tangent to a circle in the complex plane, given by the center point C and radius R, if and only if

$$\frac{dK_1(\omega)}{d\omega} = 0 \quad \text{or} \quad \frac{dK_2(\omega)}{d\omega} = 0 \quad (7)$$

Proof In vector form, the open-loop frequency response becomes

$$G_o(i\omega) = K \begin{pmatrix} X(\omega) \\ Y(\omega) \end{pmatrix}. \quad (8)$$

There are two conditions that has to be fulfilled in order for the open-loop Nyquist curve to be tangent to the circle at a given frequency point ω^*. The open-loop Nyquist curve should lie on the circle determined by

$$(KX(\omega^*) - C)^2 + (KY(\omega^*))^2 = R^2, \quad (9)$$

while the tangent of the open-loop Nyquist curve and the vector between the center point and Nyquist curve should be orthogonal

$$\left(\frac{dG_o(i\omega^*)}{d\omega}\right)^T \begin{pmatrix} KX(\omega^*) - C \\ KY(\omega^*) \end{pmatrix} = 0. \quad (10)$$

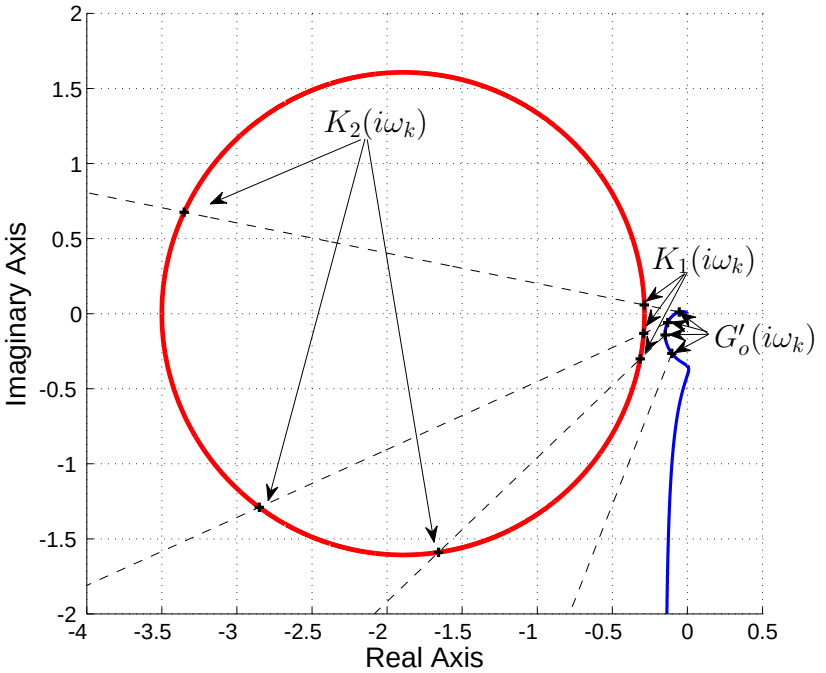

Figure 4. Proportional gain functions $K_1(\omega_k)$, $K_2(\omega_k)$, for which $K_j G'_o(i\omega_k)$, $j \in [1,2]$, is tangent to a circle in the complex plane. $G'_o(i\omega_k)$ is the open-loop frequency response with $K = 1$ and ω_k denotes the discrete frequency points.

Denoting $X'(\omega) = dX(\omega)/d\omega$, $Y'(\omega) = dY(\omega)/d\omega$, (10) can be rewritten as

$$\begin{pmatrix} KX'(\omega^*) \\ KY'(\omega^*) \end{pmatrix}^T \begin{pmatrix} KX(\omega^*) - C \\ KY(\omega^*) \end{pmatrix} = \\ K^2 X(\omega^*)X'(\omega^*) - KCX'(\omega^*) + K^2 Y(\omega^*)Y'(\omega^*) = 0,$$

and in turn, by solving for K, we end up with

$$K = \frac{CX'(\omega^*)}{X(\omega^*)X'(\omega^*) + Y(\omega^*)Y'(\omega^*)}. \tag{11}$$

Let us now go back to Eq. (5). Taking the derivative with respect to ω,

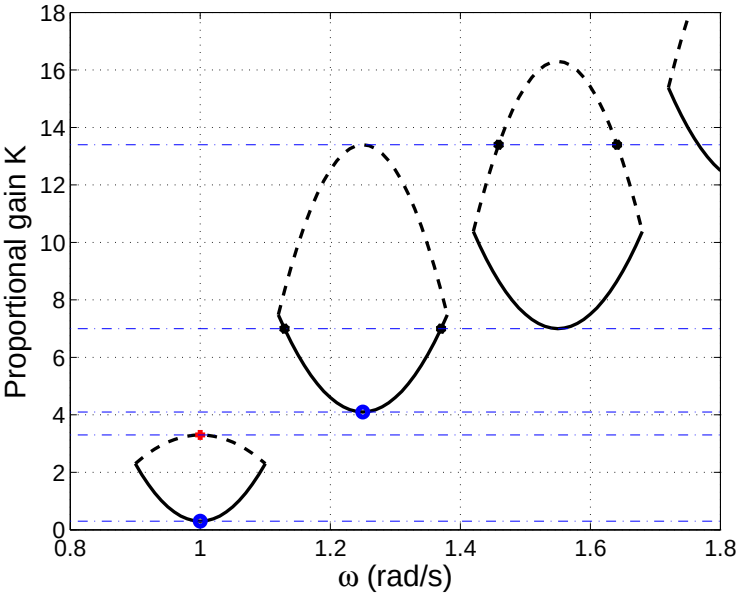

Figure 5. A constructed sketch of how the functions $K_1(\omega)$ (solid) and $K_2(\omega)$ (dashed) could look for a time-delayed system. Only K-values unique in ω - i.e. the first two minima and first maximum in this case - will fulfill the circle constraints in (1).

given that $K'(\omega) = dK(\omega)/d\omega$, leaves us with

$$2KK'X^2 + 2K^2XX' - 2CK'X - 2CKX' + 2KK'Y^2 + 2K^2YY' = 0, \quad (12)$$

with ω omitted. Using $K'(\omega) = 0$, results in

$$K(\omega)X(\omega)X'(\omega) - CX'(\omega) + K(\omega)Y(\omega)Y'(\omega) = 0 \Rightarrow$$
$$K(\omega) = \frac{CX'(\omega)}{X(\omega)X'(\omega) + Y(\omega)Y'(\omega)}, \quad (13)$$

which is identical to (11) in ω^*. Since (9) is fulfilled for all frequencies in $K(\omega)$, the proof is concluded. □

$G_o(i\omega)$ can, however, be tangent to the circle on both the inside or outside, but still have points within (thus giving an infeasible solution). To explain why, it is a good idea to once again view Fig. 4. At a given frequency point ω_k, it is obvious that all proportional gains between $K_1(\omega_k)$

and $K_2(\omega_k)$ will place $G_o(i\omega)$ inside the circle, thus resulting in infeasible K. Looking at Fig. 5, this means that only the minima and maxima, for which K is unique in ω, are feasible. For this particular case, it corresponds to the first two minima and first maximum.

Once all possible K-values have been determined (the two minima and the maximum from Fig. 5 for example), closed-loop stability is evaluated. If there is stability, the optimal proportional gain at a given point in the T_i-T_d plane, is then given by the K resulting in the lowest IAE-value (determined by Simulink® simulations).

Up to now it has been assumed that it is just one circle in the complex plane that the open-loop Nyquist curve may be tangent to. Since the constraints of (1) demands that the Nyquist curve is located outside both the M_s- and M_t-circles, the algorithm has to be run twice.

5. Examples

In this section there will be a few examples highlighting the benefits of the proposed program and algorithm compared to other methods. It will also show that the new method is reliable in many design cases.

EXAMPLE 1—THE AMIGO TEST BATCH

The AMIGO formulas, [Hägglund and Åström, 2004], were derived using MIGO on a test batch, which includes 134 essentially monotone systems common in process industry.

In order to compare with the MIGO PID designs on the batch, the PID design software presented in this paper was modified to use the M-circle as constraint. It took the program just more than one hour to run through all sub-batches except some integrating processes. This gives an average time of 30 seconds per process in the batch. The batch was however run to get a high accuracy on the optimal solution rather than optimized for a fast design. If speed is of essence, the average design time per process could be cut by at least two thirds of the time. The designs was run on an Intel® Dual-Core™, 2.13 GHz with 1 GB RAM and Fedora™ 7 using MATLAB® 7 R2007a. The only two parameters that had to be modified from the default values (depending on the process) in order for the batch to run through properly, were T_f and the frequency grid.

The PID parameters derived by the proposed algorithm were compared to those given by the MIGO method. Since the MIGO method was not derived to minimize IAE, the proposed method should give lower values at all times. The MIGO method is however a good indicator to see if the new algorithm works or not. The batch run showed that the two methods are very similar. In average, the new controllers resulted in IAE values at

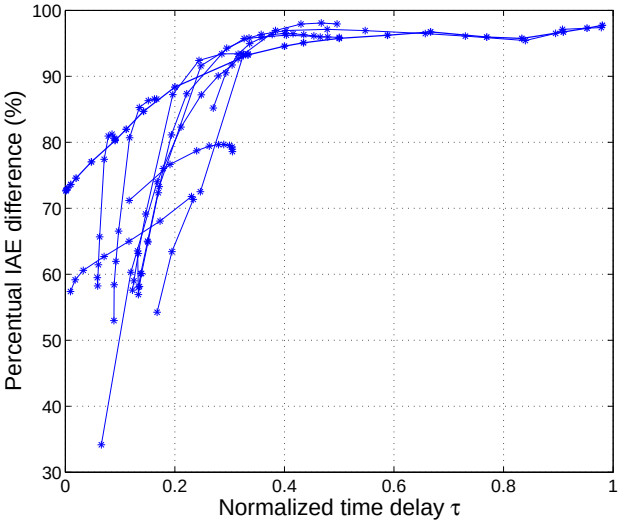

Figure 6. IAE values comparison for the testbatch using the M-circle constraint alone or both the M_s- and M_t-circles. The plot displays $100 \cdot IAE_{M_s,M_t}/IAE_M$ as a function of τ.

95% of what the MIGO controllers gave over the whole batch. This gives both a strong indicator that the new program works properly and that the MIGO method gives essentially IAE minimized controllers for the batch.

To see the benefit of using the M_s- and M_t-circles instead of the M-circle, the whole batch was compared when the two different constraints were used respectively. Figure 6 shows that the biggest percentual gain is given for low values on the normalized time delay, τ, while more delay dominated systems are less dependent on the choice of the constraints. This indicates that the IAE can be decreased a great deal without changing the essential robustness constraints.

The one subbatch where the newly proposed PIDs gave IAE values with the most deviation from the MIGO ones was

$$P(s) = \frac{1}{(s+1)((sT)^2 + 1.4sT + 1)}, \qquad (14)$$
$$\text{with} \quad T = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0.,$$

i.e. processes with complex poles. In particular, when $T = 0.1$, the new IAE is as small as 62.5% of the MIGO IAE, corresponding to a significant improvement. Figure 7 shows the output signal, y, and control signal, u, when a load disturbance, d (see Fig. 1), is acting on this process. The

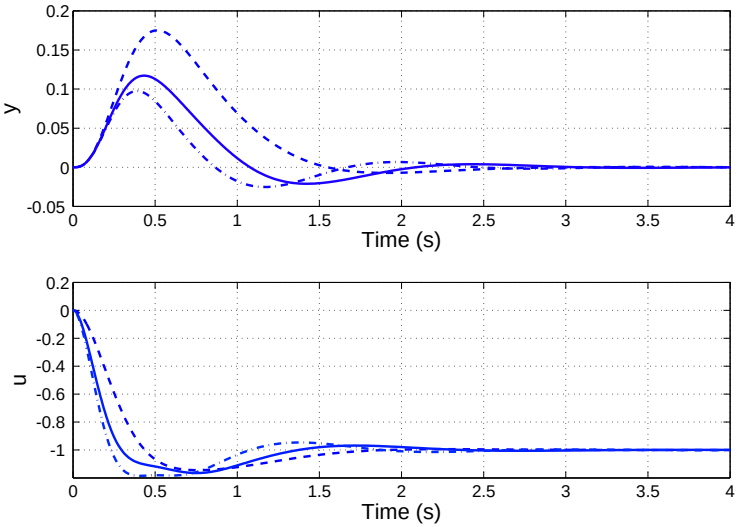

Figure 7. The output (y) and control signal (u), during a load disturbance, for three different designs on (14), $T = 0.1$. Dashed line: MIGO PID; Solid line: Proposed PID with M-circle constraint; Dash-dotted line: Proposed PID with the M_s- and M_t-circle constraints.

dashed curves correspond to the MIGO controller ($K = 3.96$, $T_i = 0.46$, $T_d = 0.08$), the solid lines to the proposed controller with the M-circle constraint ($K = 5.42$, $T_i = 0.29$, $T_d = 0.16$) and the dash-dotted line to the new design method with the M_s- and M_t-constraints ($K = 6.53$, $T_i = 0.22$, $T_d = 0.16$). The open-loop Nyquist curves for the three cases are shown in Fig. 8. It is known that the MIGO method discards solutions that touches the M-circle twice. This example shows that this choice may be overly conservative. It is also evident that the substitution of the M-circle to the M_s- and M_t-circles gives a much lower IAE-value in this case. □

EXAMPLE 2—AN OSCILLATORY PROCESS

Consider an oscillatory system with the linear transfer function

$$P(s) = \frac{9}{(s^2 + s + 9)(s + 1)}, \tag{15}$$

which has two poles with a relative damping of $\zeta = 0.17$. An IE-cost function is not suitable for PID design when the system is oscillatory, ruling

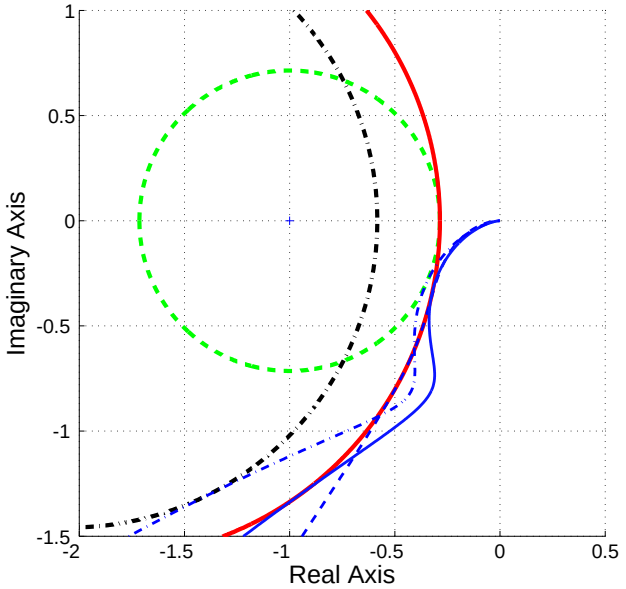

Figure 8. The open-loop Nyquist curves when three different design methods were used on (14), $T = 0.1$. Dashed line: MIGO control; Solid line: Proposed design with M-circle constraint; Dash-dotted line: Proposed design with the M_s- and M_t-circle constraints.

out use of the MIGO method. The proposed design algorithm, however, can derive a PID design without problems. For $M_s = M_t = 1.4$ the program gave the parameters: $K = 0.37$, $T_i = 0.23$, $T_d = 0.80$. Figure 9 shows the control- and output signals. □

6. Conclusions

This paper has presented a new software tool that can help educate people in PID control systems as well as provide them with controller designs in short time. The controller designs are made to minimize the integrated absolute error during a load disturbance on the process input. The optimization is constrained by robustness conditions on two of the sensitivity functions. The finding that a lot of processes only give one unique minimum solution to the optimization problem lead to the use of the Nelder-Mead method. The initial simplex is provided by the AMIGO method, a

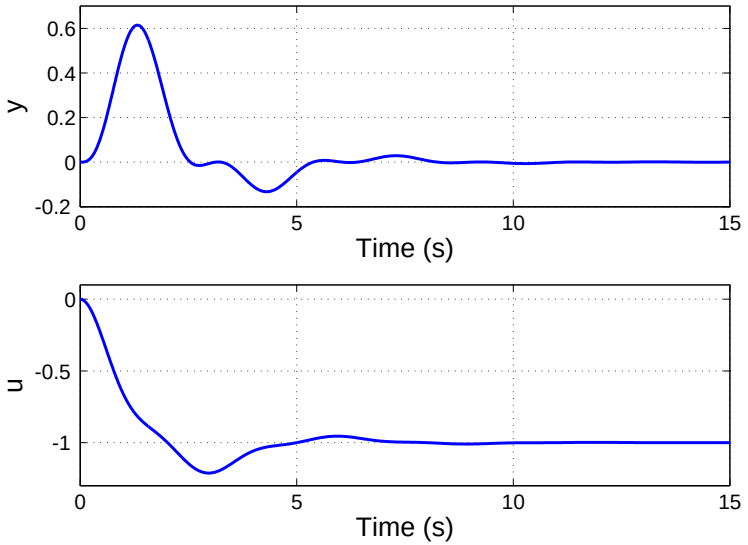

Figure 9. The output signal, y, and control signal, u, when the proposed design method was used to find a controller for the oscillatory process (15). $M_s = M_t = 1.4$.

choice made rather for the speed of the algorithm than it being necessary to find the global minimum.

The software tool was shown to give reasonable controllers on a large batch of processes common in process industry. The use of IAE as cost function also give the possibility to run the program on highly oscillatory systems, as was shown in an example.

Future research should provide a better way of handling the effect of measurement noise on the control signal, providing a sophisticated way of choosing the filter constant T_f. It may also be needed to include even more constraints in the optimization problem in order to, e.g, give robustness to time delay uncertainty. With these modifications it should be possible to use the program for PID controller design on real plants.

Acknowledgements

This work was supported by the Swedish Research Council (VR).

References

Åström, K. J. and T. Hägglund (2005). *Advanced PID Control*. ISA – The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC.

Åström, K. and T. Hägglund (2001). "The future of PID control". *Control Engineering Practice* **9**, pp. 1163–1175.

Hägglund, T. and K. J. Åström (2004). "Revisiting the Ziegler-Nichols step response method for PID control". *Journal of Process Control* **14**:6, pp. 635–650.

Lagarias, J. C., J. A. Reeds, M. H. Wright, and P. E. Wright (1998). "Convergence properties of the Nelder-Mead simplex algorithm in low dimensions". *SIAM Journal on Optimization* **9**, pp. 112–147.

Li, Y., K. Ang, and G. Chong (2006). "PID control system analysis and design - Problems, remedies, and future directions". *IEEE Control Systems Magazine* **26**, pp. 32–41.

Nelder, J. A. and R. Mead (1965). "A simplex method for function minimization". *Computer Journal* **7**, pp. 308–313.

Nordfeldt, P. (2005). *PID Control of TITO Systems*. Licentiate Thesis ISRN LUTFD2/TFRT--3238--SE. Department of Automatic Control, Lund University, Sweden.

Oviedo, J. J. E., T. Boelen, and P. van Overschee (2006). "Robust Advanced PID Control (RaPID) - PID Tuning Based on Engineering Specifications". *IEEE Control Systems Magazine* **26**, pp. 15–19.

Panagopoulos, H., K. J. Åström, and T. Hägglund (2002). "Design of PID controllers based on constrained optimisation". *IEE Proceedings - Control Theory & Applications* **149**:1, pp. 32–40.

Walters, F. H., L. R. Parker Jr, S. L. Morgan, and S. N. Deming (1991). *Sequential Simplex Optimization*. CRC Press LLC.

Paper III

Modeling for optimal PID design

Olof Garpinger Tore Hägglund

Abstract

Even though PID controllers have been around for a long time, few industrial controllers use derivative action and the remaining PI controllers are often designed with formula-based tuning rules rather than through computer-based optimization. This paper will delve into some of the reasons behind these choices and show potential benefits of instead using software-based PID tuning. Three commonly used tuning rules are compared to software tuning with respect to performance and robustness over a large process batch. The study shows the importance of combining a fast, accurate modeling tool with the software design method and gives guidelines for future modeling tools with regards to desired process information. With moderate process knowledge it is possible to design controllers that are much closer to optimal than the three tuning rules, with significant performance improvements as a result.

1. Introduction

The low order of the PID controller is well-suited for use in the process industry where tuning time is of the essence. A good PID tuning method should thus both be fast and easy to carry out for the large number of control loops in a factory. This has led to the great popularity of formula-based tuning rules, which typically need some basic knowledge or model of the process. In [O'Dwyer, 2009], there are 1,730 PI and PID tuning rules collected. We will, however, compare some commonly used tuning rules to computer-driven optimization and argue that there should be at least one more tuning method.

Although the benefits of derivative action are well-known, it is most often turned off in industrial PID controllers. Reasons for this include increased noise sensitivity, variety of controller structure, and the difficulty of tuning 1–2 more parameters including noise filter design. To hand-tune a PID controller quickly is thus rather difficult, and there are no PID tuning rules that have gained wide acceptance in industry. In this paper, we will show the importance of combining the tuning method with a suitable modeling tool, similar to the results by [Leva and Schiavo, 2005]. Together with our software-based design tool we will also show the potential benefits of using such a tuning method both in terms of robustness and performance.

2. Theory

A PI controller is often parametrized in terms of proportional gain K and integral time T_i, while a PID controller also includes the derivative time T_d. In this paper we will mainly consider ideal PI and PID controllers

$$C_{PI}(s) = K\left(1 + \frac{1}{sT_i}\right), \tag{1}$$

$$C_{PID}(s) = K\left(1 + \frac{1}{sT_i} + sT_d\right), \tag{2}$$

without noise filtering.

2.1 Criteria for control comparison

Closed-loop requirements typically include specifications on load disturbance attenuation, robustness to process uncertainty, measurement noise and set-point tracking. Load disturbance attenuation and robustness are primary concerns in process control and will therefore be in focus here when comparing the different tuning methods. The set-point response can

be handled separately, see e.g. [Åström and Hägglund, 2005], and the effect of noise will only be discussed briefly in the end of this paper.

Minimization of the integrated absolute error (IAE)

$$\text{IAE} = \int_0^\infty |e(t)|dt, \tag{3}$$

will define optimal control performance in this paper, where $e(t)$ is the control error due to a unit step load disturbance, $d(t)$, on the process input.

Robustness to process uncertainty can be captured by the sensitivity functions

$$S(i\omega) = \frac{1}{1 + G_l(i\omega)}, \quad T(i\omega) = \frac{G_l(i\omega)}{1 + G_l(i\omega)}, \tag{4}$$

where $G_l(s) = P(s)C(s)$ is the loop transfer function with process $P(s)$ and controller $C(s)$. We will use

$$|S(i\omega)| \leq M_s, \quad |T(i\omega)| \leq M_t, \quad \forall \omega \in \mathbb{R}^+ \tag{5}$$

to constrain IAE optimization, and

$$M_{st} = \max(|S(i\omega)|, |T(i\omega)|), \quad \forall \omega \in \mathbb{R}^+ \tag{6}$$

to provide a robustness measure of the closed-loop system. M_{st} will vary depending on process model and tuning method. Reasonable robustness is given for M_{st} ranging between 1.2–2.0.

2.2 Modeling

Since the modeling time should be as short as possible, it is reasonable to believe we only have time for quick experiments that provides limited process knowledge. We will therefore assume that our models are of low order, either a first-order time-delayed (FOTD) system

$$P_m(s) = \frac{K_p}{sT + 1} e^{-sL}, \tag{7}$$

or a second-order time-delayed (SOTD) model

$$P_m(s) = \frac{K_p}{(sT_1 + 1)(sT_2 + 1)} e^{-sL}, \tag{8}$$

with the special case $T_1 = T_2$. Processes can be characterized based on the normalized time delay $\tau = L/(L + T)$ (FOTD) or $\tau = L/(L + T_1 + T_2)$ (SOTD), ranging from 0 to 1. A process is lag-dominated if τ is small, delay-dominated if τ is large, and balanced if τ is around 0.5.

A common way to determine K_p, L and T in (7) is based on an open-loop step response of the process. K_p is the steady state gain. The apparent time delay L is the t-coordinate of the intersection of the steepest tangent with the time axis, and $L + T$ is the time when the step response has reached 63% of its steady state value. We call this method the 63%-rule.

Another way to determine either an FOTD or SOTD model is through reduction of a higher-order process model with the so called half-rule, see [Skogestad, 2003].

A relay test is made in closed loop where the control signal switches amplitude whenever the process output crosses a certain hysteresis threshold. This method is less sensitive to disturbances than the step test and keeps the process closer to its set-point during the modeling experiment. However, it typically only gives information about one frequency point in the process spectrum and it is seldom used for deriving models like (7) and (8).

2.3 Tuning methods

We have chosen to compare our own software-based tuning method with three commonly used tuning rules: Lambda tuning; SIMC; and AMIGO.

Lambda tuning Lambda tuning is today widely adopted in the process industry, see e.g. [Sell, 1995]. Modeling is typically based on measured step responses and the 63%-rule is used to obtain an FOTD model. The desired closed-loop time constant T_{cl} is used as a tuning parameter, for which we have used the classic choice $T_{cl} = T$ in this paper even though there are better recommendations for delay-dominated processes. Lambda tuning does not refer to any specific robustness, but here we have chosen to compare it to IAE-optimal controllers with $M_s = M_t = 1.4$.

SIMC [Skogestad, 2003] introduced modifications of the Lambda tuning method called SIMC, that improves performance especially for lag-dominant processes. An FOTD (PI) or SOTD model (PID) is obtained by model reduction using the half-rule. SIMC is closely related to Lambda tuning, but uses the desired closed-loop time constant $T_{cl} = L$, which typically gives a sensitivity close to $M_{st} = 1.6$.

A modified method for PI control, here called SIMC+, was presented in [Skogestad and Grimholt, 2012] to improve performance for delay-dominated systems. For PI control we therefore use SIMC+ and for PID control we use the original SIMC rule.

AMIGO The AMIGO method, [Hägglund and Åström, 2004], was obtained by applying constrained optimization to a large test batch of process models and then use parameter fitting to find the tuning rules. The

parameters of an FOTD model are determined by the 63%-rule. The controller is tuned for a robustness of $M_s = M_t = 1.4$.

SWORD Our own Software-based optimal robust design (SWORD) of PI and PID controllers was first introduced in [Garpinger and Hägglund, 2008]. Using a linear process model of any order and any robustness constraints on the sensitivity and complementary sensitivity functions, one can find the IAE-optimal controller. Here, we will choose $M_s = M_t$ for simplicity. The user can also specify a first (PI) or second-order (PID) measurement noise filter before the optimization. This can then be used to set an upper limit for the control signal activity due to measurement noise, as shown in [Garpinger, 2009].

3. Comparison of the tuning methods

3.1 Approach

It is reasonable to believe that the four tuning methods would be used together with the 63%-rule in practice since the step response test is the most common modeling experiment in industry. SIMC and SWORD will also be compared when used on perfect process models, which means that SWORD will use an exact model of the process while SIMC will use models derived with the half-rule from the exact model. Given the need for modeling speed, however, it is unlikely that one would have access to an accurate model in every process case.

The four tuning methods will be compared with respect to IAE and M_{st} for the batch of processes common in process industry, that was presented in [Hägglund and Åström, 2004] and used to derive the AMIGO rules. The integrating processes in the batch are left out of our study since the Lambda tuning method does not handle such systems.

For each process in the batch, we have derived one 63%-rule FOTD model to use with all tuning methods, as well as half-rule FOTD and SOTD models to also use with the SIMC method. PI and PID controllers were derived based on these models, after which M_{st} and IAE were derived with respect to the nominal process. The IAE-values were compared with the PI and PID controllers giving minimal IAE = IAE_{opt} with $M_s = M_t = 1.4$ for Lambda tuning, AMIGO, SWORD and $M_s = M_t = 1.6$ for the SIMC-methods. The measure $100 \cdot \text{IAE}/\text{IAE}_{opt}$ was used to compare performance to the optimal (100%).

3.2 Comparison

The results from the comparison of PI controllers are collected in Fig. 1. The variation in both closed-loop robustness and performance is large for

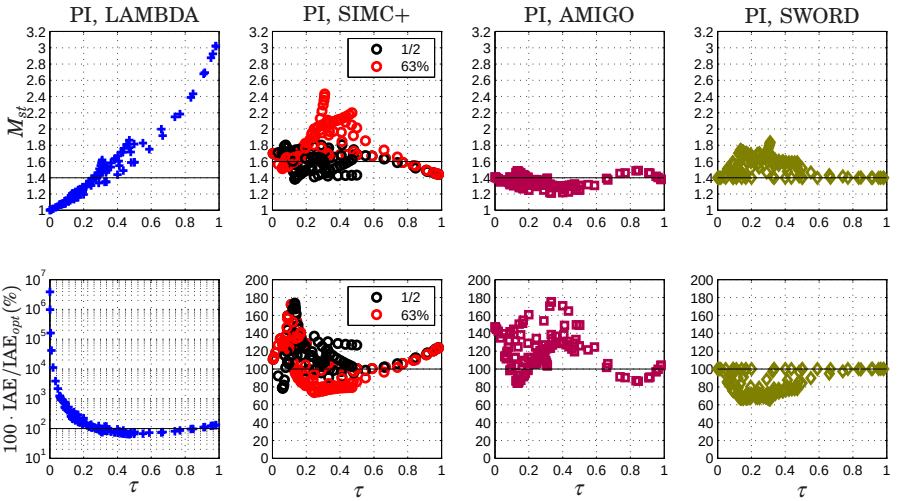

Figure 1. Comparison of the four different tuning methods for PI control. The upper plots compare robustness with respect to the nominal process. The lower plots compare nominal closed-loop performance to optimal performance given a robustness associated with the specific methods. 1/2 denotes controllers derived from half-rule models and 63% denotes controllers given by 63%-rule models. Notice the log-scaled performance plot for the Lambda method.

the Lambda method, even if we disregard delay-dominant processes. On the other hand, it seems quite easy to predict both of them if the normalized time delay, τ, is known. If SIMC+ is used together with the half-rule (1/2), the robustness will vary roughly between 1.4 and 1.8. Assuming use of 63%-rule (63%) models instead, the robustness will vary between 1.45 and 2.45, resulting in poor robustness for quite a few processes. The variation in the performance of the SIMC+ method, on the other hand, does not depend that much on the modeling method. Even though the AMIGO method does not need as advanced models as SIMC+, the robustness varies less, between $1.2 \leq M_{st} \leq 1.5$ with performance on par with SIMC+. On the other hand, AMIGO does not come with a tuning parameter like SIMC+, Lambda tuning and SWORD, which means that one can not trade robustness for better performance and vice versa. Using SWORD with a perfect process model gives controllers that are exactly as good as the optimal controllers. However, if 63%-rule models are used, the robustness will instead vary between 1.4 and 1.85. Notice that there is a clear correlation between loss in robustness and gain in performance.

The results from the PID controller comparison are collected in Fig. 2. Lambda tuning results in poor performance for $\tau < 0.3$ and poor robust-

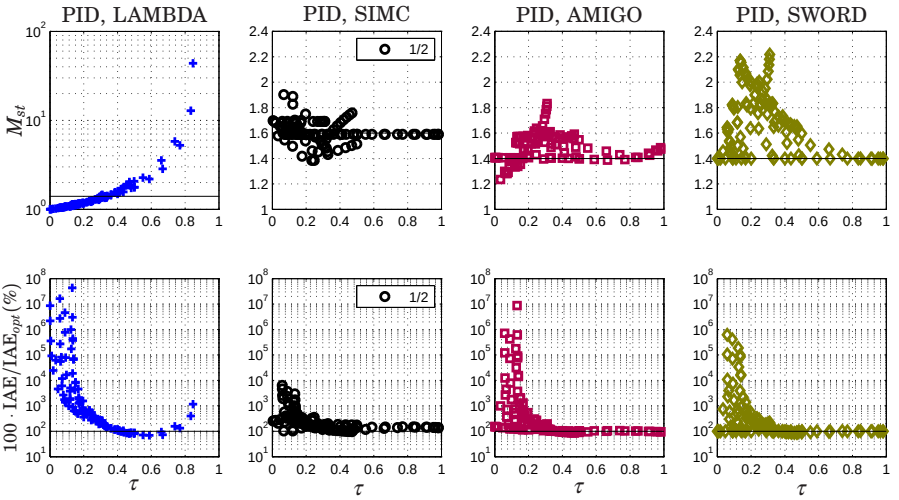

Figure 2. Comparison of the four different tuning methods for PID control. The upper plots compare robustness with respect to the nominal process. The lower plots compare nominal closed-loop performance to optimal performance given a robustness associated with the specific methods. 1/2 denotes controllers derived from half-rule models. Notice that several plots have log-scales.

ness for $\tau > 0.5$. Since SIMC needs an SOTD model to work, we have only used the half-rule for the comparison. The spread in both robustness and performance is on par or better than AMIGO, but the need for a good model is still very limiting for this method. For most processes, the robustness of the AMIGO method is within ± 0.2 from the design values $M_s = M_t = 1.4$. Performance is good for $\tau > 0.3$, but almost as widespread as the Lambda method for $\tau \leq 0.3$. SWORD is obviously in need of a different modeling method than the 63%-rule.

3.3 Visions for better tuning methods

The comparison shows that there is a great deal of variation in both robustness and performance for all four tuning methods. PI control can be improved considerably and it is easy to understand why people in industry hesitate to use PID control. Lambda tuning is intuitive and easy to use, but varies too much in quality. The SIMC methods and SWORD needs too accurate models to work properly and, while AMIGO is the best out of the four tuning methods it still lacks a tuning variable. Clearly, there is room for an improved tuning method.

A properly working SWORD method, with M_{st} close to the design values and almost optimal IAE, would have great benefits. One could

use $M_s = M_t$ as a tuning variable and get much better control performance than the other methods given the same maximum value of M_{st}. The biggest challenge is to find a fast, robust and simple modeling tool that provides good enough models for the tuning method to work. Step response modeling seems to limit the four tuning methods and we will therefore investigate possibilities to use relay modeling instead. The aim is to handle tuning with robustness constraints from $M_s = M_t = 1.4$ to 1.8 and provide guidelines for autotuning of PI and PID controllers.

4. Model quality

Ideally, we would like a process model that preserves closed-loop robustness as well as performance. For simplicity, we will focus on robustness in this article and hope that good performance follows. We would thus like our models to be as accurate as possible around the frequency for which M_{st} is given with optimal control.

Assume that our relay test can give us process knowledge around a single phase angle, $\phi°$, of the process, which should it be? Say that we derive FOTD models (7) and SOTD models (8), with $T_1 = T_2$, using exact process information about the static process gain, K_p, and around the phase ϕ. The static gain is only used to simplify the modeling and we would have preferred if the model was based only on information around ϕ. To see how important the static gain information is, we have also investigated models with a 10% static gain error, $P_m(0) = 1.1K_p$, and found little to no difference in the results. Therefore, the rest of the study will assume perfect knowledge about the static gain, K_p. Such FOTD and SOTD models were derived for phase angles $\phi = -105, -110, ..., -250, -255°$ on a representative subset of the process batch and SWORD was used to obtain IAE-optimal PI and PID controllers for each model with the design values $M_s = M_t = 1.4$. The closed-loop robustness M_{st} was calculated for each relay-based model and the intervals of phase angles for which $1.35 \leq M_{st} \leq 1.45$, were noted. Figure 3 shows these intervals for PI and PID control. For PI control, only FOTD models were used and for PID control SOTD models, with $T_1 = T_2$, were used for all processes except for the FOTD processes. The red circles in the plots show the largest phase angle, within the range of investigated ϕ, that satisfies the given robustness interval, while the blue crosses indicate the least phase angle. All process models within this interval will thus also satisfy the robustness interval. For PI control, this means that all process models based on phase angles between at least $-105°$ and $-130°$ will give accurate closed-loop robustness. For PID control, the dependence is more complex, but prior knowledge of τ can help.

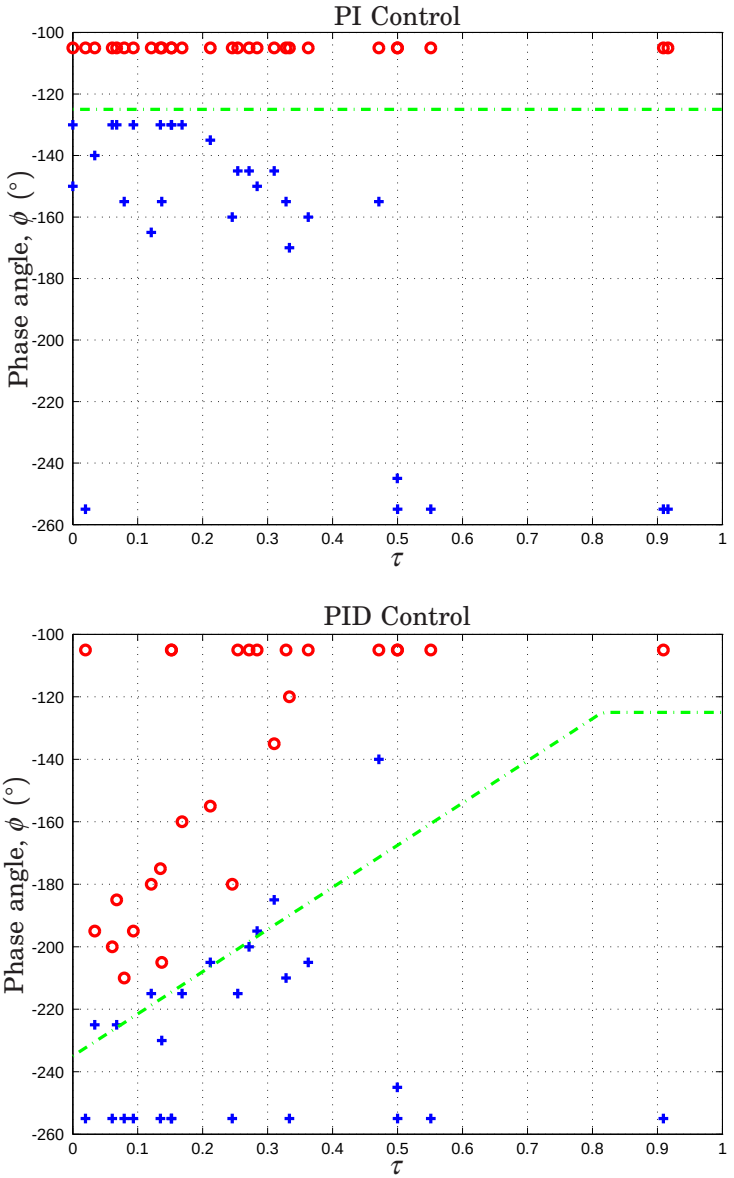

Figure 3. The plots (PI upper, PID lower) show phase angle intervals between the blue crosses (lower boundary) and red circles (upper boundary) for which the phase angle models need to be accurate to preserve M_{st}. The green dash-dotted lines show reasonable phase angle functions $\phi(\tau)$.

Given the information from the plots, we want process knowledge somewhere around the phase angles

$$\phi(\tau) = -125^\circ, \quad \tau \in [0, 1] \tag{9}$$

for PI control and

$$\phi(\tau) = \min(135\tau - 235, -125)^\circ, \quad \tau \in [0, 1] \tag{10}$$

for PID control. These functions are plotted as green, dash-dotted, lines in Fig. 3. The reason why the functions are closer to the lower boundary (crosses) than the upper (circles) is because we want our tuning method to work for M_s- and M_t-values larger than 1.4. Such closed-loop systems will typically have greater bandwidth and should thus use lower values of ϕ. In the next Section, we will show that these two choices of functions are reasonable.

5. Results

Equation (9) was used to determine relay FOTD models for the whole process batch in the same way modeling was carried out in Section 4. PI controllers with $M_s = M_t = 1.4, 1.6$ and 1.8 were then determined through SWORD and compared with IAE-optimal PI controllers for the same robustness values. The results are plotted in Fig. 4 and show that the choice of the phase angle ϕ is almost perfect for PI control with $M_s = M_t = 1.6$. The robustness varies between $1.58 \leq M_{st} \leq 1.62$ and the performance is within 10% higher than the optimum. For $M_s = M_t = 1.4$ the performance variation is the same, but M_{st} varies between $1.4 \leq M_{st} \leq 1.45$. The design choice of $M_s = M_t = 1.8$, will also have reasonable variations with $1.72 \leq M_{st} \leq 1.8$ and IAE less than 30% worse than optimum. Notice that for $M_s = M_t = 1.4$ we have slightly more aggressive controllers than optimum, while for $M_s = M_t = 1.8$ we are more conservative.

PID control was handled in the same way as PI control, but with equation (10) and SOTD models ($T_1 = T_2$) for all processes except the FOTD processes. The results are shown in Fig. 5. For the design choice $M_s = M_t = 1.4$ robustness varies between $1.37 \leq M_{st} \leq 1.53$ and IAE between $90 - 165\%$ of the optimal. The corresponding values for $M_s = M_t = 1.6$ are $1.57 \leq M_{st} \leq 1.80$ and $80 - 165\%$, and for $M_s = M_t = 1.8$ they are $1.76 \leq M_{st} \leq 2.04$ and 75–180%. The robustness variation is thus almost the same in all three cases while performance variation is greater for higher values of M_s and M_t. Thus, unlike PI control, both robustness and performance deteriorates at the same time. Even so, the robustness is kept within the boundaries for decent robustness $1.2 \leq M_{st} \leq 2.0$ for all cases except one.

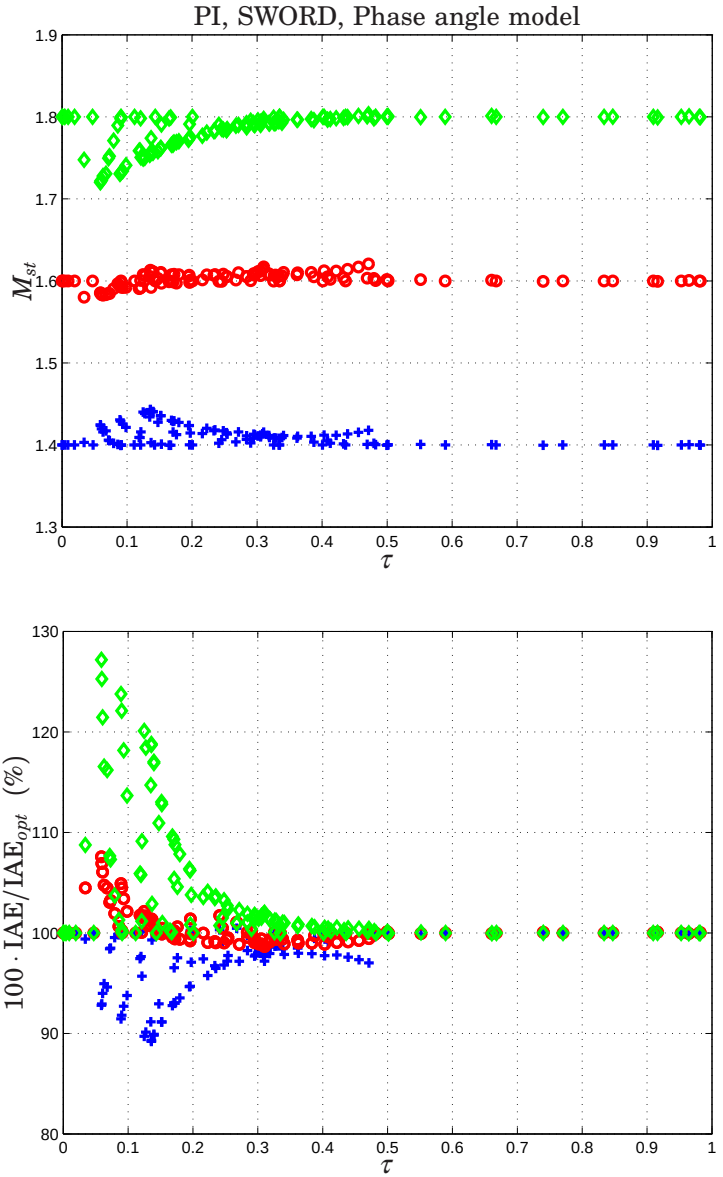

Figure 4. Results in terms of robustness (upper plot) and performance (lower plot) when using SWORD to design PI controllers for the process batch with three design choices, $M_s = M_t = 1.4$ (blue crosses), 1.6 (red circles), 1.8 (green diamonds) on phase angle models derived using process knowledge given by (9).

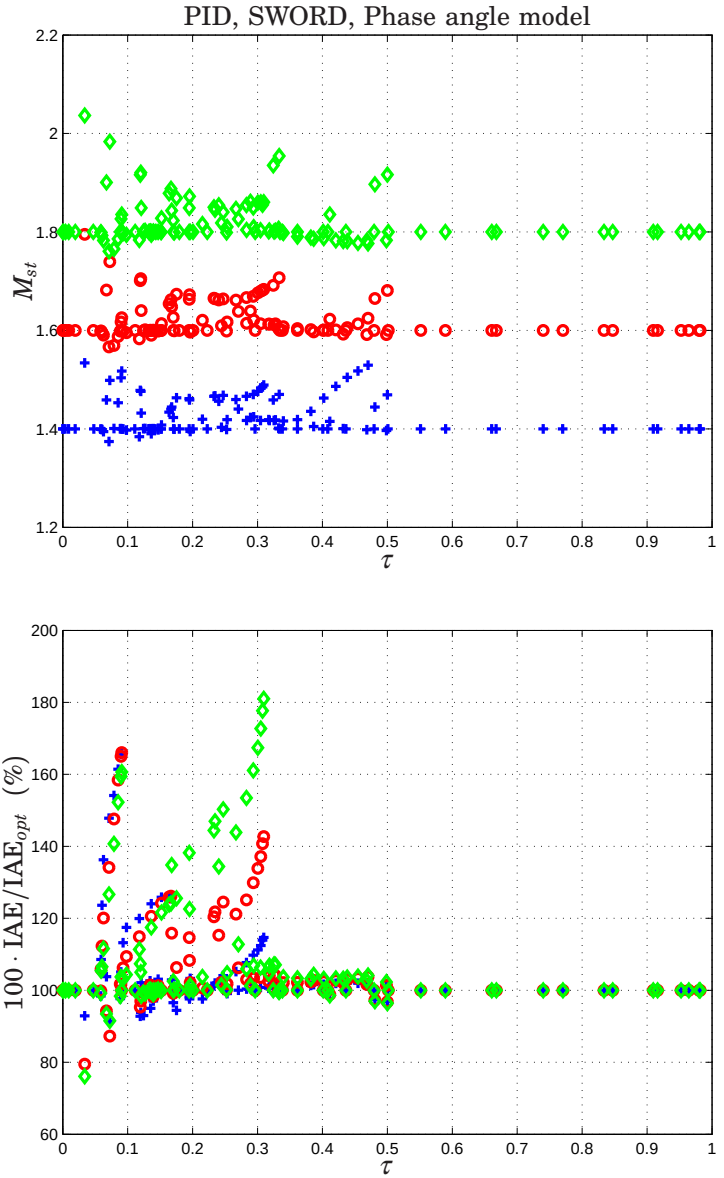

Figure 5. Results in terms of robustness (upper plot) and performance (lower plot) when using SWORD to design PID controllers for the process batch with three design choices, $M_s = M_t = 1.4$ (blue crosses), 1.6 (red circles), 1.8 (green diamonds) on phase angle models derived using process knowledge given by (10).

6. Conclusions

The comparison of the four tuning methods showed some severe shortcomings. For PI control, closed-loop robustness and performance varies a lot, especially for the Lambda method. The SIMC and SWORD methods need accurate models to work well and the AMIGO method lacks a tuning parameter. Furthermore, none of the methods give satisfactory PID control since the performance varies too much.

The biggest benefit of finding a method with less robustness variation is that an increase in M_s and M_t will still guarantee the same M_{st} as the other methods and at the same time improve the performance. Accuracy in performance will of course add further to this. We have focused our study on a software-based tuning method because it can easily adapt itself directly to the process when trying to find the optimal controller. Finding a good tuning rule is hard because it needs to describe every possible case, which is a difficult task especially for PID control. With the optimization software one can also use robustness as a tuning variable. Improving the robustness will thus give worse performance and vice versa, which makes it possible to trade one for the other directly and still guarantee good enough robustness. It is thus our belief that a robust software optimization tool is the future for PI and PID tuning, the question is just how it needs to be built to work properly.

Even if a really good PID software tool is available, it is imperative that it is combined with a fast modeling method that provides good enough models. In this paper, we have shown that the amount of process knowledge needed for both less robustness and performance variation is quite modest. An FOTD model (7) accurate around the phase $\phi = -125°$, with decent static gain knowledge, is enough to provide PI control very close to optimum when used together with SWORD tuning on the whole process batch. PID controller tuning is more complex since SOTD models (8) are needed and because the necessary process knowledge depends on the normalized time delay, τ. Adding a noise filter after the process modeling will also alter τ, thus posing even greater demands on model accuracy. Finding a tuning method that works for both PI and PID control will also present a challenge since the suggested phase angles are different for the two choices.

We have suggested use of relay-based modeling even though there is little research done on relay methods for transfer function modeling. Work by [Friman and Waller, 1997] as well as [Soltesz and Hägglund, 2011], however, suggest that it should be possible to concentrate the relay tests around the suggested phase angles by use of alternative strategies. One important advantage of the relay test to other more advanced modeling methods is that it is already implemented in many commercial control

systems and thus readily used.

The main purpose of this article has been to show the potential for future tuning methods rather than to present a method ready to use. SWORD is our choice of design tool, but the ideas can be used together with any other software-based tuning method. It may even provide guidelines for making better tuning rules for those who wish to continue on that track. No matter the method, however, we think that the key to develop a really good tuning method is to combine both modeling and design in the research and find balance in model accuracy and tuning speed.

Acknowledgements

This work was partly funded by the Swedish Foundation for Strategic Research through the PICLU center. The authors are members of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University.

References

Åström, K. J. and T. Hägglund (2005). *Advanced PID Control*. ISA – The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC.

Friman, M. and K. V. Waller (1997). "A two-channel relay for autotuning". *Industrial and Engineering Chemistry Research* **36**:7, pp. 2662–2671.

Garpinger, O. and T. Hägglund (2008). "A Software Tool for Robust PID Design". In: *17th IFAC World Congress*. Seoul, South Korea.

Garpinger, O. (2009). *Design of Robust PID Controllers with Constrained Control Signal Activity*. Licentiate Thesis ISRN LUTFD2/TFRT--3245--SE. Department of Automatic Control, Lund University, Sweden.

Hägglund, T. and K. J. Åström (2004). "Revisiting the Ziegler-Nichols step response method for PID control". *Journal of Process Control* **14**:6, pp. 635–650.

Leva, A. and F. Schiavo (2005). "On the role of the process model in model-based autotuning". In: *16th IFAC World Congress*. Praha, Czech Republic.

O'Dwyer, A. (2009). *Handbook of PI and PID Controller Tuning Rules*. 3rd Edition. Imperial College Press, London.

Sell, N. J., (Ed.) (1995). *Process Control Fundamentals for the Pulp & Paper Industry*. Tappi Press, Technology Park, Atlanta, GA.

Skogestad, S. (2003). "Simple analytic rules for model reduction and PID controller tuning". *Journal of Process Control* **13**:4, pp. 291–309.

Skogestad, S. and C. Grimholt (2012). "The SIMC Method for Smooth PID Controller Tuning". In: Vilanova, R. et al. (Eds.). *PID Control in the Third Millenium*. Springer-Verlag, London. Chap. 5, pp. 147–175.

Soltesz, K. and T. Hägglund (2011). "Extending the relay feedback experiment". In: *18th IFAC World Congress*. Milano, Italy.

Paper IV

Software-based optimal PID design with robustness and noise sensitivity constraints

Olof Garpinger Tore Hägglund

Abstract

Even though PID control has been available for a long time, there are still no tuning methods including derivative action that have gained wide acceptance in industry. Also, there is still no general consensus for when one should use PID, PI or even I control on a process. The focus of this article is to present a new method for optimal PID control design that automatically picks the best controller type for the process at hand. The proposed PID design procedure uses a software-based method to find controllers with optimal or near optimal load disturbance response subject to robustness and noise sensitivity constraints. It is shown that the optimal controller type depends on maximum allowed noise sensitivity as well as process dynamics. The design procedure thus results in a set of PID, PI and I controllers with different noise filters that the user can switch between to reach an acceptable control signal activity. The software is also used to compare PI and PID control performance with equivalent noise sensitivity and robustness over a large batch of processes representative for the process industry. This can be used to show how much a particular process benefits from using the derivative part.

Submitted to *Journal of Process Control*.

1. Introduction

The derivative part (D-part) of the proportional integral derivative (PID) controller has been available for a long time, but most process control applications still only exploit proportional and integral action. However, process control experts like Shinskey [Shinskey, 1996, p 105], Isaksson and Graebe [Isaksson and Graebe, 2002] agree that the derivative part can add considerable value in many control applications. [Isaksson and Graebe, 2002] lists several reasons why the D-part is seldom used in industry:

- The D-part can lead to extensive control signal activity, i.e. high noise sensitivity.
- The many ways in which the PID controller can be implemented must be matched with the parameters given from the PID design method.
- The lack of a simple four-parameter controller design method that determines both the noise filter parameter and the controller parameters.

In [Garpinger and Hägglund, 2014] it is also pointed out that:

- High performing PID control requires better models than PI control.

First-order time-delayed models derived from step response tests were for example shown to be especially unsuitable for the design of PID controllers on lag-dominated processes and could simultaneously lead to poor robustness and performance.

The most common way to handle noise sensitivity in industry is to low-pass filter either the derivative part alone or the whole measurement signal. The filter is typically designed either before or after the modeling experiments and controller design. Both approaches have their disadvantages. With the first method it is difficult to decide on a reasonable filter without knowing how it affects the closed-loop dynamics. In the latter method the final controller will be different from the one designed, which could lead to worse robustness and performance than intended. As [Isaksson and Graebe, 2002] points out, the filter should ideally be designed at the same time as the PID controller parameters. However, there are still few methods that support such an approach and none of these have gained wide acceptance in industry. This is also one of the reasons why the D-part is so seldom used. In this article we will argue that software-based optimal PID design has several advantages over commonly used tuning rules like Lambda and IMC tuning [Dahlin, 1968; Rivera et al., 1986],

especially when designing the D-part and low-pass filter. In Section 7 we summarize both advantages and disadvantages of both approaches with respect to the results presented in this article.

Several studies, like [Åström et al., 1998; Panagopoulos et al., 2002; Hast et al., 2013; Alfaro and Vilanova, 2013], exploit constrained optimization to determine the PID parameters, but their main objective was not to explore the noise sensitivity problem. Other studies like [Nordfeldt and Hägglund, 2006; Oviedo et al., 2006; Harmse et al., 2009; Sadeghpour et al., 2012], present software tools that can be readily used for constrained optimization of the PID parameters, but they too are little concerned with the noise sensitivity problem. In [Romero Segovia et al., 2014a; Romero Segovia et al., 2014b] second-order measurement filters are designed in combination with three common tuning rules for PID control. The authors relate trade-offs for performance, robustness and noise sensitivity to find tuning rules such that the filter has little influence on the original, unfiltered, control performance and robustness. A similar method is presented in [Leva and Maggio, 2011] where ideal PID parameters, given by an arbitrary design method, are converted into a PID with derivative filter. The filter is chosen in relation to closed-loop cut-off frequency or high-frequency gain such that the effect on nominal performance and robustness is limited. However, neither of these methods [Romero Segovia et al., 2014a; Romero Segovia et al., 2014b; Leva and Maggio, 2011] use controller optimization. The methods presented in [Kristiansson and Lennartson, 2002; Kristiansson and Lennartson, 2006], on the other hand, use constrained optimization to find several tuning rules for both the PID parameters and the noise filter. They too choose the filter time constant such that performance is barely affected. Studies by [Šekara and Mataušek, 2009; Larsson and Hägglund, 2011; Micić and Mataušek, 2014] show how constrained optimization can be used to design PID controllers and noise filters of different orders. These methods only describe how the optimization can be carried out, they do not include derivation of tuning rules nor do they present any software tools for solving their respective optimization problems.

In this article we will base our method for design of PID controllers and noise filters on the one presented in [Garpinger, 2009]. The original idea uses the MATLAB® toolbox presented in [Garpinger and Hägglund, 2008] to solve a constrained optimization problem for design of robust PID controllers. The noise filter time constant is fixed each time a new controller is designed and the closed-loop noise sensitivity is constrained through repeated PID design. In this method there is no need to keep the performance close to the unfiltered, nominal, case. An increase of the filter time constant does not affect the robustness like it would in the earlier mentioned methods [Romero Segovia et al., 2014a; Romero Segovia

et al., 2014b; Leva and Maggio, 2011; Kristiansson and Lennartson, 2002; Kristiansson and Lennartson, 2006]. Noise sensitivity is measured by the gain in variance from measurement noise to control signal and Simulink® simulations are used to calculate this measure by use of process noise data. In this study we will determine a set of PID, PI and I controllers that the user can switch between to select the best performing controller that still gives an acceptable level of control signal variation. Noise sensitivity is thus only defined by the variation of the control signal such that there is no need to collect noise data nor to model the noise in any other way. The user can thus find a preferred controller only through use of the software and visual feedback of the control signal activity. In this article, we also show that the preferred controller type follows naturally from the trade-off between performance and noise sensitivity, given a fixed robustness level. While the optimization toolbox solves an optimization problem that is not concerned with noise sensitivity, we show that repeated PID design give optimal or near optimal controllers also with noise sensitivity in mind. The noise sensitivity for which the transition from PID to PI control occurs is shown to depend on the process dynamics. By plotting the relative performance between PID and PI control for different noise sensitivities, we can also reveal how much or how little a process benefits from using derivative action.

2. Theoretical background

2.1 The closed-loop system

The closed-loop system in Fig. 1 will be used to set up a constrained optimization problem for design of PID controllers. The process, $P(s)$, is manipulated by a controller, $C(s)$, such that the controlled variable, z, is kept as close as possible to a set-point, r, in order to minimize the control error, e. The process is affected by a load disturbance, d, on the process input. The measurements of the controlled variable, y, typically contain noise, n, and are fed through a low-pass filter, $F(s)$, to keep the noise sensitivity of the control signal, u, low.

The choice of letting the load disturbance act on the process input is supported by e.g. [Shinskey, 1996, p 5] that claims that this is the usual case in process control. For research on the output disturbance case, see e.g. [Alcántara et al., 2013] and [Garpinger et al., 2014].

2.2 Controller transfer functions

While some process control applications benefit less from the integrator part, we have chosen to focus on the majority that does. Therefore, we

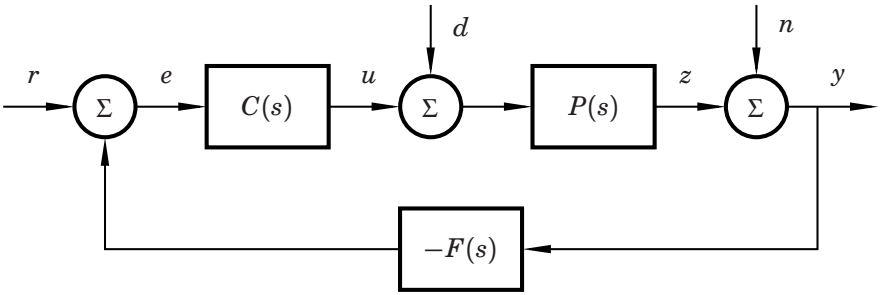

Figure 1. A load disturbance, d, measurement noise, n, and set-point, r, act on the closed-loop system with process $P(s)$, PID controller $C(s)$ and measurement filter $F(s)$.

will consider I, PI and PID controllers in this article. The I controller

$$C_I(s) = \frac{k_i}{s} \tag{1}$$

is the easiest to characterize since it only has one parameter, the integral gain k_i. This is also the simplest controller that will ensure zero steady state error after a step load disturbance d. The PI controller

$$C_{PI}(s) = K \cdot \left(1 + \frac{1}{T_i s}\right) \tag{2}$$

has two parameters, the proportional gain K and the integral time T_i. We have chosen to study the parallel form of the PID controller

$$C_{PID}(s) = K \cdot \left(1 + \frac{1}{T_i s} + sT_d\right), \tag{3}$$

which adds a derivative part to the PI controller with the derivative time T_d. The series form is another parametrization of the PID controller which is convenient for design based on lead-lag compensation, see e.g. [Franklin et al., 2010]. The parallel form is, however, more general since the controller can have complex zeros. Hägglund and Åström [Hägglund and Åström, 2004] among others have previously shown that this is preferred for many processes. For conversion formulas between some common PID controller forms, see [Alfaro and Vilanova, 2012].

The low-pass filter is an important component of the PID controller since the derivative part is very noise sensitive. There are several ways in which filtering can be implemented, but it is common practice to use filters of order one either on the derivative part alone or on the whole measurement signal, see e.g. [Isaksson and Graebe, 2002; Sadeghpour

et al., 2012; Leva and Maggio, 2011; Kristiansson and Lennartson, 2002; Šekara and Mataušek, 2009]. An advantage with the measurement signal filter is that one can design controllers for the combination $P(s)F(s)$. We have thus chosen this approach, but decided to use a second-order filter

$$F_{PID}(s) = \frac{1}{(sT_f)^2/2 + sT_f + 1}, \tag{4}$$

for PID control in order to guarantee amplitude roll-off for high frequencies. Some other studies [Larsson and Hägglund, 2011; Micić and Mataušek, 2014; Garpinger, 2009; Romero Segovia et al., 2013] also explore higher-order low-pass filters for PID control. For PI control we have chosen to use a first-order filter

$$F_{PI}(s) = \frac{1}{sT_f + 1}, \tag{5}$$

also for the sake of high-frequency roll-off. The filter time constant T_f is the only parameter that needs to be set in both F_{PI} and F_{PID}. They would have been more general with two or more parameters, but this form was chosen to keep the amount of controller parameters as low as possible. F_{PID} has two complex poles with relative damping $\zeta = 1/\sqrt{2}$, which is the smallest damping ratio for which there is no amplitude increase caused by the filter. Larsson and Hägglund [Larsson and Hägglund, 2011] showed that the filters F_{PI} and F_{PID} are well suited for the closed-loop system in Fig. 1 and that filters of lower and higher order are not likely to give any performance benefits for equivalent noise sensitivity when using white Gaussian noise. I control has natural roll-off, so $F_I(s) = 1$.

2.3 Constrained optimization for PID design

Closed-loop requirements typically include specifications on load disturbance attenuation, set-point tracking, robustness to process uncertainty and measurement noise sensitivity. Set-point and load disturbance control can be handled separately as shown in e.g. [Åström and Hägglund, 2005]. Furthermore, Shinskey [Shinskey, 1996, p 11] indicates that most set-points remain constant in continuous process control. For these two reasons we have chosen to focus on load disturbance attenuation, robustness and noise sensitivity in this article, leaving the set-point $r = 0$.

We choose to minimize the integrated absolute error (IAE)

$$\text{IAE} = \int_0^\infty |e(t)|dt \tag{6}$$

during a unit step load disturbance, to optimize closed-loop performance. The variable t denotes time in [s] and the IAE-value is equal to the total area of the load disturbance response. Shinskey [Shinskey, 1996, p 17] points out that the IAE is a valuable performance measure since it can be related to the economic cost of adding either too much or too little of an expensive ingredient. Other criteria for control performance such as the integrated square error, ISE, and the integrated time-weighted absolute error, ITAE, have also been suggested by e.g. [McMillan, 1983; Marlin, 1995].

According to e.g. [Åström and Hägglund, 2005, pp 112–116] and [Zhou and Doyle, 1998, pp 142–143] robustness can be captured by the sensitivity function

$$S(s) = \frac{1}{1 + P(s)C(s)F(s)}, \tag{7}$$

and the complementary sensitivity function

$$T(s) = \frac{P(s)C(s)F(s)}{1 + P(s)C(s)F(s)}. \tag{8}$$

We choose to set robustness constraints on the maximum values of these functions

$$|S(i\omega)| \leq M_s, \quad |T(i\omega)| \leq M_t, \quad \forall \omega \in \mathbb{R}^+, \tag{9}$$

where ω is the frequency in [rad/s] and $\mathbb{R}^+$ denotes the set of all non-negative real numbers. As shown in [Åström and Hägglund, 2005, pp 116–117] this corresponds to the open-loop Nyquist curve of $P(i\omega)C(i\omega)F(i\omega)$ not entering the M_s- or M_t-circles shown in Fig. 2. The circles expand with decreasing values of M_s and M_t, resulting in greater closed-loop robustness. M_s- and M_t-values ranging between 1.2 and 2.0 give reasonable robustness and correspond to gain margins between 6 and 2, as well as phase margins between 49° and 29°.

Large control signal activity generated by measurement noise could cause undesirable actuator wear and tear. The transfer function from measurement noise to control action is

$$S_c(s) = \frac{C(s)F(s)}{1 + P(s)C(s)F(s)}, \tag{10}$$

and its impact depends on many factors, with the controller parameters and low-pass filter being particularly important. In this article we have chosen to constrain the H_2-norm of $S_c(s)$

$$\|S_c(s)\|_2 \leq \kappa_u, \tag{11}$$

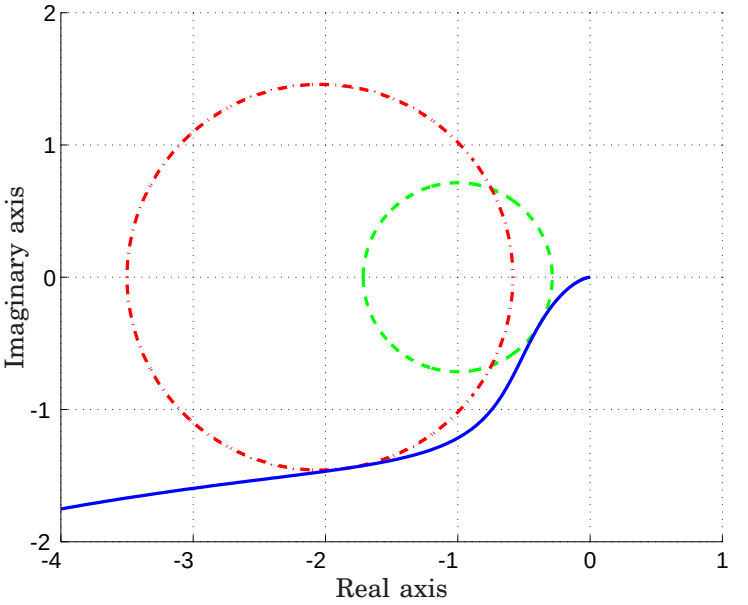

Figure 2. The robustness constraints in (9) are fulfilled if the open-loop Nyquist curve (solid) does not enter the M_s-circle (dashed) or the M_t-circle (dash-dotted).

to limit closed-loop noise sensitivity with κ_u as the upper limit of control signal variation. $\|S_c(s)\|_2$ can be directly related to the control signal magnitude via the control signal standard deviation due to measurement noise, σ_u, which should be easy for practitioners to relate to. To determine σ_u we must also know the characteristics of the measurement noise, for example its power spectral density. For the purpose of analysis we will, therefore, assume continuous-time white Gaussian measurement noise with unit spectral density. $\|S_c(s)\|_2$ is derived using the integral formula

$$\|S_c(s)\|_2 = \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} |S_c(i\omega)|^2 d\omega}. \tag{12}$$

For real processes, where the noise characteristics is typically different, it makes more sense to use σ_u as the measure of noise sensitivity. However, one of the biggest advantages with the proposed PID design procedure, is that there is no need to derive σ_u explicitly for real processes. See Section 5 for more information on this.

Taking all three criteria of performance, robustness and noise sensitivity into consideration, we can now formulate a constrained optimization problem

$$\begin{aligned} \underset{K,T_i,T_d,T_f\in\mathbb{R}^+}{\text{minimize}} \quad & \int_0^\infty |e(t)|dt, \\ \text{subject to} \quad & |S(i\omega)| \leq M_s, \\ & |T(i\omega)| \leq M_t, \quad \forall\omega \in \mathbb{R}^+, \\ & \|S_c(s)\|_2 \leq \kappa_u, \end{aligned} \tag{13}$$

which can be used to design any of the controllers $C(s)$ given in Eqs. (1–3) as well as the filters in Eqs. (4–5). However, the non-convexity of this optimization problem makes it difficult to solve directly.

2.4 A software tool for robust PID design

A MATLAB®-based software tool for robust PID design was presented in [Garpinger and Hägglund, 2008]. This tool solves a modified version of Eq. (13) without the noise sensitivity constraint

$$\begin{aligned} \underset{K,T_i,T_d\in\mathbb{R}^+}{\text{minimize}} \quad & \int_0^\infty |e(t)|dt, \\ \text{subject to} \quad & |S(i\omega)| \leq M_s, \\ & |T(i\omega)| \leq M_t, \quad \forall\omega \in \mathbb{R}^+, \\ & |S(i\omega^s)| = M_s, \quad \text{and/or} \\ & |T(i\omega^t)| = M_t, \end{aligned} \tag{14}$$

where ω^s are frequencies for which the open-loop frequency response is tangent to the M_s-circle and vice versa for ω^t and the M_t-circle. Either ω^s or ω^t could be an empty set, but not simultaneously. As a direct result of these additional equality constraints at least one of the inequality constraints will be active, thus forcing the Nyquist curve to touch either one or both robustness circles. This makes the optimization problem easier to solve than the problem with only inequality constraints. The drawback is that there could be a controller with greater robustness that yields better performance. Several studies by [Garpinger, 2009; Garpinger et al., 2014; Grimholt and Skogestad, 2012] do, however, indicate that the solution to the optimization problem (14) is, in most cases, optimal also without equality constraints just as long as the robustness is reasonable with M_s and M_t roughly below 1.8. The low-pass filter time constant, T_f, is fixed in this optimization problem such that we design controllers for the process and filter combination $P(s)F(s)$.

We have chosen to call this PID controller tuning method Software-based optimal robust design (SWORD). The software tool needs a stable linear process model as well as specified values of M_s, M_t and T_f to work. The optimization problem is solved using a simplex-based algorithm called the Nelder-Mead method, see e.g. [Walters et al., 1991; Lagarias et al., 1998]. The simplex propagates through the space of (T_i,T_d) for PID control and (T_i) for PI control. The proportional gain K is chosen such that each open-loop Nyquist curve fulfills either or both equality constraints. The IAE-value is derived through Simulink® simulations for every controller that the algorithm propagates through. More detailed descriptions of the program are given in [Garpinger, 2009; Garpinger and Hägglund, 2008]. The software tool is freely available at:

`http://www.control.lth.se/project/PID`

2.5 A process batch for controller evaluation

A batch of 134 stable processes with monotonous step responses, representative for the process industry, was presented in [Hägglund and Åström, 2004]. These processes will be utilized here to show which types of processes have the greatest benefit of the derivative part of the PID controller. The nine different process types from the batch can be found in Fig. 3, while the parameter values used are the same as in [Hägglund and Åström, 2004]. Each process has been assigned a symbol in the figure that will later be used in plots.

Notice that the process gains will affect the value of $\|S_c(s)\|_2$. According to [Shinskey, 1996, p 99] many processes have static gains in the range $1-10$. It is reasonable to believe, however, that integrating processes (like P_6) could have much smaller integral gains e.g. for large tanks. In this article we have chosen to set both static and integral process gains to 1 for the batch. Like the characteristics and intensity of the noise, we believe that these gains vary more or less randomly from process to process. In our controller comparisons in Section 6, we will make conclusions with regards to this assumption. However, processes with e.g. high noise intensity, a sensitive actuator or small process gain, will always be more sensitive to noise throughput.

The processes in Fig. 3 will be approximated with first-order time-delayed (FOTD) models

$$P_m(s) = \frac{K_p}{sT+1}e^{-sL}, \tag{15}$$

in order classify them using the normalized time delay,

$$\tau = \frac{L}{L+T}, \tag{16}$$

$$P_1 : \frac{e^{-s}}{1+sT} \ (\circ) \qquad P_2 : \frac{e^{-s}}{(1+sT)^2} \ (+)$$

$$P_3 : \frac{1}{(s+1)(1+sT)^2} \ (\times) \qquad P_4 : \frac{1}{(s+1)^n} \ (\square)$$

$$P_5 : \prod_{i=0}^{3} \frac{1}{(1+\alpha^i s)} \ (\Diamond), \qquad P_6 : \frac{e^{-sL_1}}{s(1+sT_1)} \ (*)$$

$$P_7 : \frac{e^{-sL_1}}{(1+sT)(1+sT_1)} \ (\triangle), \qquad P_8 : \frac{1-\alpha s}{(s+1)^3} \ (\star)$$

$$P_9 : \frac{1}{(s+1)((sT)^2+1.4sT+1)} \ (\ast)$$

Figure 3. Test batch of processes representative for the process industry. The parameter values used are the same as in [Hägglund and Åström, 2004]. Notice that each type of process has been assigned a symbol in parenthesis.

which ranges from 0 to 1. We define a process as lag-dominated if $0 \leq \tau \leq 0.2$, balanced if $0.2 < \tau < 0.7$, and delay-dominated if $0.7 \leq \tau \leq 1.0$. The FOTD model parameters have been derived through step response tests using the 63%-rule, described in e.g. [Garpinger and Hägglund, 2014]. In this article we will mainly run SWORD with the actual process models, not the FOTD models. The principal reason for doing so is because we want to analyze the control at its maximum potential, using other models would lead to comparisons that has as much to do with the modeling method as it has to do with controller design. In Section 5, however, we will briefly comment on why the step response models should not be used together with SWORD. For a more thorough discussion on modeling for optimal PID control, see [Garpinger and Hägglund, 2014].

We will use three processes from the batch to further analyze the trade-offs between performance, robustness and noise sensitivity. The first process

$$P_I(s) = \frac{1}{(s+1)(0.1s+1)(0.01s+1)(0.001s+1)}, \tag{17}$$

is lag-dominated ($\tau = 0.067$), the second process

$$P_{II}(s) = \frac{1}{(s+1)^4}, \tag{18}$$

is balanced ($\tau = 0.33$), and the third process

$$P_{III}(s) = \frac{e^{-s}}{(0.05s+1)^2}, \tag{19}$$

is delay-dominated ($\tau = 0.93$).

3. Performance and noise sensitivity trade-off

In Section 5 we suggest a PID design method where robustness is set to a fixed value $M = M_s = M_t$, with M_s equal to M_t for simplicity. For now, we will assume $M = 1.4$, see Section 4 for a discussion on how the robustness level affects PID controller design. The filter time constant, T_f, is varied to create a set of controllers given by the SWORD method. Figure 4 shows how the IAE-value depends on $\|S_c(s)\|_2$ for an example set of PID controllers on the process $P_{II}(s)$, (18). As pointed out in [Garpinger, 2009], there is little to gain in terms of performance for small values on T_f, but it costs a lot in terms of noise sensitivity. For low values on $\|S_c(s)\|_2$ the opposite holds, where decreasing noise sensitivity quickly deteriorates performance. A good trade-off between performance and noise sensitivity for the curve in Fig. 4 would be to choose the controller that gives $\|S_c(s)\|_2 \approx 10$. A different approach is to pick the controller that results in a specified maximum allowed noise sensitivity, κ_u. Figure 5 shows how performance (top plot) and noise sensitivity (bottom plot) depend on T_f. The filter has a major impact on the PID controller and the closed-loop performance for filter time constants roughly between $T_f = 0.1 - 10$. In discrete-time control, sampling will also act as a filter which will limit the best possible performance and the maximum noise sensitivity, see e.g. [Garpinger, 2009].

The transfer functions of the controller and filter combinations will be C_I, $C_{PI}F_{PI}$ and $C_{PID}F_{PID}$. As the filter time constant T_f increases, the PID controller will gradually become similar to a PI controller, while the PI controller becomes similar to an I controller. The PI and PID forms are fundamentally different since the PI controller will not use a second-order filter and because it has no D-part. However, in order to simplify our analysis we will say that the combination $C_{PID}F_{PID}$ becomes a PI controller when its proportional, derivative and filter parts

$$K \cdot (1 + sT_d) \cdot \frac{1}{(sT_f)^2/2 + sT_f + 1}, \tag{20}$$

have no phase greater than 0°, i.e. if

$$\arctan(\omega T_d) - \arctan\left(\frac{\omega T_f}{1 - 0.5\omega^2 T_f^2}\right) \leq 0, \forall \omega \in \mathbb{R}^+. \tag{21}$$

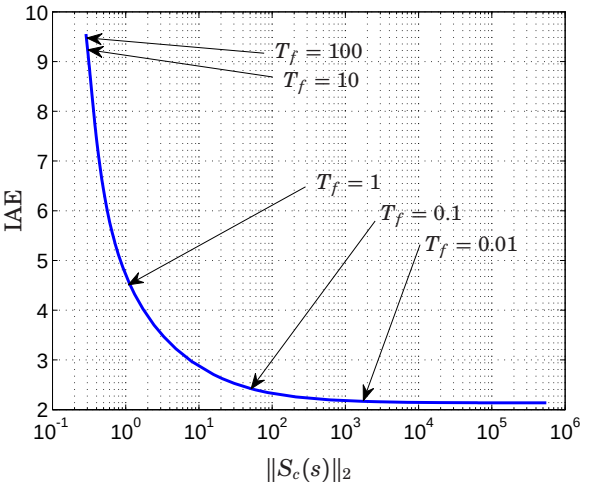

Figure 4. Varying the filter time constant T_f in SWORD, for a given robustness $M = M_s = M_t$, results in a set of controllers, in this case PID. For the balanced process $P_{II}(s)$ and $M = 1.4$ both performance and noise sensitivity will decrease along the curve as the filter time constant T_f increases. Values of T_f are indicated in the plot.

The phase will always be zero for $\omega = 0$. The arctangent function is monotonically increasing and the order of the filter is higher than that of a PD controller. Thus, (21) is only true if the phase has no more zero crossings, i.e. if

$$T_d = \frac{T_f}{1 - 0.5\omega^2 T_f^2}, \tag{22}$$

has no solution. Solving (22) with respect to ω gives

$$\omega = \frac{\sqrt{2}}{T_f}\sqrt{1 - \frac{T_f}{T_d}}, \tag{23}$$

which only has a solution $\omega > 0$ if $T_d > T_f$. Therefore, our PID controllers will be called PI controllers if $T_f \geq T_d$. Notice that $T_d \neq 0$ when this happens. As T_f increases, the PI and PID forms in (2) and (3) will eventually converge to the same I controller (1) when the zeros of the controllers cancel the poles of the filters. This effect can be seen in Fig. 5 for the PID case, where the performance and noise sensitivity curves are almost flat for $T_f > 10$. Notice that the filter time constant for the PI controller will not be the same as for the PID controller at this point.

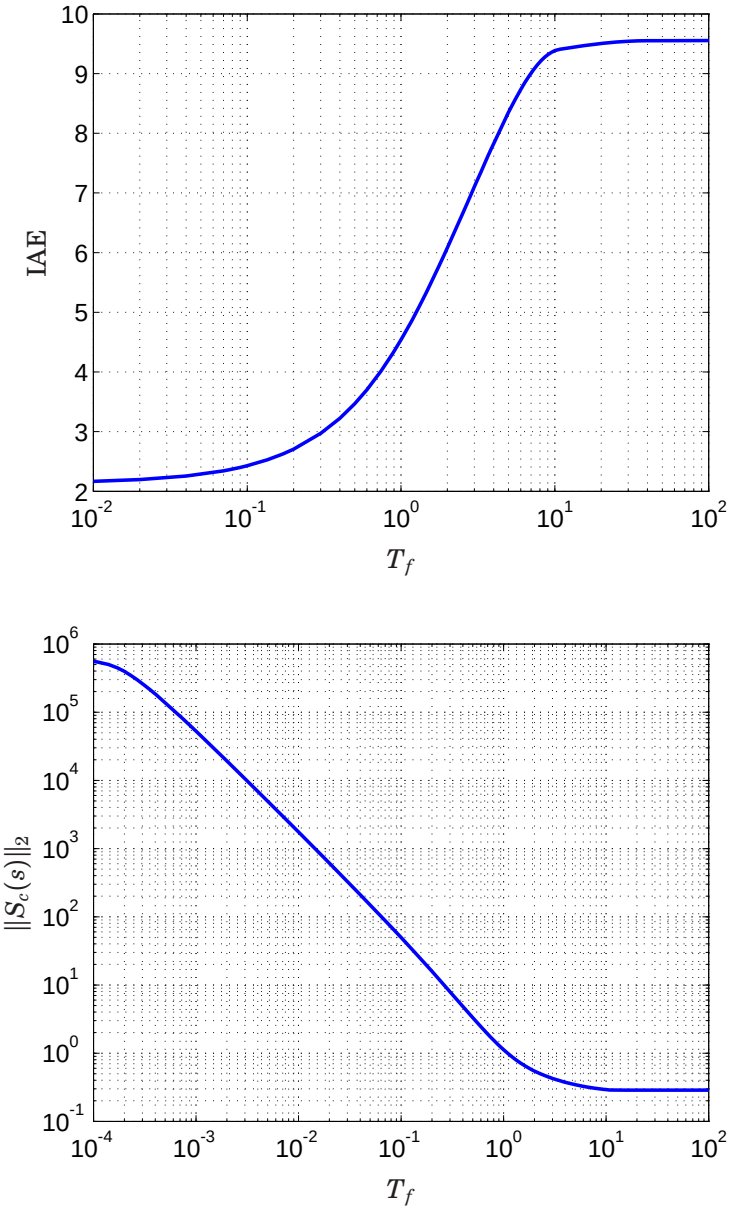

Figure 5. The relation of the filter time constant T_f with performance IAE (top) and noise sensitivity $\|S_c(s)\|_2$ (bottom) for the balanced process $P_{II}(s)$ with $M = 1.4$ when SWORD is used to derive PID controllers.

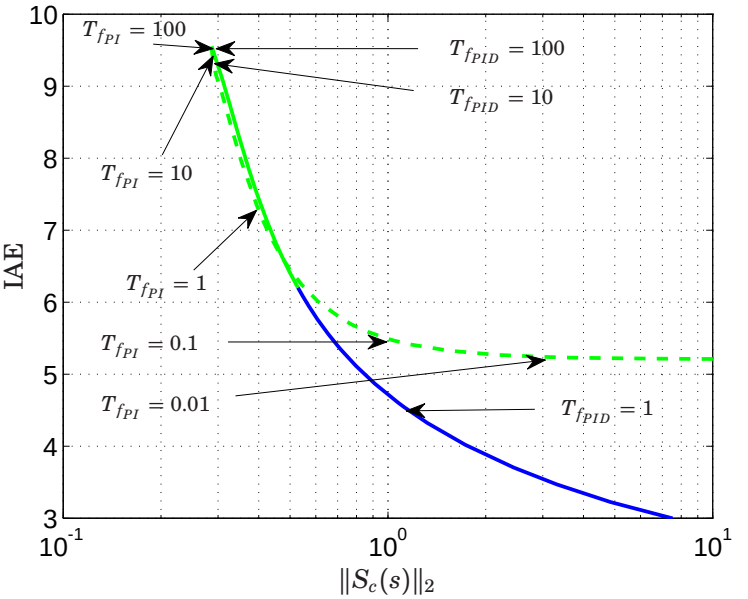

Figure 6. Performance and noise sensitivity trade-offs for $P_{II}(s)$ with $M = 1.4$, comparing the sets of PI (dashed) and PID controllers (solid) derived with the SWORD method. The green part of the PID curve shows when they are defined as PI controllers, i.e. when $T_f \geq T_d$. PI (left) and PID (right) values of T_f are indicated in the plot.

In [Garpinger, 2009], optimal Youla parametrized controllers of high order were derived for almost the same optimization problem as (13). It was shown that the optimal controller type depends on the maximum allowed noise sensitivity. Higher-order controllers are preferable for higher noise sensitivity, while PID, PI and even I controllers become optimal as the maximum noise sensitivity limit is gradually decreased. It was also shown in the same thesis that for a certain value on $\|S_c(s)\|_2$, the PI controllers from SWORD become better than the PID controllers. Figure 6 shows this effect, which stems from the different controller forms. The set of PID controllers (3) are shown as the solid line in the figure, while the set of PI controllers (2) are given by the dashed line. The PID controllers become PI controllers ($T_f \geq T_d$) almost at the same point as the PI form becomes better than the PID form, around $\|S_c(s)\|_2 = 0.48$. As the two controller sets approaches $\|S_c(s)\|_2 = 0.29$, they both converge to the same I controller. If the noise sensitivity is still higher than specified at this point, we suggest that the user switch to an I controller and de-

crease the integral gain k_i until $\|S_c(s)\|_2$ is small enough. We will explain why in Section 4. The relative performance gain from using a PID controller depends on the maximum allowed noise sensitivity. If we can allow $\|S_c(s)\|_2 = 10$, the IAE-value for PI control is up to 80% higher than for PID control, while for $\|S_c(s)\|_2 = 1$ the IAE value is 16% higher. These trade-off curves will help us determine which processes have the biggest benefits from the D-part. They will also show us when to switch from PID to PI, and finally from PI to I control.

4. Controller optimality and robustness level selection

The SWORD method is used to design PI and PID controllers with different values on T_f. This is done by solving the optimization problem (14) with the given controller forms (2) and (3). But, how close to optimal will the set of controllers be with respect to the full optimization problem (13) and how will different choices of the robustness level M affect the design? In Section 2.4 we have already reasoned why controllers that give active robustness constraints in most cases also give the same solution as optimization without equality constraints. However, we still need to examine how close the set of controllers with different T_f and fixed robustness $M = M_s = M_t$ are to the optimal of (13) for different κ_u. The lag-dominated, balanced and delay-dominated processes $P_I(s) - P_{III}(s)$ (17–19) will be used to represent the batch of processes in Fig. 3, which is motivated in [Garpinger et al., 2014]. Varying T_f and M we can find all controllers that could possibly solve (13). To be sure we have found the optimal controller, however, there are two conditions that needs to be fulfilled. First of all, the performance and noise sensitivity trade-off curves should be monotonically decreasing with $\|S_c(s)\|_2$, i.e. as T_f decreases. Otherwise, we could miss controllers with both better performance and lower noise sensitivity unless we examine a set of controllers large enough to find these local minima. Secondly, we should beware that the trade-off curves do not intersect each other for different values on M, or else there are controllers with higher robustness that give better performance for the same noise sensitivity. To investigate whether or not we can expect any of these two cases when using SWORD, we have chosen to plot the trade-off curves for $P_I(s) - P_{III}(s)$ with the different robustness constraints $M = 1.2, 1.25, 1.3, 1.35, 1.4, 1.45, 1.5, 1.55, 1.6$.

We will define our controllers as PID controllers if they have the form (3) and $T_f < T_d$. PID controllers with $T_f \geq T_d$ and PI controllers (2) will be defined as PI controllers until they converge to the same I controller for high enough values on T_f. Thereafter, all controllers will be I controllers with different integral gains k_i.

The top plot of Fig. 7 shows the nine, tightly packed, trade-off curves for the lag-dominated process $P_I(s)$. Blue dash-dotted curves indicate PID controllers, green solid lines PI controllers, and red dashed curve I controllers. Notice that the IAE-value generally decreases as M increases. A closer look at the trade-off curves shows that they are all monotonically decreasing. Several of the curves with the highest M-values do, however, intersect each other when the PI controllers are close to become I controllers. The largest relative difference in performance due to these intersections is found between the curves for $M = 1.4$ and 1.6. The latter has an IAE-value 20% higher than the former at the $\|S_c(s)\|_2$-value where it becomes an I controller. Even so, we think it is fair to say that almost all controllers derived in the nine sets are optimal or at least near optimal for the full optimization problem (13).

The trade-off curves for the balanced process $P_{II}(s)$ are shown in the middle plot of Fig. 7. Notice that the span of PI controllers is narrower than for the lag-dominated process in the top plot of Fig. 7. This indicates that we gain more from using the D-part on this balanced process. Similar to the lag-dominated process, all curves are monotonically decreasing with $\|S_c(s)\|_2$. Some curves still intersect each other, but they are both fewer and give less relative increase in IAE compared to $P_I(s)$.

The delay-dominated process $P_{III}(s)$, has PID trade-off curves that are not monotonically decreasing. While this can be handled with our suggested method by collecting a large enough set of controllers, we have chosen to only consider the trade-off curves given by PI and I controllers. Several studies have reported that derivative action has practically no benefit for delay-dominated processes, see e.g. [Hägglund and Åström, 2004], and we will add further evidence to this standpoint in Section 6. The nine trade-off curves for $P_{III}(s)$ are presented in the bottom plot of Fig. 7. Notice that the highest obtainable noise sensitivity is naturally lower than for PID controllers. All nine trade-off curves are monotonically decreasing and the relative IAE loss due to intersecting curves is lower than for both $P_I(s)$ and $P_{II}(s)$.

Given the three examples, we now conclude that our suboptimal routine for the optimization problem (13) gives either optimal or close to optimal control for these three processes representing lag-dominated, balanced and delay-dominated processes. At least for the case with white Gaussian noise. Figure 7 also reveals that trade-off curves for higher robustness are generally above those for lower robustness. This means that M should always be set as high as possible to optimize performance, given that the user can still feel comfortable with the choice. Notice that the $\|S_c(s)\|_2$-value for which the PID to PI controller transition occurs decreases as the robustness increases. This has to do with the PID controllers having lower gains for higher robustness.

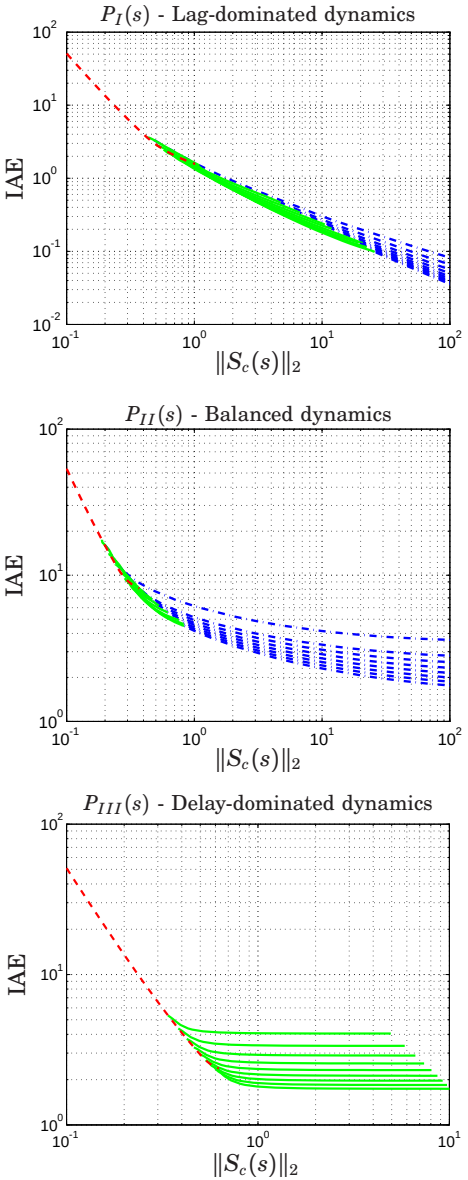

Figure 7. Performance and noise sensitivity trade-off curves for the processes $P_I(s)$ (top), $P_{II}(s)$ (middle) and $P_{III}(s)$ (bottom) with $M = 1.2, 1.25, 1.3, ..., 1.6$. Blue dash-dotted curves indicate PID controllers, green solid lines show PI controllers and red dashed curves I controllers. Notice that the trade-off curves with higher relative IAE-values have lower relative values on M.

5. PID design procedure

In this section we will introduce our proposed software-based PID design method for real processes. Given the information in Sections 3 and 4, we suggest the following procedure:

1. Determine a process model and specify a maximum allowed robustness measure $M = M_s = M_t$. For example $M = 1.4$.
2. Find an initial controller along the trade-off curve and employ it on the process.
3. Derive a set of controllers along the trade-off curve by adjusting T_f for PI and PID control and k_i for I control.
4. Use e.g. a control knob for online selection of controllers along the set until finding the best controller that keeps the high-frequency control signal variations below a maximum allowed level.

The same set of controllers can be used to detune the controller if it is deemed too aggressive. Here we suggest use of the SWORD method to find the set of controllers, but it should be possible to use roughly the same steps with other methods, e.g. the one presented in [Hast et al., 2013]. Steps 1–3 should be easy enough to automate, but the fourth step is preferably carried out manually. The initial controller could be determined in several ways. If the process is either lag-dominated or delay-dominated one option is to start with the I controller that gives the chosen robustness M. If the process is balanced, one can instead find the best PI controller along the trade-off curve, i.e. at the transition to PID controllers. The set of controllers could initially contain somewhere between 10 to 20 different controllers, with the possibility to expand it if desired. It can be expected to take somewhere between 2–10 minutes to find the entire set of controllers with SWORD on a computer with reasonable performance. If the set of controllers is dense enough, parameters could also be interpolated in order to derive a continuous set of parameters. The final controller can be chosen either by visual feedback of the control signal level or by finding a controller for which the losses in performance and noise sensitivity are balanced. There should also be some information to the user on how much of the nominal performance is lost when selecting different controllers from the set.

We will illustrate the design procedure with an example. Assume that we want to find a controller for the balanced process $P_{II}(s)$ that gives good closed-loop performance with the constraints that $M = 1.4$ and the control signal amplitude should be roughly below 5. Using SWORD on $P_{II}(s)$ we derive an initial PID controller with $T_f = 0.5$, called PID_1.

Table 1. A set of PID, PI and I controller parameters derived for the process $P_{II}(s)$ using the constraint $M = 1.4$ in SWORD.

Controller	K	T_i	T_d	k_i	T_f	IAE	$\|S_c\|_2$
PID_1	0.84	2.32	1.36	-	0.50	3.47	3.31
PID_2	0.72	2.39	1.43	-	0.78	4.10	1.61
PID_3	0.59	2.59	1.75	-	1.50	5.40	0.70
PI_1	0.39	2.57	-	-	1.00	7.34	0.40
PI_2	0.47	4.29	-	-	3.70	9.10	0.31
I_1	-	-	-	0.05	-	20.00	0.17

The reason for choosing an initial PID is just to show monotonous decreases in both control signal activity and control performance. The top plot of Fig. 8 shows the performance and noise sensitivity trade-off plot for $P_{II}(s)$. PID_1 is marked with the blue triangle furthest to the right in the trade-off curve. This controller results in the solid blue load disturbance response in the middle plot and the control signal noise response furthest to the left (0–200 seconds) in the bottom plot. The measurement noise response was simulated in MATLAB® Simulink® using band-limited white measurement noise with unit spectral density and a sampling period of $h = 0.01$ seconds. From this response, we see that the control signal amplitude is greater than our specification allows. Next we derive five more controllers, two PID controllers with $T_f = 0.78$ (PID_2) and $T_f = 1.50$ (PID_3), two PI controllers with $T_f = 1.00$ (PI_1) and $T_f = 3.7$ (PI_2), as well as an I controller (I_1) with $k_i = 0.05$. This gives us a total set of six controllers with the parameters, IAE and $\|S_c(s)\|_2$-values given in Table 1. In Fig. 8 (top plot) the PID controllers are marked with blue triangles, PI controllers with green squares and the I controller with a red circle. The load disturbance responses are collected in the middle plot with the same color scheme as the top plot. I_1, PI_1 and PID_1 are marked with solid lines, PI_2 and PID_2 with dash-dotted lines, while PID_3 is marked with a dashed line. The measurement noise responses are shown within the time intervals: $0-200$ (PID_1), $200-400$ (PID_2), $400-600$ (PID_3), $600-800$ (PI_1), $800-1000$ (PI_2), and $1000-1200$ seconds (I_1). Since PID_2 satisfies our noise sensitivity specification of a control signal amplitude roughly below 5, we finally select this controller for our process.

Notice that we have used a perfect model of $P_{II}(s)$ in the example. In [Garpinger and Hägglund, 2014], it was pointed out that SWORD should not be combined with FOTD models derived from step responses, as described in the same paper. For the process batch in Fig. 3, such models result in robustness values on M_s and M_t between $1.4-2.3$ for the choice of $M = 1.4$. For the previous example, the resulting M_s-values would be

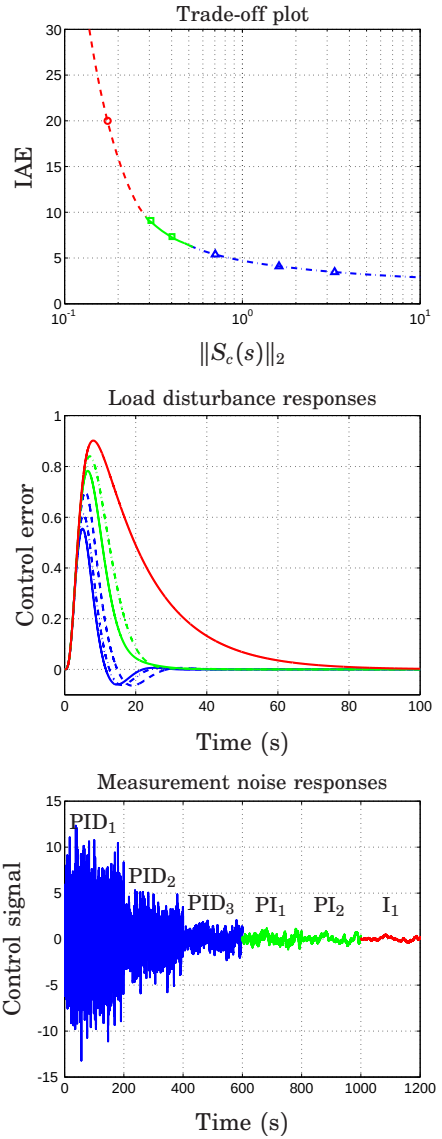

Figure 8. Given the process $P_{II}(s)$ and the constraint $M = 1.4$, the performance and noise sensitivity trade-off curve is shown in the top plot where the six different controllers in Table 1 are marked with triangles (PID), squares (PI) and a circle (I). The load disturbance and control signal noise responses are shown in the middle and bottom plots respectively. PID controllers are shown in blue, PI controllers in green and the I controller in red. From this information, the user can select the best performing controller that gives the maximum acceptable noise sensitivity.

1.70 (PID_1), 1.65 (PID_2), 1.58 (PID_3), 1.55 (PI_1), 1.47 (PI_2), 1.17 (I_1). In [Garpinger and Hägglund, 2014], it is also shown that the same step response models work poorly together with PID tuning rules like Lambda tuning [Dahlin, 1968], AMIGO [Hägglund and Åström, 2004] and SIMC [Skogestad, 2003]. It is thus highly desirable to find modeling methods that work well together with PID design methods.

6. Benefits of the derivative part

As we could see in Fig. 7, the $\|S_c(s)\|_2$-values for which the transitions between PID, PI and I controllers occur depends on the process and the robustness constraints. To investigate which processes have the most to gain from using the D-part of the PID controller, we will assume a robustness of $M = 1.4$ and examine where the controller transition from PI to PID occurs for the processes in Fig. 3. We will also compare the performance of the PI and PID controllers for $\|S_c(s)\|_2 = 0.7, 2, 10$, which will be referred to as low, medium and high noise sensitivity respectively.

Hägglund and Åström [Hägglund and Åström, 2004] used the integrated errors of PI and PID controllers derived with the method in [Panagopoulos et al., 2002] to show which processes in the test batch have the most to gain from using the derivative part. This study showed that lag-dominated processes have the most to gain from using the D-part although the variation in relative performance increase is large. These controllers were, however, designed without any filter or noise sensitivity constraint. A PID/PI performance plot, similar to the one in [Hägglund and Åström, 2004], could be made with SWORD if it is run with low values on the filter time constant T_f, but we believe that the addition of a noise sensitivity constraint is the key to reveal when the control really benefits from the D-part. Figure 9 shows how the value of $\|S_c(s)\|_2$ for which the PI controller is as good as the PID controller, $\|S_c(s)\|_2 = \kappa_{eq}$, varies with normalized time delay τ over the process batch. Most processes seem to follow the same general pattern that κ_{eq} increases as τ decreases. This shows that many lag-dominated processes should be controlled with PI controllers even if we can allow relatively high values on the noise sensitivity. There are, however, some processes that deviate from this pattern. The integrating processes in P_6 with $\tau = 0$ have low values on κ_{eq}, suggesting that they have a lot to gain from the D-part. The second-order time-delayed processes with identical poles, P_2 in the same figure, obtain the smallest values on κ_{eq} as τ decreases. This indicates that they too will benefit a lot from the D-part. The processes in P_3 that resembles second-order processes, P_4, and P_8 with larger values on α also have slightly lower κ_{eq} than most other processes.

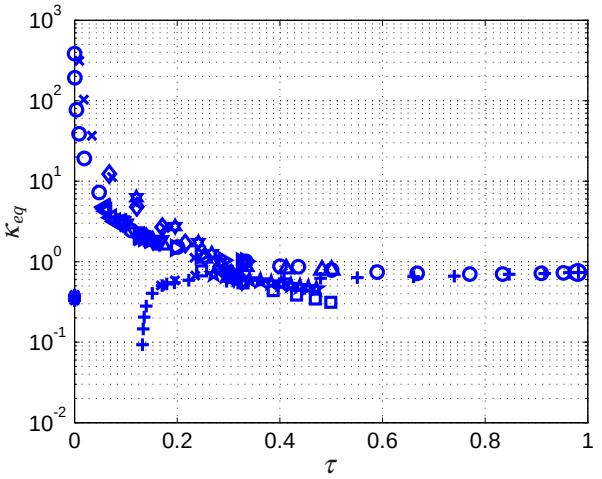

Figure 9. Variation of the noise sensitivity for which PI controllers perform equally well to PID controllers, $\|S_c(s)\|_2 = \kappa_{eq}$, with normalized time delay τ. The different symbols are associated with the processes of the batch as shown in Fig. 3.

The SWORD method gives either a PI or a PID controller and it is useful to know when to switch from PID to PI form. Figure 10 shows how T_f/T_d, for which the PI form in (2) performs equally well to the PID form (3), varies with normalized time delay τ. The function $f(\tau) = 0.43\tau^2 + 0.45\tau + 0.78$ is marked as a dashed green line in the figure and provides a lower limit for the test batch. If the ratio between T_f and T_d surpasses this function, one should start comparing PID controllers with PI controllers or perhaps even switch over to PI control altogether.

While Fig. 9 indicates which processes have the best potential to benefit from the D-part, it says nothing about the relative performance gain one can expect when going from PI to PID. To give a measure of this, we have calculated the IAE-values of the PI and PID controllers for three specific noise sensitivities, $\|S_c(s)\|_2 = 0.7, 2$ and 10. These will correspond to low, medium and high noise sensitivity. This study covers the whole test batch, but we will start by looking at the process types P_1 and P_2. Figure 11 shows the relative performance loss in using PI control compared to PID control for the first-order time-delayed processes P_1. The circles along the red line show the ratios for $\|S_c(s)\|_2 = 0.7$, the green squares show the same for $\|S_c(s)\|_2 = 2$, and the blue triangles show the benefits for $\|S_c(s)\|_2 = 10$. For the low noise sensitivity, there is nothing to gain for any of the processes in P_1. With medium noise sensitivity, processes with

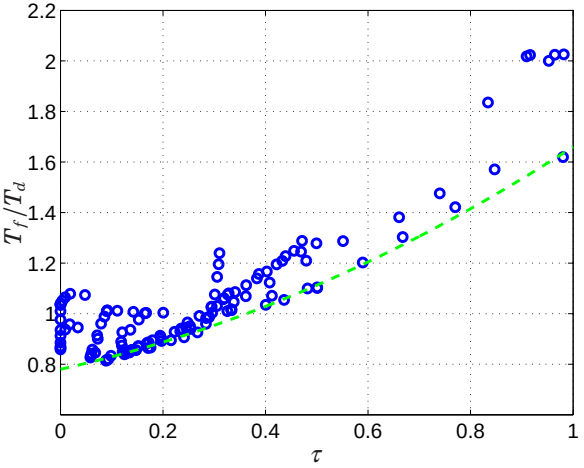

Figure 10. Values on T_f/T_d for which the PI form becomes better than the PID form and how they vary with τ. The dashed green line shows the function $f(\tau) = 0.43\tau^2 + 0.45\tau + 0.78$. If the ratio surpasses this function, one should start to compare PID controllers with PI controllers.

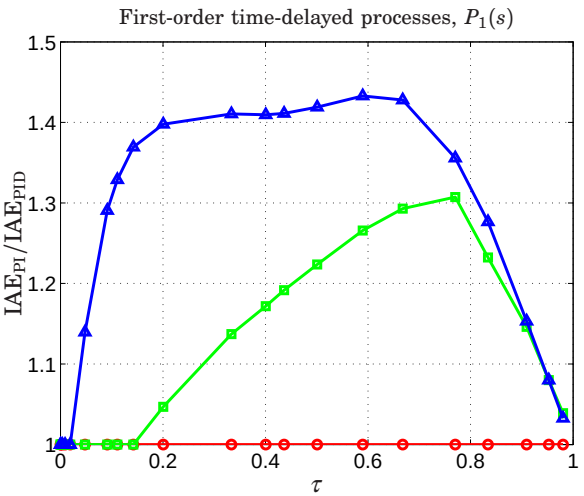

Figure 11. The relative performance loss from using PI control instead of PID control for the first-order time-delayed processes in P_1 when the noise sensitivity is $\|S_c(s)\|_2 = 0.7$ (red circles), $\|S_c(s)\|_2 = 2$ (green squares) or $\|S_c(s)\|_2 = 10$ (blue triangles).

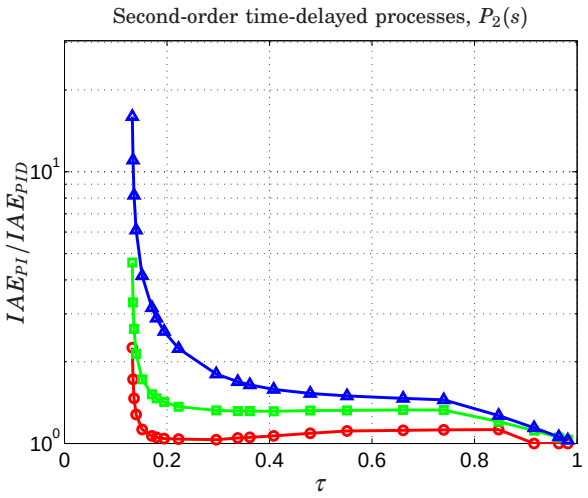

Figure 12. The relative performance loss from using PI control instead of PID control for the second-order time-delayed processes in P_2 when the noise sensitivity is $\|S_c(s)\|_2 = 0.7$ (red circles), $\|S_c(s)\|_2 = 2$ (green squares) or $\|S_c(s)\|_2 = 10$ (blue triangles). Notice the logarithmic scale for the performance ratio.

$\tau = 0.3 - 0.9$ have the most to gain from using the D-part. The relative gains are, however, rather low for all processes. The process with $\tau = 0.77$ has the biggest performance loss ratio, with its PI controller having 30% worse IAE than the PID controller. For the high noise sensitivity, processes with $\tau = 0.1 - 0.8$ have the most use of the D-part. This covers the whole field of balanced processes. The highest relative losses in IAE from PID to PI are, however, still quite modest. It is obvious from looking at Fig. 11, that the most lag-dominated and delay-dominated processes have very little to gain from using the D-part. These lag-dominant processes could, however, benefit from the D-part if the noise sensitivity is allowed to be very high. Given that PI control is not more than 43% worse than PID control for any of the processes with any of the three noise sensitivities, one could argue that PI control could be used for all processes with essentially first-order time-delayed dynamics unless noise is not an issue.

Figure 12 shows the relative performance losses for second-order processes with identical poles, P_2. The processes and relative noise sensitivities are marked with the same signs and color schemes as in Fig. 11. Notice the logarithmic scale for the performance loss ratio. For $\tau = 0.5 - 1$, the three curves are similar to those for P_1, although there is now also a slight benefit in going from PID to PI for the low noise sensitivity. The

main difference between process type P_1 and P_2 is, however, what happens for the most lag-dominated second-order processes. For $\|S_c(s)\|_2 = 0.7$, there is as much as a 225% performance loss in using PI controllers instead of PID. The same numbers for $\|S_c(s)\|_2 = 2$ and 10 are 464% and 1590% respectively. This shows that process dynamics with two identical poles and small relative time delay have a lot to gain from using the D-part.

Figure 13 displays how $\text{IAE}_{\text{PI}}/\text{IAE}_{\text{PID}}$ varies over the whole test batch in Fig. 3, for $\|S_c(s)\|_2 = 0.7$ (top plot), $\|S_c(s)\|_2 = 2$ (middle plot) and $\|S_c(s)\|_2 = 10$ (bottom plot). Notice the different scales for the performance loss ratios. As expected, process types P_2 and P_6 stand out of all three plots as those where PID control is the most beneficial. We have already described the pattern for P_2 and for P_6 the processes with least time delay give the largest performance loss ratios. Notice, however, that for integrating processes with low values on the integral gain, this effect will be much smaller. These could be compared with the most lag-dominated processes in P_1 if these are approximated as integrating processes with $K_v = 1/T$. For low noise sensitivity, process types P_4 and P_8 also stand out as those where PID is preferred. For the rest of the process types there is more of a general pattern that differs over the three noise sensitivities. The conclusions are, however, pretty much the same as those for P_1 alone. Lag- and delay-dominated processes, besides P_2 and P_6, have little to gain from using PID control. Processes roughly in the interval $\tau = 0.1 - 0.8$ have somewhat more to gain, but for processes with requirements on low noise sensitivity there seems to be generally little to gain from using the D-part. Second-order processes with similar poles and little time delay as well as second-order integral processes are, however, worthwhile to design PID controllers for even in these cases.

7. Conclusions and discussion

In this article, we have stated a non-convex optimization problem (13) for design of optimal PID controllers with respect to robustness and noise sensitivity constraints. We have also shown how to derive optimal or near optimal controllers for this problem by repeated use of a MATLAB®-based software method called SWORD. Variation of the noise filter time constant, its order and the controller form for the least acceptable robustness, lead to a set of PID, PI and I controllers which the user can switch between until the control signal activity is acceptable. This way, the optimization can be handled without any noise modeling. The same set of controllers can also be used to detune the controller if it is deemed too aggressive for the actuator. Even though we have chosen to only use our

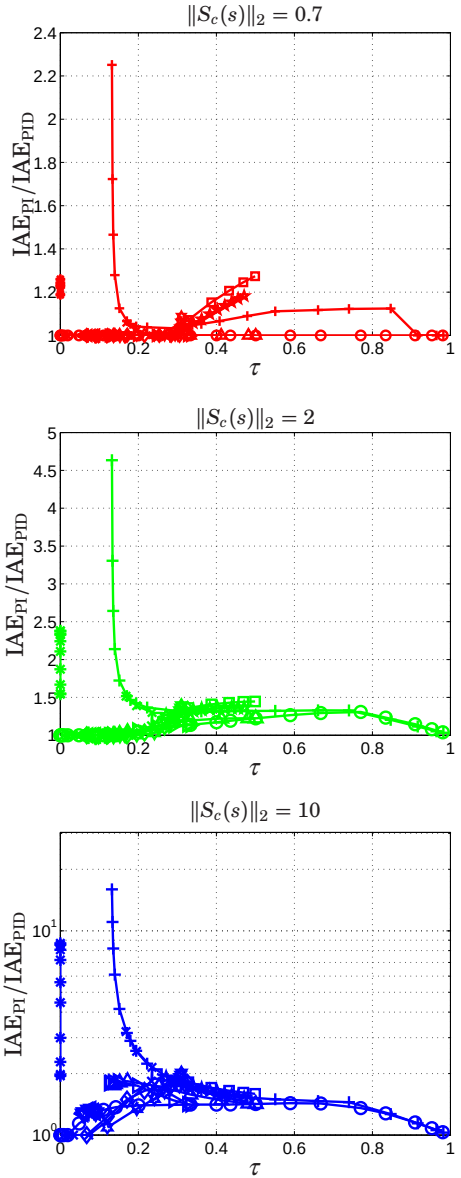

Figure 13. The relative performance loss from using PI control instead of PID control for the test batch in Fig. 3 when the noise sensitivity is $\|S_c(s)\|_2 = 0.7$ (top plot), $\|S_c(s)\|_2 = 2$ (middle plot) or $\|S_c(s)\|_2 = 10$ (bottom plot). Notice the different scales for the performance loss ratios, especially the logarithmic scale in the bottom plot. The different symbols are associated with the processes as shown in Fig. 3.

own PID design tool in this study, it seems reasonable to believe that the same method could also work with other toolboxes like those presented in e.g. [Oviedo et al., 2006; Harmse et al., 2009; Sadeghpour et al., 2012].

The H_2-norm of the sensitivity function between measurement noise and control signal is used as a measure of the closed-loop noise sensitivity. We show that by deriving the PI and PID controllers that give equivalent noise sensitivity, one can compare how useful the derivative part is for a certain process. A batch of processes representative for the process industry was used to highlight some types of processes that benefit more from the D-part as well as some processes that benefit less. The same method should also work to determine which processes gain the most from using even higher-order controllers if the PI controllers are exchanged for e.g. optimal Youla parametrized controllers derived in the same way as in [Garpinger, 2009]. From our study of PI and PID controllers, one can conclude that the optimal controller type depends on both the maximum allowed noise sensitivity of the closed-loop system and the process dynamics. For example, many lag-dominated processes benefit very little from the D-part unless the noise sensitivity is allowed to be very high. In accordance with other studies, delay-dominated processes are also shown to have little use of the D-part, even for high values on the noise sensitivity. Balanced processes have more to gain from the derivative action. However, of the processes examined the two process types that have shown to have the biggest benefit of the D-part are:

- Second-order processes with two identical poles and small relative time delay.
- Second-order integrating processes with high integral gain and little delay.

These two process types are both lag-dominated, which separate them from the rest of the lag-dominated processes that benefit less from the D-part. We believe that this has to do with the difference in relative order between the processes since the other lag-dominated processes are essentially first-order systems. For this reason there seems to be room for an even better classification of process dynamics than what can be done with the normalized time delay, τ. The fact that PID control is superior to PI for these two process types also makes it highly desirable to be able to distinguish them from other processes. There should be a lot to gain from using the D-part on these processes even if non-optimal control design is used.

Tuning rules, like Lambda and IMC tuning, are the most commonly used methods for systematic PID tuning in the process industry. In this article we have proposed a new software-based optimal design method for

Table 2. Pros and cons of software-based tuning and tuning rules

	Tuning Rules	Software Tuning
Pros	1. Fast	1. Optimality
	2. Intuitive	2. Three criteria
	3. Easy to use	3. Pedagogic tool
Cons	1. Less than three criteria	1. Need better models
	2. Not optimal	2. Need autotuning
		3. Poor availability

PID controllers and in Table 2 we have summarized some of the most important advantages (pros) and disadvantages (cons) of both tuning rules and software-based optimal design. Tuning rules have the advantages that they are fast, intuitive and simple to use, at least the ones mentioned. These properties make them popular in the process industry where general control knowledge often is poor and there is little time to spend on tuning each loop. It is, on the other hand, very difficult to capture all three criteria of performance, robustness and noise sensitivity in just a couple of simple formulas and they can thus be quite far from optimal, see e.g. [Garpinger et al., 2014]. As soon as the complexity and number of tuning rules needed to tune a process increase, people will stop using them. Tuning rules are thus not so suitable for design of controllers with derivative action and a low-pass filter. As we have shown in this article, software-based optimal design methods are able to provide optimal controllers with respect to performance, robustness and noise sensitivity. They are also pedagogic tools that can help us understand PID design better, for example by showing which processes benefit the most from the D-part. In [Garpinger and Hägglund, 2014] we also showed that optimal PID control design needs better models than what is typically given in process industry today. The level of control knowledge needed to understand and use software-based optimal design methods together with this demand on more accurate models make us believe that the methods should rather be part of a new generation of autotuners than stand-alone methods. The lack of such tools in industry rules out current use of software-based optimal design methods, at least on a broader scale. We do, however, believe that the development potential is much greater for software-based optimal design than for tuning rules and that the key to success in the field is to find fast modeling tools that provide just enough process knowledge to work with optimal control design, see [Garpinger and Hägglund, 2014]. For these reasons, we believe that easy to use software tools for optimal controller design will be the future for PID control rather than tuning rules.

8. Acknowledgements

This study was partly funded by the Swedish Foundation for Strategic Research through the PICLU center. The authors are members of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University.

References

Alcántara, S., R. Vilanova, and C. Pedret (2013). "PID control in terms of robustness/performance and servo/regulator trade-offs: A unifying approach to balanced autotuning". *Journal of Process Control* **23**:4, pp. 527–542.

Alfaro, V. M. and R. Vilanova (2012). "Conversion Formulae and Performance Capabilities of Two-Degree-of-Freedom PID Control Algorithms". In: *17th. IEEE Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE. Krakow, Poland.

Alfaro, V. M. and R. Vilanova (2013). "Performance and Robustness Considerations for Tuning of Proportional Integral/Proportional Integral Derivative Controllers with Two Input Filters". *Industrial & Engineering Chemistry Research* **52**:51, pp. 18287–18302.

Åström, K. J. and T. Hägglund (2005). *Advanced PID Control*. ISA – The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC.

Åström, K. J., H. Panagopoulos, and T. Hägglund (1998). "Design of PI Controllers based on Non-Convex Optimization". *Automatica* **34**:5, pp. 585–601.

Dahlin, E. B. (1968). "Designing and tuning digital controllers". *Instruments and Control Systems* **41**:6, pp. 77–83.

Franklin, G. F., J. D. Powell, and A. Emami-Naeini (2010). *Feedback Control of Dynamic Systems*. 6th Edition. Pearson, Upper Saddle River, NJ.

Garpinger, O. and T. Hägglund (2008). "A Software Tool for Robust PID Design". In: *17th IFAC World Congress*. Seoul, South Korea.

Garpinger, O. (2009). *Design of Robust PID Controllers with Constrained Control Signal Activity*. Licentiate Thesis ISRN LUTFD2/TFRT--3245--SE. Department of Automatic Control, Lund University, Sweden.

Garpinger, O. and T. Hägglund (2014). "Modeling for Optimal PID Design". In: *19th IFAC World Congress*. Cape Town, South Africa.

Garpinger, O., T. Hägglund, and K. J. Åström (2014). "Performance and robustness trade-offs in PID control". *Journal of Process Control* **24**:5, pp. 568–577.

Grimholt, C. and S. Skogestad (2012). "Optimal PI-Control and Verification of the SIMC Tuning Rule". In: *IFAC Conference on Advances in PID Control*. Brescia, Italy.

Hägglund, T. and K. J. Åström (2004). "Revisiting the Ziegler-Nichols step response method for PID control". *Journal of Process Control* **14**:6, pp. 635–650.

Harmse, M., R. Hughes, R. Dittmar, H. Singh, and S. Gill (2009). "Robust optimization-based multi-loop PID controller tuning: A new tool and an industrial example". In: *7th IFAC International Symposium on Advanced Control of Chemical Processes, ADCHEM'09*. Istanbul, Turkey.

Hast, M., K. J. Åström, B. Bernhardsson, and S. P. Boyd (2013). "PID design by convex-concave procedure". In: *2013 European Control Conference*. Zürich, Switzerland.

Isaksson, A. and S. Graebe (2002). "Derivative filter is an integral part of PID design". *Control Theory and Applications, IEE Proceedings* **149**:1, pp. 41–45.

Kristiansson, B. and B. Lennartson (2002). "Robust and optimal tuning of PI and PID controllers". *Control Theory and Applications, IEE Proceedings D* **149**:1, pp. 17–25.

Kristiansson, B. and B. Lennartson (2006). "Evaluation and simple tuning of PID controllers with high-frequency robustness". *Journal of Process Control* **16** (2), pp. 91–102.

Lagarias, J. C., J. A. Reeds, M. H. Wright, and P. E. Wright (1998). "Convergence properties of the Nelder-Mead simplex algorithm in low dimensions". *SIAM Journal on Optimization* **9**, pp. 112–147.

Larsson, P.-O. and T. Hägglund (2011). "Control Signal Constraints and Filter Order Selection for PI and PID Controllers". In: *2011 American Control Conference*. San Francisco, California, USA.

Leva, A. and M. Maggio (2011). "A systematic way to extend ideal PID tuning rules to the real structure". *Journal of Process Control* **21**:1, pp. 130–136.

Marlin, T. (1995). *Process Control – Designing Processes and Control Systems for Dynamic Performance*. Chemical Engineering. McGraw-Hill, New York, USA.

McMillan, G. (1983). *Tuning and Control Loop Performance*. ISA, Research Triangle Park, NC.

Micić, A. D. and M. R. Mataušek (2014). "Optimization of PID controller with higher-order noise filter". *Journal of Process Control* **24**:5, pp. 694–700.

Nordfeldt, P. and T. Hägglund (2006). "Decoupler and PID controller design of TITO systems". *Journal of Process Control* **16**:9, pp. 923–936.

Oviedo, J. J. E., T. Boelen, and P. van Overschee (2006). "Robust Advanced PID Control (RaPID) - PID Tuning Based on Engineering Specifications". *IEEE Control Systems Magazine* **26**, pp. 15–19.

Panagopoulos, H., K. J. Åström, and T. Hägglund (2002). "Design of PID controllers based on constrained optimisation". *IEE Proceedings - Control Theory & Applications* **149**:1, pp. 32–40.

Rivera, D. E., M. Morari, and S. Skogestad (1986). "Internal Model Control. 4. PID Controller Design". *Ind. Eng. Chem. Process Des. Dev.* **25**, pp. 252–265.

Romero Segovia, V., T. Hägglund, and K. J. Åström (2013). "Noise filtering in PI and PID Control". In: *2013 American Control Conference*. Washington, D.C., USA.

Romero Segovia, V., T. Hägglund, and K. J. Åström (2014a). "Design of measurement noise filters for PID control". In: *19th IFAC World Congress*. Cape Town, South Africa.

Romero Segovia, V., T. Hägglund, and K. J. Åström (2014b). "Measurement noise filtering for PID controllers". *Journal of Process Control* **24**:4, pp. 299–313.

Sadeghpour, M., V. de Oliveira, and A. Karimi (2012). "A toolbox for robust PID controller tuning using convex optimization". In: *IFAC Conference on Advances in PID Control*. Brescia, Italy.

Shinskey, F. G. (1996). *Process-Control Systems. Application, Design, and Tuning*. 4th. McGraw-Hill, New York, NY.

Skogestad, S. (2003). "Simple analytic rules for model reduction and PID controller tuning". *Journal of Process Control* **13**:4, pp. 291–309.

Šekara, T. B. and M. R. Mataušek (2009). "Optimization of PID controller based on maximization of the proportional gain under constraints on robustness and sensitivity to measurement noise". *IEEE Transactions on Automatic Control* **54**:1, pp. 184–189.

Walters, F. H., L. R. Parker Jr, S. L. Morgan, and S. N. Deming (1991). *Sequential Simplex Optimization*. CRC Press LLC.

Zhou, K. and J. Doyle (1998). *Essentials of Robust Control*. Prentice Hall International, Upper Saddle River, NJ.

Paper V

Cascade control of the friction stir welding process to seal canisters for spent nuclear fuel

Lars Cederqvist Olof Garpinger Tore Hägglund
Anders Robertsson

Abstract

This article presents the development to reliably seal copper canisters containing the Swedish nuclear waste, using friction stir welding. To avoid defects and welding tool fractures, it is important to control the welding temperature within a span of 790 to 910°C. A cascade controller is used to efficiently suppress fast power input disturbances reducing their impact on the temperature. The controller is tuned using a recently presented method for robust PID control. Results show that the controller keeps the temperature within ±10°C during the 40 minute long joint line sequences. Apart from the cascaded control structure, good process knowledge and control strategies adapted to different weld sequences have contributed to the successful results.

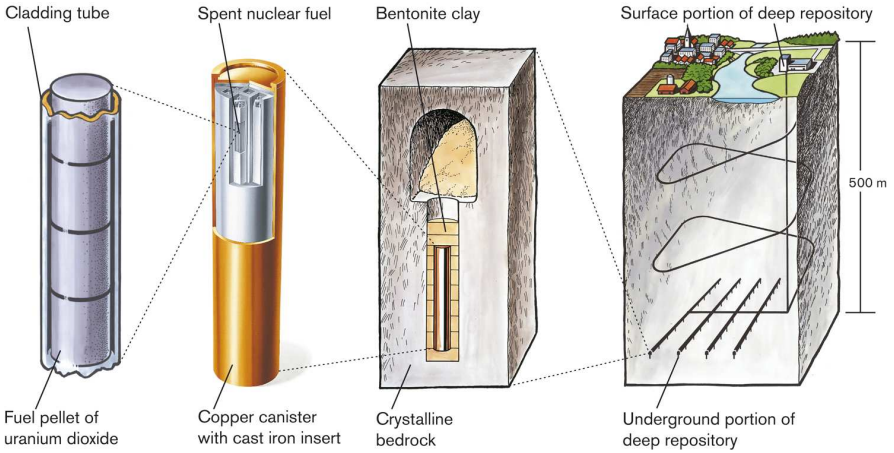

Figure 1. Protection barriers in the Swedish method for nuclear waste management.

1. Introduction

1.1 Background

The Swedish Nuclear Fuel and Waste Management Company (SKB) plans to join at least 12,000 lids and bases to the extruded copper tubes containing Sweden's nuclear waste, using friction stir welding (FSW) as described by [Thegerström, 2004]. The canisters produced (5 m height, 1 m diameter, made out of oxygen-free copper) are a major component of the Swedish system for managing and disposing of radioactive waste. They will be stored in the Swedish bedrock and must remain intact for 100,000 years. A corrosion barrier of 5 cm thick copper and a cast iron insert for mechanical strength are used to meet this requirement. Figure 1 illustrates the barriers that will jointly prevent the radioactive substances in spent fuel from spreading into the environment.

To ensure that high quality welds are produced repeatedly during more than 40 years of production, there is an evident need for automated welding instead of the past procedure depending on a skilled welding operator. A correctly tuned controller will not only be able to react to process disturbances faster, but also produce a more reliable process throughout the approximately 45 minute long weld cycles.

1.2 Friction stir welding

FSW is a thermo mechanical solid-state process that was invented in 1991 at The Welding Institute (TWI) in Cambridge, England [Thomas et al.,

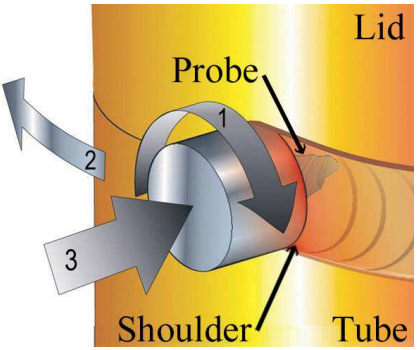

Figure 2. Illustration of the friction stir welding process to seal copper canisters.

1991]. A rotating non-consumable tool, consisting of a tapered probe and shoulder, is plunged into the weld metal and traversed along the joint line, see Fig. 2. For thick section welds (e.g. the copper canisters), a pilot hole has to be drilled to make the plunge sequence possible without significant probe wear. Frictional heat is generated between the tool and the weld metal, causing the metal to soften, normally without reaching the melting point, and allowing the tool to traverse the joint line. Deeper knowledge of FSW and its applications in industry can be gained from reading e.g. [Mishra and Mahoney, 2007], [Nandan et al., 2008] and [Lohwasser and Chen, 2009].

2. Process description

2.1 Technical data

In 2003, a purpose-built machine from ESAB was installed at SKB's Canister laboratory in Oskarshamn (see Fig. 3). In this machine, the welding head rotates up to 425 degrees around the canister, which is clamped with a force of 3200 kN. The lid is pressed down with a force of 390 kN.

All welds are carried out under force control, i.e. constant axial force (input 3 in Fig. 2) throughout the weld cycle. For more information on the machine, see e.g. [Ronneteg et al., 2006].

2.2 Available control and measurement signals

One reason for the fast (and growing) implementation of FSW in industry is that the method has few input variables. The input parameters, that can be modified during the copper canister welds, are listed according to Fig. 2; 1. tool rotation rate, 2. welding speed along the joint, and 3. axial

Figure 3. Purpose-built FSW machine for copper canister welding.

force (controls the position of the tool in relation to the canister surface, i.e. the tool depth). The measured variables are the welding (tool) temperature, the torque required by the spindle (that drives the tool rotation) to maintain the tool rotation rate, and the depth of the tool into the welded material.

There are also some elementary relationships between the signals that are important to understand. The tool rotation rate multiplied with the spindle torque is equal to the power input in units of kW. In addition to this, the heat input (J/mm) can be derived by dividing the power input with the welding speed. Both quantities are closely correlated to the welding (tool) temperature.

2.3 Argon shielding gas

[Upadhyay and Reynolds, 2010] investigated the effects of varying thermal boundary conditions by for example, welding under water. Similarly, the process stabilizing effect that argon gas has on the copper canister welds, was described by [Cederqvist et al., 2010a]. Today, all welds are produced in argon gas by placing a gas chamber over the tool. The main reasons for using argon as a shielding gas are to achieve a weld zone with minimal oxide inclusions and to get a more robust process, partly thanks to less shoulder wear. As a direct result of this, the spindle torque variations are smaller and the controller will thus have an easier task obtaining its objectives.

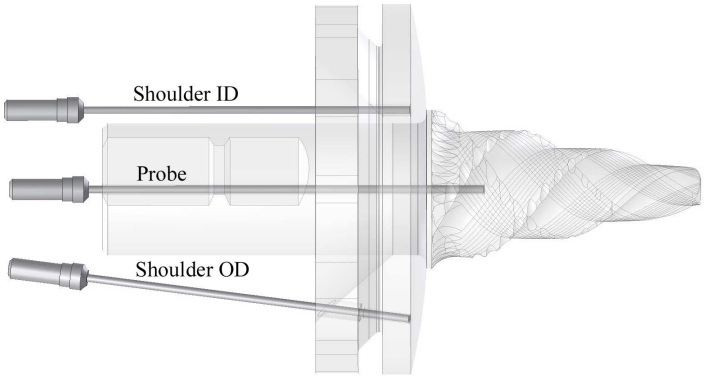

Figure 4. Placements of the three thermocouple sensors used to measure the welding temperature.

2.4 Temperature measurements and process window

Measuring the welding temperature is not a trivial matter. The sensors, so called thermocouples, should ideally be placed such that the temperature response has as little delay as possible, but they should also correlate well with weld quality. Typically, the sensors are placed in the welding tool. Figure 4 shows the tool in profile and the location of the three thermocouples currently used to measure the welding temperature (ID and OD stands for Inner and Outer Diameter, respectively). All three thermocouples have reaction times that are less than 620 ms, and calibration errors of 0.1%, i.e. $< 1°C$. The probe sensor is situated 10 mm from the surface, while the shoulder ID and OD sensors are only 1 and 2 mm away, respectively. It should also be mentioned that the probe material is a nickel-based superalloy called Nimonic 105. The shoulder material, on the other hand, is made of a tungsten alloy called Densimet D176. Depending on the sensor position and the probe/shoulder material, the measurement signals will have slightly different dynamic relations to the input signals of the process. In terms of delay, the probe temperature takes the longest time to react, followed by the shoulder OD and then the shoulder ID, which has the least dead time.

If the welding temperature gets too high, for a longer period of time, there is a risk for probe fracture resulting in a rejected canister with both extensive and expensive work to take out the nuclear waste. Similarly, too low temperatures may result in discontinuities in the weld (so called wormholes) that could, depending on size, also lead to a rejected canister. It is, in other words, very important to keep the temperature within this, so called, process window. The process window for FSW on the copper can-

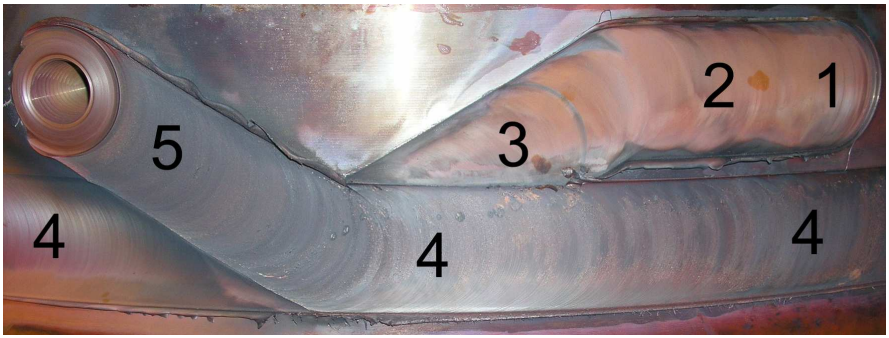

Figure 5. Sequences during a full weld cycle.

isters is roughly between 790 and 910°C (probe sensor), which has been determined through minimum and maximum temperature tests together with non-destructive testing [Ronneteg et al., 2006].

2.5 The weld cycle

The simplest weld cycle (in terms of constant thermal boundary conditions and controller requirements) would have been if the weld had started and ended at the joint line. However, since a probe-shaped exit-hole is left when the tool is retracted, the weld cycle needs to end above the joint line where it will not affect the 5 cm thick corrosion barrier. In addition, the weld is started above the joint line to further reduce the risk of defect formation at the joint line. This also makes it possible to abort the process in case of malfunction during start-up. Another weld will then be made in a new pilot hole without rejecting the canister.

A full circumferential weld can be divided into several different sequences as illustrated in Fig. 5. The sequences are:

1. *The dwell sequence*, that is used to bring the welding temperature high enough for the tool to start moving
2. *The start sequence*, in which the tool is accelerated to a constant welding speed and run until achieving a welding temperature close to the reference value
3. *The downward sequence*, where the tool moves down to the joint line
4. *The joint line sequence*, in which the tool runs along the joint for 360 degrees
5. *The parking sequence*, where the tool moves back into the lid so it can be withdrawn

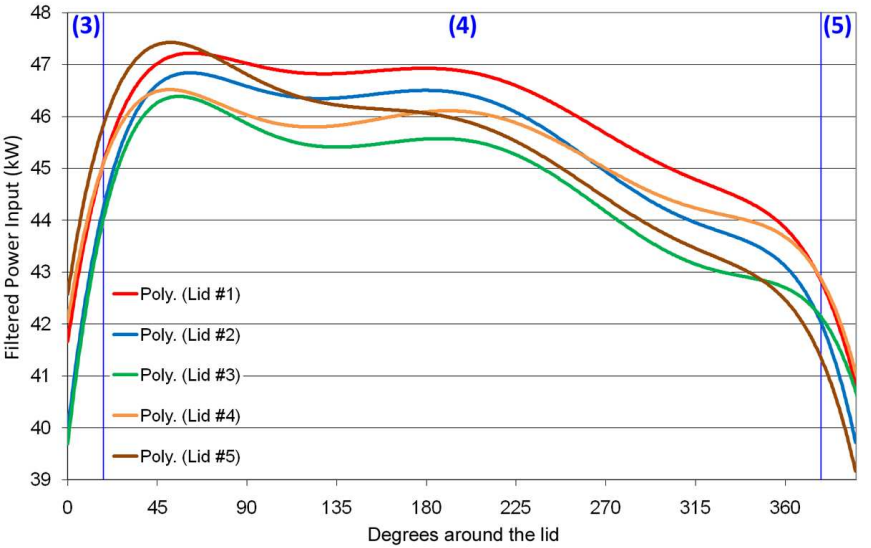

Figure 6. Varying power input requirements for multiple circumferential welds, using manual control. The different sequences are indicated by vertical lines and numbers in parentheses.

These will be indicated by a number from 1–5, in parenthesis, every time a sequence is referred to. This includes some of the plots.

The different sequences in a weld cycle results in non-uniform thermal boundary conditions throughout the weld. As a consequence, the power input requirement to keep the welding temperature within the process window varies throughout the 45 minute long weld cycle (see Fig. 6).

2.6 Known process disturbances

During a full weld cycle, there are three types of disturbances typically encountered. Two of these can be directly related to the temperature signal, while the third is spotted in the torque measurements. These disturbances (including the variable thermal boundary conditions due to the path of the weld cycle) motivates why closed-loop control of the process is needed.

Temperature disturbances The first type of temperature disturbance is associated with the tool moving to or from areas that have been significantly heated already. During, e.g., the dwell sequence (1) the temperature increases while the tool still have not started moving. This will mainly affect the immediate area around the tool which is later left behind when the tool starts moving. Such disturbances are also encountered during the joint line sequence (4) as the tool constantly moves towards

a warmer area after approximately half the cycle. Due to the relatively slow welding speed (86 mm/min), this kind of disturbance is rather low frequent in nature. Figure 6 shows how the power input varies during 5 full weld cycles in order to keep a constant probe temperature. Note especially how the mentioned temperature disturbance requires the power to drop (in order to keep a constant temperature) from around 200 degrees around the lid and forward. In addition to this, the power input requirement profile will vary from weld cycle to weld cycle according to the same figure. Reasons for these variations can be; different properties in the manufactured components (tubes and lids), wear in spindle gear and/or replaced spindle (necessary several times during a production period of 40+ years). As a result, it is not advisable to control towards a preset power input requirement for every weld cycle like the early proposed regulator [Cederqvist et al., 2009], but to have an adjustable power input requirement.

The second type of temperature disturbance occurs during the downward and parking sequences (3 and 5) and is caused by greater heat conduction at the joint line compared to the lid. A consequence of this is that the power input will have to increase by a fair amount during the downward sequence (3). Similarly, during the parking sequence (5), the power input will have to drop instead. Together with the first type of temperature disturbance, this explains why the power inputs will need to drop rather fast at the end of the welds in Fig. 6 (360 degrees and onwards). Even though these conduction disturbances are rather slow, they will still have quite an impact on the temperature profile. The main reason is that they are relatively large in magnitude. The results of this will be described thoroughly in Section 5.3.

Torque disturbances The torque required by the spindle to maintain the tool rotation rate will vary depending on the properties of the material. The tool is, e.g., more likely to penetrate a bit deeper into the copper in areas that have been significantly preheated, thus resulting in a higher torque value. The slightly different characteristics of the tube and lid will also give rise to such torque variations that will primarily affect the power input, but secondarily also the welding temperature. While these disturbances appear in all five sequences, it is only during the joint line sequence (4) that they are relatively insignificant. Figure 7 shows an abrupt torque disturbance that takes place during the downward sequence (3) (occurs at time $t = 127$ seconds in the figure, which is 13 seconds before the joint line sequence (4) starts). If such a disturbance is not compensated for, by changing the tool rotation rate, the power input will drop by approximately 8%, causing the probe temperature to change in the order of -40°C. While this particular disturbance is both faster and

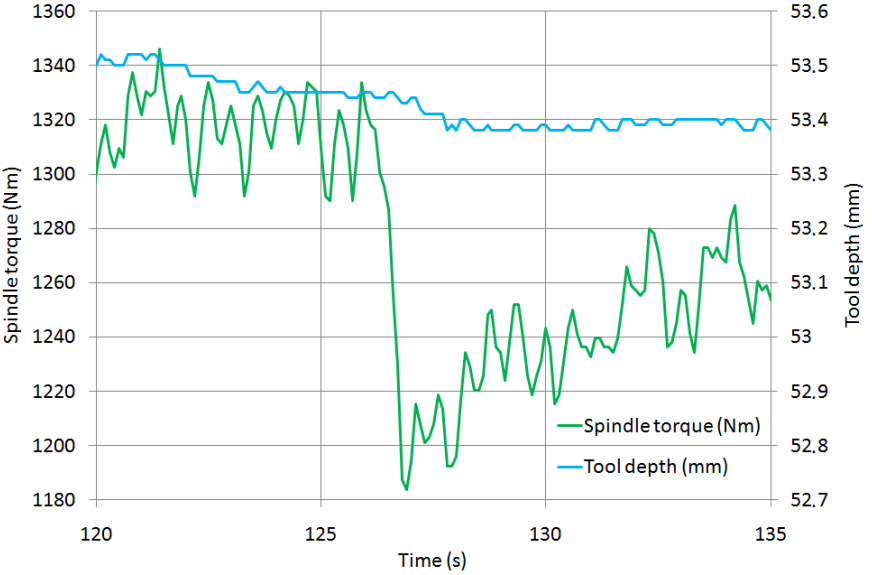

Figure 7. Torque disturbance during the downward sequence (3). The tool depth signal confirms that the torque disturbance is not related to a sudden change in the tool position.

of larger magnitude than most torque variations, it still gives a hint of what one can expect will happen occasionally during production. Considering that 1 degree in Fig. 6 corresponds to 6 seconds, it is also evident how much faster these disturbances are compared to their temperature counterparts. In Section 5.3, Fig. 18, it is shown how well this particular disturbance was handled by the current controller (in terms of power and temperature changes).

3. Controlling the FSW process

3.1 Background on FSW control

As of today, most friction stir welds are made on plates with more uniform thermal boundary conditions than cylinders (as well as other complex geometries), and in aluminum, that has a relatively large process window, compared to steel, nickel and titanium alloys. For these reasons, there are currently only a small (although growing) amount of FSW applications using temperature control. The welding temperature is, for instance, rarely even measured. Instead, a welding operator might just decrease ax-

ial force if the tool depth increases as the tool and plate get hotter at the end of a long weld cycle.

Robotic FSW applications face several challenges that the process described in this article does not share; e.g. axial force control that is influenced by limited force capacity and deflection in the linkages and joints of the robot (see e.g. [Soron, 2007]). [Longhurst et al., 2010], on the other hand, use torque control to control tool depth instead of axial force control. With this method, they also present an alternative way of controlling the power input and thus also the temperature indirectly, although the axial force remains uncontrolled.

[Fehrenbacher et al., 2008] have approached closed-loop control of the welding temperature through non-uniform boundary conditions (varying backing plate materials) by manipulation of the welding speed, and more recently by adjusting tool rotation rate [Fehrenbacher et al., 2010]. A new study by [Mayfield and Sorensen, 2010] suggests development of a cascaded control strategy similar to those presented by [Cederqvist et al., 2009] and in this article. [Wang et al., 2004] control the spindle torque within limited values by adjusting the depth of the tool into the welded material.

Furthermore, the need for reliable control of the welding temperature should increase as FSW is used on metals like titanium and steels that will have much smaller process windows. In addition, more complex geometries of welding objects will result in non-uniform thermal boundary conditions and thus also a need for welding temperature control.

3.2 Past and present control of the current process

While open-loop control may be a suitable solution for aluminum plate welding, it is just not sufficient for welding the copper canisters described in this article. Such an approach would suffer from the many disturbances occurring during a full weld cycle, possibly bringing the welding temperature outside the process window. A series of 20 high quality welds of the copper canisters have, however, already been produced thanks to manual changes of the tool rotation rate by a skilled welding operator [Cederqvist and Öberg, 2007]. Still, to not be dependent on such an operator for 40+ years of production, an automatic welding procedure needs to be developed. A well-tuned controller will, in addition, react faster to disturbances and be more reliable and repeatable than a welding operator.

A previously used cascaded control strategy, controlling both towards a fixed power input and temperature reference, was examined by [Cederqvist et al., 2009]. The current control strategy, presented in this article, has a clearer control design method as well as a better suited cascade structure (further described in Sections 3.5 and 3.6).

3.3 Controller objectives

The main objective of the controller is to minimize the risk of the temperature ending up outside the process window. It is particularly crucial that the probe does not break and, therefore, necessary to keep the probe temperature below 910°C. The temperature disturbances occurring during the downward sequence (3) will, however, generally make it easier to end up close to the lower limit of the window rather than the upper. During the parking sequence (5), the opposite will hold. Taking all these observations into consideration, it has been decided to generally use a temperature reference slightly below the middle of the process window. Another option would be to vary the set-point depending on the sequence. This is utilized during the parking sequence (5).

To make sure that the controller can fulfill its main objective, it is important that it can suppress disturbances in the spindle torque and temperature efficiently. In addition, the controller should be able to respond quickly enough to reference tracking during the start-up and parking sequences (1–3 and 5) without experiencing any significant under or overshoots.

3.4 Choice of control- and measurement signals

A previous study by [Cederqvist et al., 2008] investigated the significance of tool rotation rate, axial force and welding speed on the most important output variable, the welding temperature (measured in the probe). It was clear that changes in the tool rotation rate had the largest effect, which makes sense since it is directly proportional to heat and power inputs (defined in Section 2.2). Additionally, it was noted that power input had better linear correlation with welding temperature than heat input. As a direct result, changes in welding speed and/or axial force have been deemed less useful to control the welding temperature and are, therefore, held constant. Changes in the tool rotation rate are instead used to control the welding temperature (and power input).

[Cederqvist et al., 2009] proposed that the controller should use the fastest responding tool temperature measurements, i.e. from the shoulder ID sensor (see Fig. 4). This signal has a time delay of approximately 1 second to disturbances in the power input, while the probe temperature, on the other hand, has about 5 seconds delay. From a control theoretical perspective it is important to have as little dead time as possible in the process [Åström, 2000]. However, since the probe temperature measurements, to current knowledge, have the best correlation with regards to probe fractures and wormhole defects [Cederqvist, 2011], it was decided to let the controller use this signal rather than the less delayed ones. In addition, it has been found that the repeatability of the three tempera-

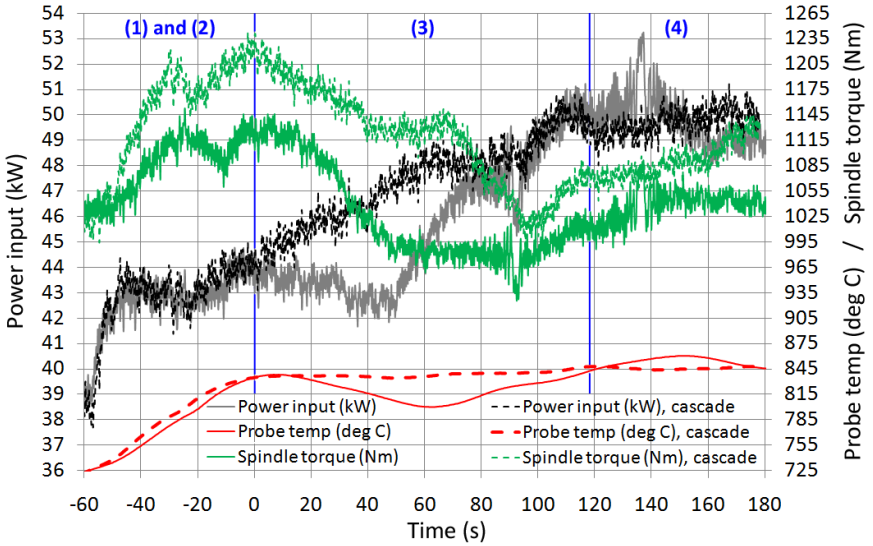

Figure 8. Results of single-loop (solid) versus cascade control (dashed). The different sequences are indicated by vertical lines and numbers in parentheses.

ture measurements, between weld cycles, is not adequate [Cederqvist et al., 2010b]. For example, even if the steady state probe temperature is 845°C two welds in a row, the shoulder ID and OD measurements can vary by at least 40°C between the two weld cycles. Therefore, it would be risky to control towards a fixed temperature reference on any of the other two sensors. For these two reasons, the shoulder measurements will only be used as back-up signals, and a future controller should be able to switch over to these readings if necessary.

One option would be to only use temperature measurements in the controller. This would result in a single-loop controller with simplicity as its main advantage. Figure 8 shows the probe temperature, power input and torque signals from two welds on the FSW process. The downward sequences (3) start at 0 seconds and until then both welds are controlled in the same way, but thereafter the solid curves expose what can happen if a single-loop controller is used. The fast torque disturbances are spotted much too late in the temperature signal and the controller will thus not be able to counteract these before they have already had a major impact on the probe temperature. The probe temperature drops to 800°C for the single-loop control, while the cascade control gives a minimum probe temperature of 835°C, to be compared with the reference value 845°C.

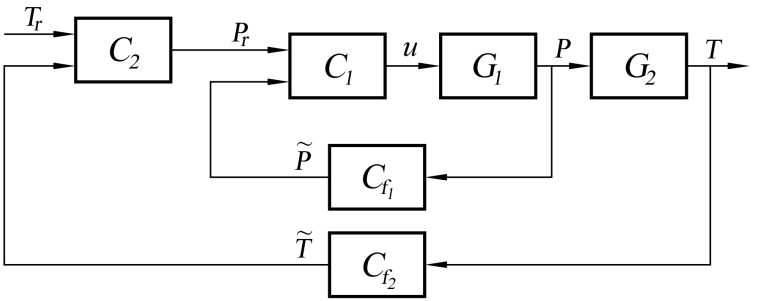

Figure 9. The proposed cascaded control structure.

The dotted curves, on the other hand, show the potential of using feedback of the power input signal. The cascade controller used in this weld is the one presented in the next section. A possible challenge with the use of a cascaded strategy is that the power input signal is rather noisy. If the controller design is made without taking this into consideration, one could risk ending up with a very high control signal variance, described by e.g. [Garpinger, 2009]. While an aggressive controller may give optimal performance, it could also result in unnecessary wear and tear on the equipment. Section 3.6 will include a deeper discussion on control trade-offs.

3.5 Cascade control of the FSW process

A cascaded control strategy seems ideal for the characteristics of this specific FSW application with its fast, multiplicative, torque disturbances and slower temperature counterparts. It has, therefore, been decided to use the controller structure displayed in Fig. 9. The process has been divided into two subsystems. Process G_1 holds the dynamics from the tool rotation rate (i.e. the control signal), u (rpm), to the power input, P (kW). This system will mainly hold the servo characteristics of the spindle motor. The outer process, G_2, on the other hand, describes how the power input is related to the probe temperature, T (°C). If needed, the measurements can be fed through low-pass filters, $C_{f_1}(s)$ and $C_{f_2}(s)$ respectively, for a less noise sensitive closed loop.

The controllers C_1 and C_2 will be tuned in accordance with the control objectives, such that disturbances are handled efficiently. One can assume that the power input reference, P_r, will be low-frequent enough to not set any major challenges on the servo qualities of the inner loop. Set-point tracking will thus mainly be an issue for the outer loop. In case the outer, load disturbance optimized, controller is not able to handle tracking well enough, it will be handled separately through e.g. set-point filtering.

3.6 Controller design principle

The decision of how to best choose controllers C_1 and C_2 depends on the dynamics of G_1 and G_2. Since initial step response tests on the FSW process have shown that both processes have relatively simple dynamics (the final models are presented in Sections 5.1 and 5.2), the choice fell on either PI or PID controllers. These controllers also have the advantage that they are easier to tune and understand than more advanced alternatives. The tuning of the controllers will be very similar to that of a newly presented method for design of robust PI- and PID-controllers with constrained control signal activity [Garpinger, 2009]. The technique is based on optimization and built around a MATLAB® software tool, first described by [Garpinger and Hägglund, 2008]. The software uses a linear transfer function model to find the PI or PID controller (either continuous or discrete time) minimizing the integrated absolute error (IAE)

$$\text{IAE} = \int_0^\infty |e(t)|dt, \tag{1}$$

during a unit step disturbance on the process input. The control error, e, is the difference between the controlled variable and its reference value.

The IAE optimization guarantees stability and is constrained by H_∞ robustness conditions

$$|S_s(i\omega)| \leq M_S, \quad |T_s(i\omega)| \leq M_T, \quad \forall \omega \in \mathcal{R}^+, \tag{2}$$

$$|S_s(i\omega_S)| = M_S \text{ and/or } |T_s(i\omega_T)| = M_T, \tag{3}$$

where $S_s(s)$ and $T_s(s)$ are the sensitivity function and the complementary sensitivity function respectively. The conditions stated in Eq. (3) force at least one of the constraints to be active (in the frequency points ω_S and/or ω_T) and will, therefore, simplify the optimization. These constraints are known to set closed-loop robustness towards process variations, disturbances and non-linearities as described by for example [Åström and Hägglund, 2005]. M_S and M_T are by default set to 1.4, known to give good robustness (e.g. phase margin 41.8° and gain margin 3.5), but they can also be changed by the user if desired.

This PID design method is similar to some earlier presented methods by [Panagopoulos et al., 2002], [Hägglund and Åström, 2004] and [Nordfeldt, 2005]. The biggest difference is the use of the fast software that is also robust, easy to use and freely available on the web.

The software can be used as described by [Garpinger, 2009] to limit the control signal sensitivity to measurement noise by tuning of a low-pass filter acting on the measurement signal (like C_{f_1} and C_{f_2} do). This

sensitivity is measured in terms of relative variance between the control signal and measurement noise

$$\frac{\sigma_u^2}{\sigma_n^2} \leq V_k. \tag{4}$$

V_k is a user-specified upper limit on the maximum allowed gain of the control signal variance, σ_u^2, due to the measurement noise (with variance σ_n^2). The method suggests that the controller design is then a trade-off between maximum allowed wear on the equipment, robustness and control performance. If the equipment is sensitive, one will simply have to rely on a slower closed loop.

4. Experimental setup

4.1 Modeling and control design procedure

A significant part of the work behind this article was done to make sure that the procedure for process modeling and controller design are both simple and intuitive. The FSW process is expected to operate for more than 40 years and it is likely to change during this time period. It is, therefore, wise to retune the controllers either when the FSW process has known modifications or if the control starts to perform worse (e.g. determined through monitoring). A simple design method should enable people in the industry, without deep knowledge in automatic control theory, to carry out the steps needed to tune the controller if necessary. Clear instructions and an intuitive strategy are also important for making sure that the necessary competence can be maintained within the company.

Modeling method and challenges Step response tests have been utilized for modeling the FSW process (i.e. determine G_1 and G_2). Two advantages of the method are that it is fast and simple. Both are properties that make it frequently used in the process industry today. A couple of its downsides are that it is disturbance sensitive and insufficient for identification of higher order systems. But, the FSW process can most likely be modeled well using only low-order models and it will also be modeled during the joint line sequence (4), when disturbances are less significant.

The FSW process is, however, also partly irreversible when using a constant axial force. A somewhat simplified explanation of the behavior will be presented here. For example, if the weld metal gets colder it will result in less tool depth and hence a lower power input. This will make the weld even colder, and so forth. Similarly, if the process gets warmer it will lead to a softer material, a greater tool depth and thus result in an even hotter process. This theoretical behavior, sketched in Fig. 10 (dashed

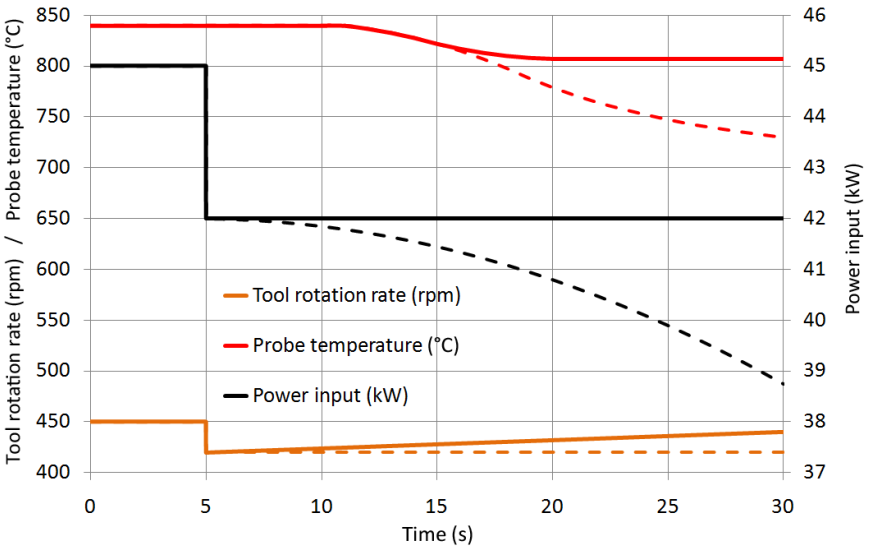

Figure 10. Theoretical sketch of the process behaviors to open-loop (dashed) and closed-loop (solid) power input changes.

curves), makes it rather hard to use step response tests to determine a model directly from tool rotation rate, u, to welding temperature, T. However, since the power input responds much faster to changes in the tool rotation rate than the temperature, it is still possible to get around these problems by closing the inner loop. This way, it will be possible to do well-behaved step responses from power input reference, P_r, to temperature, T, similar to the solid curves in Fig. 10.

Controller design procedure The steps below follow the currently recommended procedure for tuning of the controllers C_1 and C_2.

1. Do step response tests from u to P.

Make several steps of different magnitudes in both up- and downward direction to capture all relevant dynamics and possible non-linearities.

2. Find a linear model of G_1.

Use any method available to estimate a model from step response data. See e.g. [Åström and Hägglund, 2005] for some possible methods.

3. Extract noise data from the power input signal.

Make sure there are no trends left in the resulting noise series.

4. Design a robust PI or PID controller, C_1 (possibly with a low-pass filter C_{f_1}), for the inner loop.

Use e.g. the previously mentioned PID design tool to determine a robust controller, taking the trade-off between relative noise level, V_k, and performance, IAE, into consideration.

5. Close the inner loop and do step response tests from P_r to T.

Steps should be made in both up- and downward direction to reveal possible non-linearities and differences in the dynamics.

6. Derive a linear model of G_2 from the step response data.

Use the power input and temperature data to derive the model.

7. If relevant, get representative noise data from the temperature measurements.

Since the temperature measures contain very little noise, it could be that the noise throughput to P_r is so small that it literally drowns in the noise of the power measurements. If not, the procedure will be the same as for step 3.

8. Design a robust PI or PID controller, C_2 (possibly with a low-pass filter C_{f_2}), for the outer loop.

The design of C_2 will be similar to the tuning of C_1.

While some of these steps are specific for the used PID design method, the overall procedure is far from uncommon. Similar tuning schedules for cascade controllers are proposed by for instance [Forsman, 2005] and [Hägglund, 2008].

4.2 Performance test procedures on the FSW process

Once the two controllers have been derived, their performance will be trialled through two types of tests. The most common of these tests is the start-up test, which normally run until about 60 seconds into the joint line sequence (4). The main reason why these tests are so common is that the dwell, start and downward sequences (1–3) have shown to be the most difficult to handle from a control perspective (see Section 2.6). Throughout the start and downward sequences (2 and 3), the temperature disturbances will act to increase the power input needed to maintain the temperature at its set-point. These tests will determine how well the cascade controller acts to suppress the temperature reducing elements. The disturbances do, however, not come instantly once a new sequence has begun. So, while the controller should counteract temperature decreases,

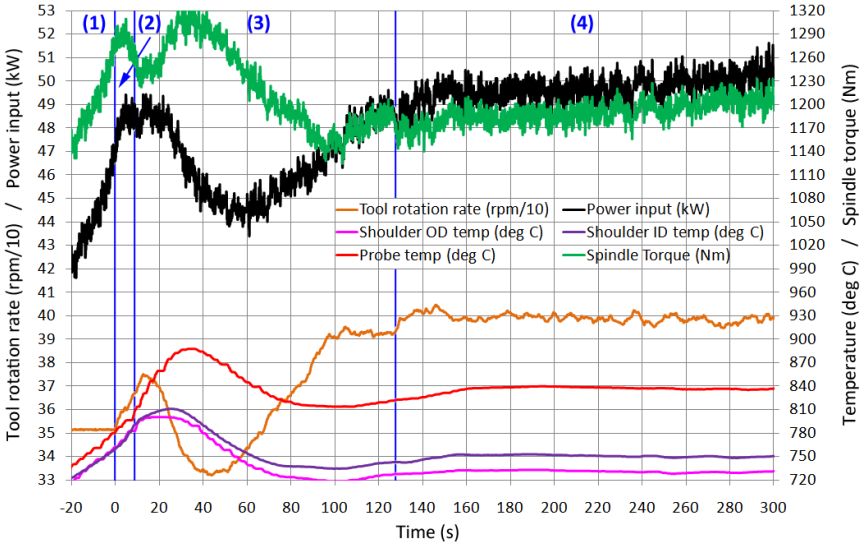

Figure 11. Results of high power input during start-up. The different sequences are indicated by vertical lines and numbers in parentheses.

it must not let the power input reach too high magnitudes before the start and downward sequences (2 and 3) begin. This is especially crucial just before the downward sequence (3) when the temperature is close to its set-point. Figure 11 shows the results of a weld start-up where the power input got too high before the downward sequence (3), resulting in a temperature overshoot of 48°C. In an act to eliminate this unwanted feature, the power controller alone will be used before the downward sequence (3). The start-up is, therefore, currently handled according to the following 3 steps:

1. Once the temperature reaches a first limit, the inner regulator starts controlling towards a constant set-point, P_r.

2. A second temperature limit indicates when the tool should start moving. At this moment, the power input reference, P_r, will start to increase slowly along a linear slope.

3. The downward sequence (3) will start once a third temperature limit has been reached (typically about 10°C from its set-point, T_r). This is when the cascade controller is finally turned on.

Results showing the benefits of this new strategy are presented in Section 5.3.

The second type of tests are the circumferential welds. The main reason for not doing too many of these is because the copper components are expensive. They will, however, capture all sequences and be representable of the final application. The start-up is handled as previously described. The joint line and parking sequences (4–5) use the cascade controller. The temperature set-point is, however, lowered during the parking sequence (5) in order to counteract the temperature and torque disturbances occurring there.

5. Results and discussion

5.1 Design of the power input controller

Looking at weld data, it is safe to say that control signal changes affect the torque much slower than they affect the spindle motor servo. The dynamics of the servo are visible in measures of the tool rotation rate, also used to calculate the power input. See e.g. Figure 12 that shows the tool rotation rate and torque responses to a step change in the control signal (at 0 seconds). While the tool rotation rate has a distinct change, the torque signal is quite random. It is thus justified to only include the dynamics of the motor servo in G_1. The torque is assumed constant and the static gain of the process will be determined by its magnitude.

In this work, two methods were utilized for modeling; comparisons between simulation and real data, as well as an optimization method called TRA (transient response analysis) [Wallén, 2000]. From looking at the step responses, it was concluded that a second-order model with well damped poles would suit the data. Since TRA does not support this type of processes, simulation comparisons were used. Figure 13 shows the tool rotation rate offsets during four different step response tests together with simulated step responses of the model

$$\frac{4.6^2}{s^2 + 2 \cdot 0.8 \cdot 4.6s + 4.6^2}, \tag{5}$$

that has a damping ratio of 0.8 and poles in $s = -3.68 \pm 2.76i$. The simulation results show a good overall match with the real weld data. It should be added that the motor servo had a steady state error of about -2.3 rpm during the step tests. This control error will, however, neither affect the modeling nor the resulting inner-loop control.

The torque signal will normally vary somewhere between 1000 and 1400 Nm after the start sequence (2) has begun. When the data in Fig. 13 was collected, the torque was close to 1170 Nm. This is actually a very representative torque level for the joint line sequence (4) and it is also

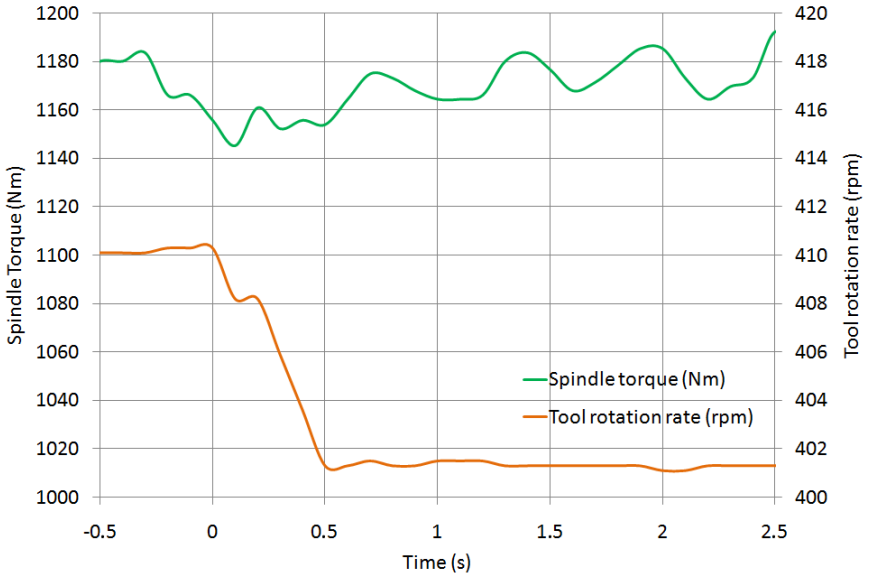

Figure 12. Changes in the tool rotation rate and torque signals to a step change in the control signal.

fairly in the middle of the interval stated above. The power input, P (kW), can be determined by

$$P = \frac{\pi}{30000} \cdot u \cdot \tau, \tag{6}$$

where τ is the torque in (Nm) and u is the tool rotation rate in (rpm). Since the model (5) is described in u, a static gain of 0.12 seems suitable for the process G_1, such that the final model becomes

$$G_1(s) = 0.12 \cdot \frac{4.6^2}{s^2 + 2 \cdot 0.8 \cdot 4.6s + 4.6^2}. \tag{7}$$

Due to the torque variations, however, the gain can vary between 0.105 and 0.145.

A time series of measurement noise data was then gathered by cropping parts of the step responses and then piecing them back together after applying a linear detrending. This data could then be used in closed-loop simulations to determine the V_k-value (see Eq. (4)) for each new controller design.

As described in section 4.1, it is a good idea to make the tuning as easy as possible, as long as the controller objectives are met. Therefore, it makes sense to start out optimizing PI controllers, rather than PID,

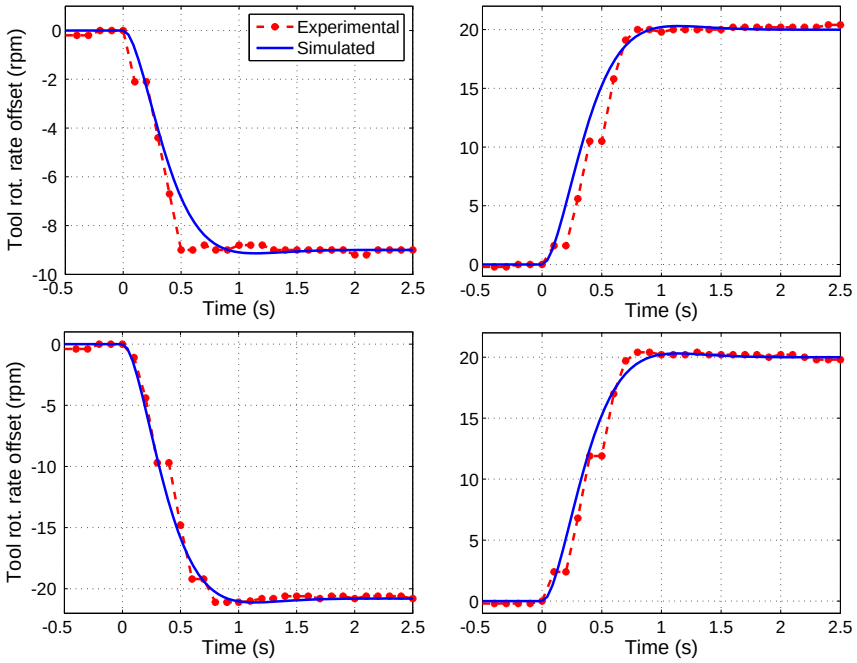

Figure 13. Experimental (dashed) and model (solid) step responses in the tool rotation rate during welding and simulation. The offsets were all close to 400 rpm.

for the process. Using Garpinger's MATLAB® tool on the process model in Eq. (7) showed that tuning of a low-pass filter together with the PI controller had little effect on the relative noise throughput, V_k (previously defined in Section 3.6). As a way to still be able to set V_k, [Garpinger, 2009] has suggested that one can instead vary the robustness measures M_S and M_T. Therefore, 10 different PI controllers, with 10 different M_S- and M_T-values (from 1.015 to 1.35), were tuned using the design tool. Figure 14 shows the trade-off curve between performance (IAE, previously defined in Section 3.6) and noise sensitivity (V_k) for these controllers. While a fast controller results in good performance (low IAE-value), it will also lead to a noise sensitive system (high V_k). A slower closed loop will, on the other hand, give too bad performance. The key in the design is to try choosing a controller somewhere around where the curve bends the most. If the controller is made faster, one will not benefit that much in better performance compared to how much is lost in noise sensitivity. If the closed loop is made slower, the noise sensitivity does not change much,

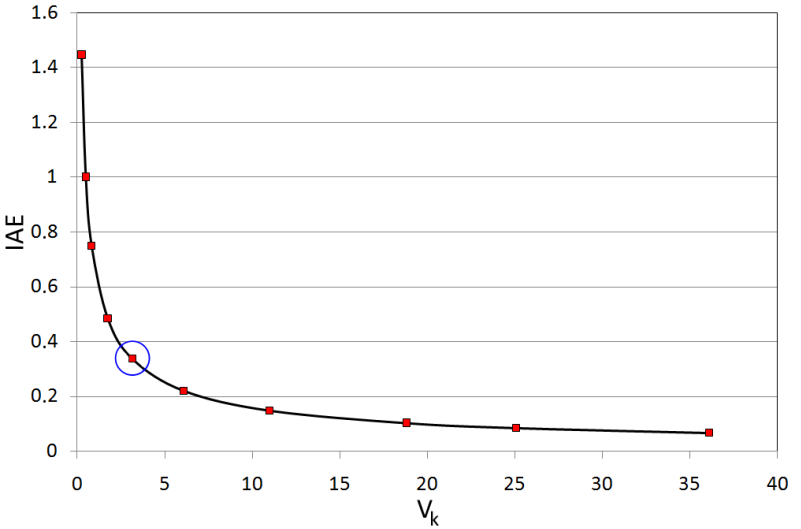

Figure 14. Performance and noise sensitivity trade-off curve for the inner control loop. The chosen controller is marked with a blue circle.

while the performance rapidly deteriorates. The controller

$$C_1(s) = 1.07\left(1 + \frac{1}{0.36s}\right), \tag{8}$$

has an IAE-value of 0.34, a V_k-value of 3.15 (marked with a blue circle in Fig. 14) and $M_S = M_T = 1.065$ (very good robustness). It is thus close to where the trade-off curve bends the most at the same time as the proportional gain is close to 1. The latter argument suggests that the controller will prevent unnecessary amplification of quick changes in the measurement noise. The inner closed loop has a bandwidth of 0.37 rad/s.

In the end, it was decided to use this PI controller over a PID. While a PID regulator may result in slightly better performance, it is not justified to add the extra complexity here. In section 5.3 it will be evident why the inner loop does not need to be tuned any faster. It is fast enough for handling the torque disturbances.

Furthermore, it should be pointed out that the static gain variations in G_1 will have little effect on the power control since the closed-loop robustness is as good as it is. Also note that all controllers derived in this article had to be discretized in order to implement them. This was done using the forward Euler method on the integral part, with a sampling time of $h = 0.1$ s. All experiments presented here were thus carried out using this sampling time.

5.2 Tuning the temperature controller

Activating the inner-loop control, it is possible to do step response tests from power input reference, P_r, to probe temperature, T. Several step changes, of different magnitudes, were carried out in the vicinity of multiple temperatures (from 815 to 845°C). Looking at the data, the process dynamics seem to vary a bit, possibly both with temperature and direction of the step. The reason why the process is direction dependent is because the control signal can only actively influence the heating of the process. For practical reasons, all step response tests have been made during the joint line sequence (4). So, while it is useful to know how the process dynamics depend on temperature, it is more important to know the model around the temperature reference, T_r. Figure 15 shows responses to a power input step in upward direction (at 0 seconds) that was made close to the temperature set-point. Running the step response data through TRA gave the following process model

$$G_2(s) = \frac{11.6}{(7s+1)^2} e^{-5s}. \tag{9}$$

The simulated model output is marked as a dashed blue curve in Fig. 15. Controller design on the model showed that it gives good robustness and almost unchanged performance for all process models derived (i.e. the ones modeled at different temperatures and in downward direction too).

Note that G_2 is both of order two and has a rather significant time delay. These characteristics could possibly be explained by the fact that the probe temperature sensor is situated inside the tool. Therefore, the tool has to be warmed up before the sensor can react to changes in the welding temperature. The reason why the probe sensor has a larger delay than the other two sensors is because the probe material has a lower thermal conductivity than the shoulder material, and the shoulder thermocouples are placed closer to the weld metal.

It was decided to use PI control for the outer loop as well. As will be shown in the next section, it is quite enough for achieving the control objectives. For FSW applications with narrower process windows, however, a PID controller may be useful in the outer loop.

The probe temperature measures hardly contain any noise at all. There is, therefore, no reason to take the noise sensitivity into account when designing C_2. Especially after choosing a PI controller to do the job. The low-pass filter is thus also left out. The trade-off between robustness and performance will instead be the important part of the outer control-loop design.

Two controllers were tuned using the PID design software, one with

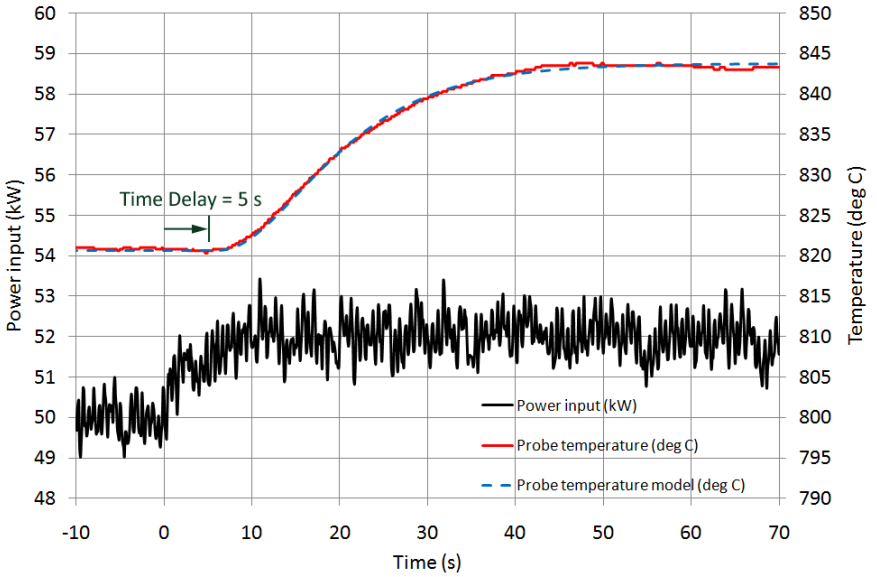

Figure 15. Step test (at 0 seconds) from power input reference to probe temperature. A model simulation gives the dashed, blue, curve.

$M_S = M_T = 1.4$ and the other with $M_S = M_T = 1.8$. The first controller

$$C_2(s) = 0.033 \cdot \left(1 + \frac{1}{11.2s}\right), \tag{10}$$

is used during the joint line sequence (4) and gives a closed-loop bandwidth of 0.064 rad/s. The second, more aggressive, controller

$$C_2(s) = 0.065 \cdot \left(1 + \frac{1}{14.0s}\right), \tag{11}$$

is active during the downward and parking sequences (3 and 5) and results in a closed-loop bandwidth of 0.14 rad/s. The reason for using a faster controller at those stages is because of the temperature disturbances active there (described in Section 2.6).

5.3 Performance experiment results

Start-up tests Figure 16 shows four weld start-ups (probe temperatures in (a), power inputs in (b)) with the downward sequences (3) starting at 0 seconds. These were produced using open-loop (dwell sequence (1)) and cascade control (start, downward and joint line sequences (2–4)) of the temperature, similar to the weld in Fig. 11. The temperature reference

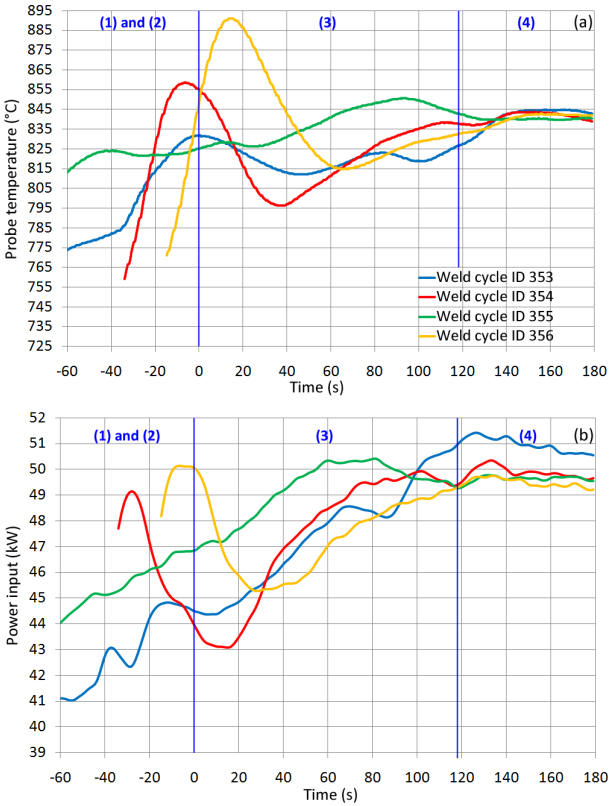

Figure 16. Four old weld start-ups showing bad repeatability in probe temperature (a) and power input (b). The different sequences are indicated by vertical lines and numbers in parentheses.

was set to 840°C. As one can see, this gave very bad repeatability between the welds, with a power input that varied between 44 and 50 kW before the downward sequence (3) had even started. As a result, the probe temperature dropped below 800°C during one downward sequence (3) while another had a peak temperature above 890°C. While this is within the limits of the process window, it is still a bit too close to really be on the safe side.

Making the changes to the control switches described in Section 4.2, four new start-up tests were run and these are presented in Fig. 17 (probe temperatures in (a), power inputs in (b)). The power input control (before 0 seconds) clearly makes the first three sequences more repeatable, producing welds that are within a satisfactory interval of 830 to 850°C.

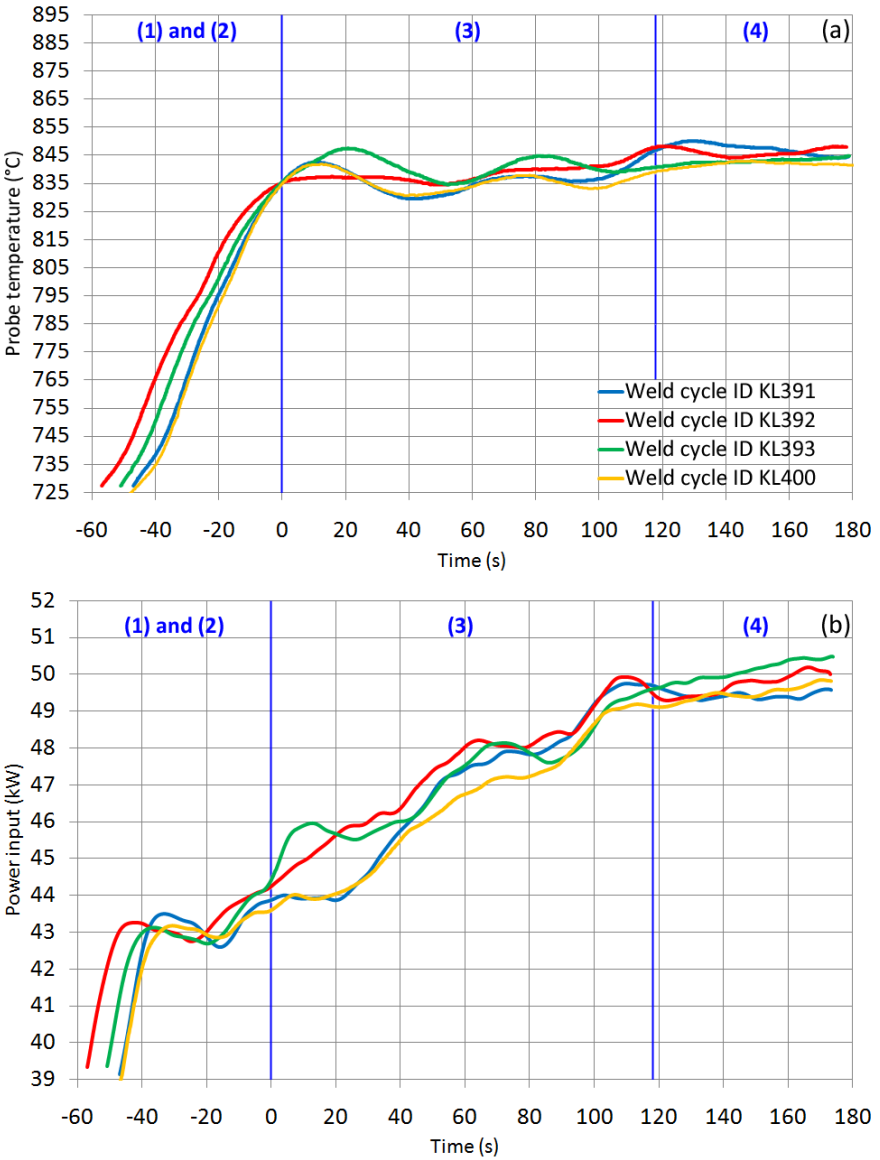

Figure 17. Four new start-up tests using an improved control strategy, thus giving better repeatability in probe temperature (a) and power input (b). The different sequences are indicated by vertical lines and numbers in parentheses.

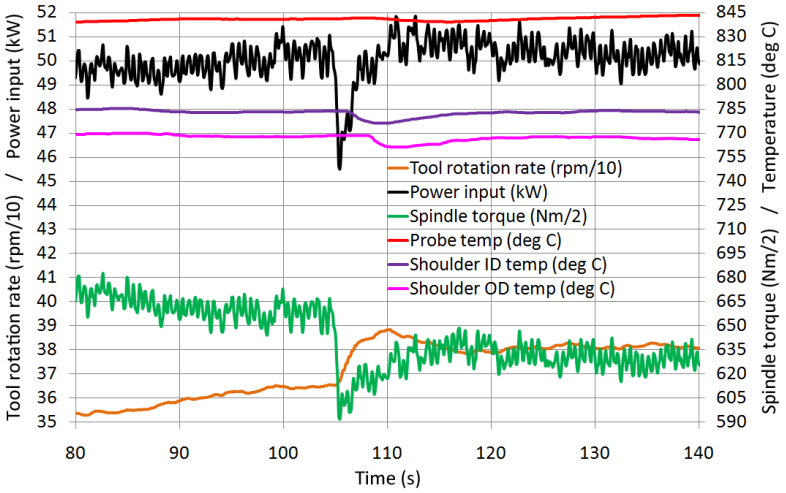

Figure 18. Torque disturbance rejection during the downward sequence (3).

The temperature reference was set to 845°C in these tests. The temperature disturbances acting on the process are also visible in the figure. At 0 seconds, all welds have approximately the same power input and temperature. Even though this temperature is close to the set-point, the power input has to be increased considerably for the process to achieve steady-state during the joint line sequence (4). The temperature curves also tend to drop a bit after the downward sequence (3) has started. These are typical results of the temperature disturbances acting on the process. The fact that a more aggressive controller is used during the downward sequence (3) will, however, reduce the temperature drop a bit.

The large amount of torque disturbances occurring during the start-up tests are good for showing how well the inner controller works. Figure 18 displays the process data during the same disturbance that was shown in Fig. 7. Although the power input instantaneously drops from 51 to 45.5 kW, the controller increases the tool rotation rate by 20 rpm in the next 2 seconds such that the power input is back at 50 kW. While the shoulder ID and OD signals have small bumps due to the disturbance, no significant probe temperature deviations can be registered. This supports the claim that the current inner PI controller is fast enough for the FSW process.

Circumferential welds Two sets of four circumferential welds were run using the cascade controller strategy. The temperature reference was set to 840 and 845°C respectively. Figure 19 (a) shows how the temperatures varied during the joint line sequences (4), which can be compared to

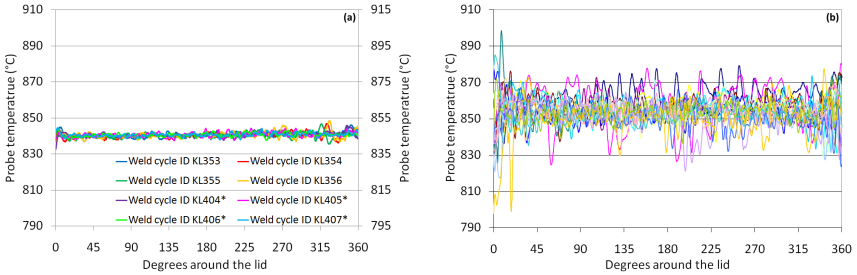

Figure 19. Probe temperatures at the joint line using the cascade controller (a) compared to using manual control (b). A * refers to values on the right y-axis.

20 circumferential welds (with a temperature reference of 850°C) that were made, in air (not argon), by a skilled welding operator in 2004 (see Fig. 19 (b)). It can be seen that there have been vast improvements since the argon gas and cascade controller were introduced.

Figure 20 shows the parking sequences (5) for two of the first four welds. In the first weld (ID 353), the temperature reference was set to a constant value during this sequence. The results of this were that all three temperature measurements showed rather large increases, due to the two temperature disturbances that are active during this sequence (see Section 2.6). To avoid the risk of ending up with a probe fracture, the temperature reference was manipulated to reduce this effect. The same figure shows that this change, although a bit primitive, is enough to keep the probe temperature at a safe level. The shoulder OD signal went up a bit even though the probe temperature decreased. This effect was, however, much smaller than during the first weld.

6. Conclusions

The results presented in this article have shown that a combination of a cascaded control structure and individual PI controllers is successful in controlling the FSW process to seal copper canisters. While this is quite a specific process, the authors still believe that the methods and ideas behind the results can be put into a wider context such that other FSW applications can benefit from them too. In particular, applications that have either varying thermal boundary conditions or narrow process windows. Apart from this, any process that can be divided into two separate processes with distinctly different time constants will likely benefit a great deal from a cascade controller.

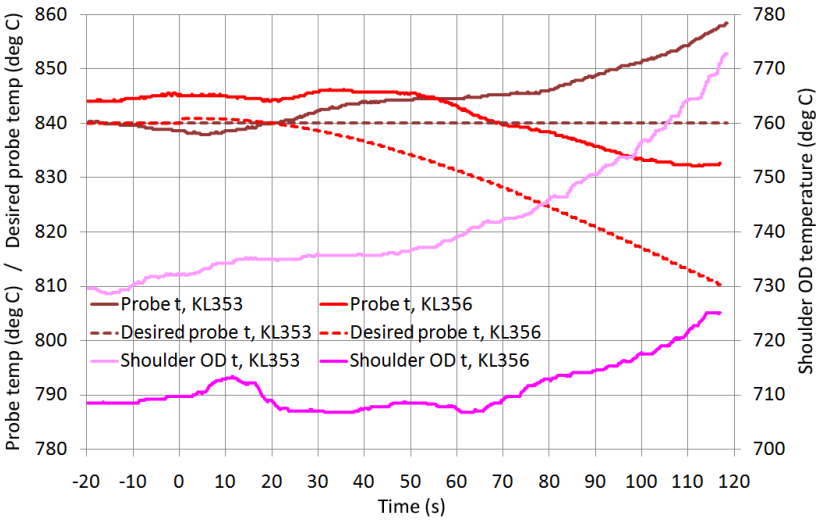

Figure 20. Two parking sequences (5). t is here an abbreviation for temperature.

Apart from the cascade structure, good process knowledge and a control strategy adapted to the changing weld sequences have probably been the biggest contributors behind the satisfactory control of this application. The choice of the individual controllers has, however, not been as crucial. PI controllers in both loops have shown to be more than enough for handling the three types of disturbances as well as the reference tracking. This also fits well with the ambition of developing a simple procedure for controller design, such that people without a major education in control theory can carry out the steps necessary.

Acknowledgments

The authors would like to thank the Swedish Nuclear Fuel and Waste Management Company (SKB) for their financial support to this project. Thanks also to the ESAB personnel for their many visits to update the research possibilities of the welding equipment including controller implementation.

References

Åström, K. J. (2000). "Limitations on Control System Performance". *European Journal of Control* **6**:1, pp. 2–20.

Åström, K. J. and T. Hägglund (2005). *Advanced PID Control*. ISA – The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC.

Cederqvist, L. and T. Öberg (2007). "Reliability study of friction stir welded copper canisters containing Sweden's nuclear waste". *Reliability Engineering & System Safety* **93**:10, pp. 1491–1499.

Cederqvist, L., C. D. Sorensen, A. P. Reynolds, and T. Öberg (2008). "Improved process stability during friction stir welding of 5 cm thick copper canisters through shoulder geometry and parameter studies". *Science and Technology in Welding and Joining* **46**:2, pp. 178–184.

Cederqvist, L., R. Johansson, A. Robertsson, and G. Bolmsjö (2009). "Faster Temperature Response and Repeatable Power Input to aid Automatic Control of Friction Stir Welded Copper Canisters". In: *Friction Stir Welding and Processing V*. San Francisco, USA.

Cederqvist, L., A. P. Reynolds, C. D. Sorensen, and O. Garpinger (2010a). "Reliable sealing of copper canisters using new tool geometry and regulator controlling tool temperature and power input". In: *8th International Friction Stir Welding Symposium*. Timmendorfer Strand, Germany.

Cederqvist, L. (2011). *Friction Stir Welding of Copper Canisters Using Power and Temperature Control*. PhD thesis. Department of Machine Design, Lund University.

Cederqvist, L., O. Garpinger, T. Hägglund, and A. Robertsson (2010b). "Cascaded Control of Power Input and Welding Temperature During Sealing of Spent Nuclear Fuel Canisters". In: *Proc. ASME Dynamic Systems and Control Conference*. Cambridge, Massachusetts.

Fehrenbacher, A., F. E. Pfefferkorn, M. R. Zinn, N. J. Ferrier, and N. A. Duffie (2008). "Closed-loop Control of Temperature in Friction Stir Welding". In: *7th International Friction Stir Welding Symposium*. Awaji Island, Japan.

Fehrenbacher, A., N. A. Duffie, N. J. Ferrier, M. R. Zinn, and F. E. Pfefferkorn (2010). "Temperature measurement and closed-loop control in friction stir welding". In: *8th International Friction Stir Welding Symposium*. Timmendorfer Strand, Germany.

Forsman, K. (2005). *Reglerteknik för processindustrin*. Studentlitteratur, Lund, Sweden.

Garpinger, O. and T. Hägglund (2008). "A Software Tool for Robust PID Design". In: *17th IFAC World Congress*. Seoul, South Korea.

Garpinger, O. (2009). *Design of Robust PID Controllers with Constrained Control Signal Activity*. Licentiate Thesis ISRN LUTFD2/TFRT--3245--SE. Department of Automatic Control, Lund University, Sweden.

Hägglund, T. (2008). *Praktisk processreglering*. Studentlitteratur, Malmö, Sweden.

Hägglund, T. and K. J. Åström (2004). "Revisiting the Ziegler-Nichols step response method for PID control". *Journal of Process Control* **14**:6, pp. 635–650.

Larsson, P.-O. and T. Hägglund (2011). "Control Signal Constraints and Filter Order Selection for PI and PID Controllers". In: *2011 American Control Conference*. San Francisco, California, USA.

Lohwasser, D. and Z. Chen, (Eds.) (2009). *Friction Stir Welding: From basics to applications*. Woodhead Publishing Ltd, 80 High Street, Cambridge CB22 3HJ, UK.

Longhurst, W. R., A. M. Strauss, G. E. Cook, and P. A. Fleming (2010). "Torque control of friction stir welding for manufacturing and automation". *International Journal of Advanced Manufacturing Technology* **51**:9, pp. 905–913.

Mayfield, D. W. and C. D. Sorensen (2010). "An improved temperature control algorithm for friction stir processing". In: *8th International Friction Stir Welding Symposium*. Timmendorfer Strand, Germany.

Mishra, R. S. and M. W. Mahoney, (Eds.) (2007). *Friction Stir Welding and Processing*. ASM International, Materials Park, Ohio 44073-0002.

Nandan, R., T. DebRoy, and H. K. D. H. Bhadeshia (2008). "Recent advances in friction-stir welding – Process, weldment structure and properties". *Progress in Materials Science* **53**:6, pp. 980–1023.

Nordfeldt, P. (2005). *PID Control of TITO Systems*. Licentiate Thesis ISRN LUTFD2/TFRT--3238--SE. Department of Automatic Control, Lund University, Sweden.

Panagopoulos, H., K. J. Åström, and T. Hägglund (2002). "Design of PID controllers based on constrained optimisation". *IEE Proceedings - Control Theory & Applications* **149**:1, pp. 32–40.

Ronneteg, U., L. Cederqvist, H. Rydén, T. Öberg, and C. Müller (2006). *Reliability in sealing of canister for spent nuclear fuel*. Tech. rep. Swedish Nuclear Fuel and Waste Management Company.

Soron, M. (2007). *Robot System for Flexible 3D Friction Stir Welding*. PhD thesis. Department of Technology, Örebro University.

Thegerström, C. (2004). “Down to Earth and Below: Sweden’s Plans for Nuclear Waste”. *IAEA Bulletin* **46**:1, pp. 36–38.

Thomas, W. M., E. D. Nicholas, J. C. Needham, M. G. Murch, P. Temple-Smith, and C. J. Dawes (1991). *Friction stir butt welding*. International Patent Application No. PCT/GB92/02203.

Upadhyay, P. and A. P. Reynolds (2010). “Effects of thermal boundary conditions in friction stir welded AA7050-T7 sheets”. *Materials Science and Engineering: A* **527**:6, pp. 1537–1543.

Wallén, A. (2000). *Tools for Autonomous Process Control*. PhD thesis ISRN LUTFD2/TFRT--1058--SE. Department of Automatic Control, Lund Institute of Technology, Sweden.

Wang, X. J., Z. K. Zhang, R. J. Guo, X. H. Han, and A. Rong (2004). “Application of MCGS for process control and realtime detection in the process of Friction Stir Welding”. In: *8th International Conference on Control, Automation, Robotics and Vision*. Kumming, China.

Supplement to Paper V

This supplement has been added to the thesis since Paper V uses the old controller design procedure from [Garpinger, 2009]. Here we will show that the newly proposed PID design method from Paper IV gives even better control of the FSW process.

Notice that all references made in this supplement are collected within the reference list of Paper V.

S.1 Inner-loop control design

The inner-loop noise sensitivity and performance trade-off shown in Paper V, Fig. 14, was derived for PI controllers and with respect to various levels of robustness. This can be compared to the procedure proposed in Paper IV, which suggests that a given robustness is used to derive a set of controllers with respect to different filter time constants and integral gains.

Since the sampling time of the FSW process, $h = 0.1$ seconds, is rather long for the power input control, a discrete-time version of SWORD was used to derive the inner-loop controllers in Paper V. Since the article was written, this tool has been updated to only use backward Euler consistently for the discrete-time controller approximations. This gives a slight improvement in both closed-loop robustness and performance. Figure S.1 shows the performance and noise sensitivity trade-off curves using both the procedure from Paper IV (solid line) and Paper V (dashed line). PID controllers are indicated by blue color, PI controllers by green and I controllers by red. The PI and PID controllers in the solid curve were derived using $M_s = M_t = 1.4$. Real noise data from the FSW process was used in closed-loop Simulink® simulations, in order to derive the noise sensitivities, here given by the by the standard deviation of the control signal. The PI controller chosen in Paper V, Eq. (8), is indicated by a black circle in the figure. Notice that it lies slightly above the dashed trade-off curve

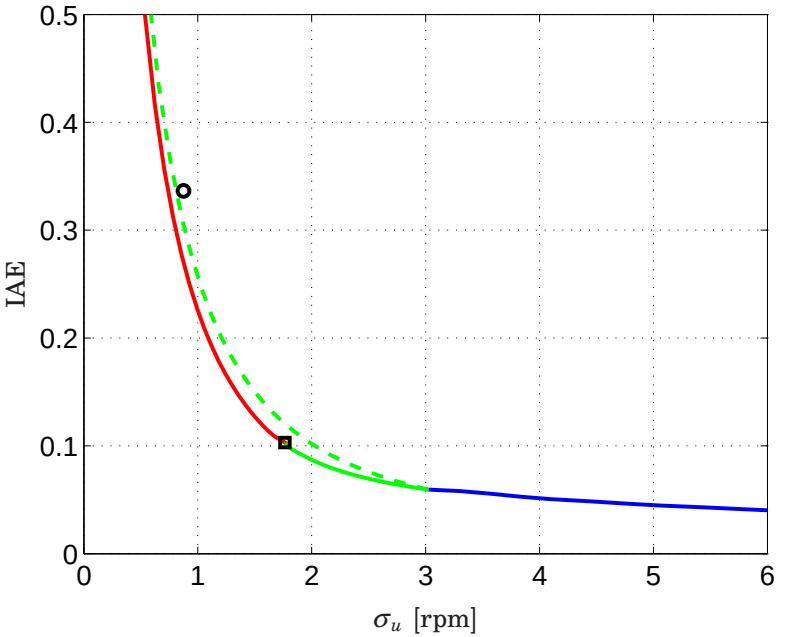

Figure S.1 Performance and noise sensitivity trade-off curves for the inner loop, with respect to control signal standard deviation. The solid line shows the proposed set of PID (blue), PI (green) and I (red) controllers, while the dashed green line shows the previously proposed set of PI controllers. The formerly used PI controller is marked with a black circle and the newly chosen controller is marked with a black square.

since it uses the old discrete-time version of SWORD. A new and more aggressive I controller,

$$C_1(s) = \frac{10}{s}, \tag{S.1}$$

marked with a black square in the figure, was chosen for the inner loop. This controller has considerably better performance than the old one, but the noise sensitivity is also greater. However, we get a different picture if we instead plot the trade-off curves with respect to a noise sensitivity given by the total variation (TV, Eq. (3.11) in the introductory chapters) per second, see Fig. S.2. Here, we see that there is a considerable difference in the two trade-off curves, and the unfiltered PI controllers give a much higher noise sensitivity than the filtered controllers do. In fact, the old controller leads to a higher TV than the new one, even though it had a much lower standard deviation. This result agrees well with those

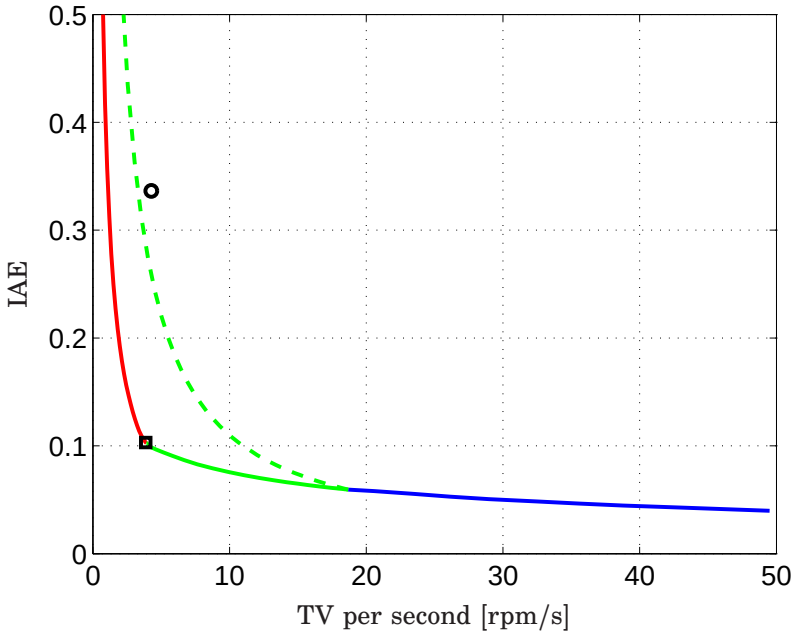

Figure S.2 Performance and noise sensitivity trade-off curves for the inner loop, with respect to the total variation per second of the control signal. The solid line shows the proposed set of PID (blue), PI (green) and I (red) controllers, while the dashed green line shows the previously proposed set of PI controllers. The formerly used PI controller is marked with a black circle and the newly chosen controller is marked with a black square.

from [Larsson and Hägglund, 2011], which suggested to use a measurement noise filter of the least order that gives a high frequency roll-off. From Fig. S.2 it now makes even more sense to pick the best performing I controller (S.1). Performance quickly deteriorates for lower integral gains at the same time as the TV does not decrease much. The relative performance gain in using PI control is also low in comparison to what it costs in terms of noise sensitivity.

S.2 Outer-loop control design

The outer temperature control loop is much slower than the inner power input loop. Continuous-time and discrete-time PID control design will

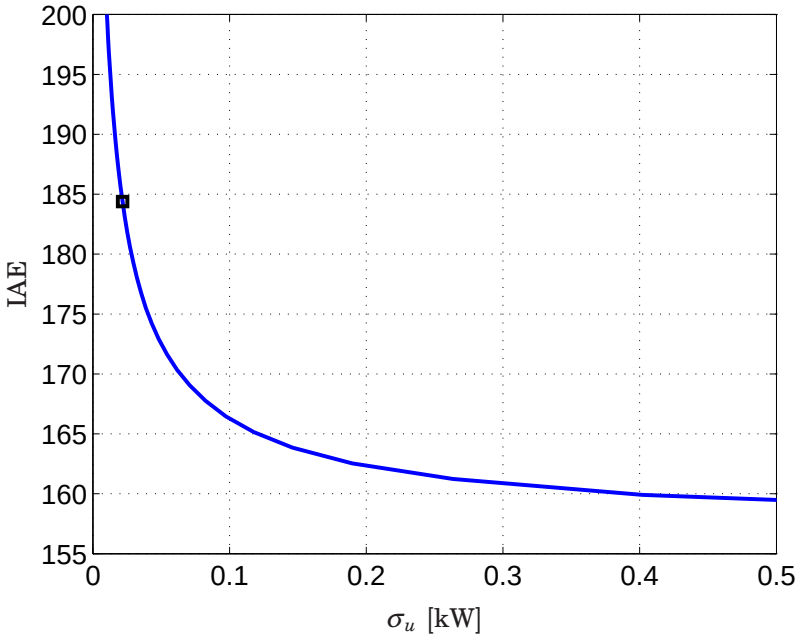

Figure S.3 Performance and noise sensitivity trade-off curve for the outer temperature loop, with respect to control signal standard deviation due to noise. The solid blue line shows the proposed set of PID controllers. The selected controller is marked with a black square.

thus lead to more or less the same controller, and we have chosen to use the continuous-time version of SWORD.

The temperature measurements contain relatively little noise, and it should thus be fine to use an outer PID controller, rather than a PI controller as suggested in Paper V. A ten second sequence of temperature noise data was used to derive a performance and noise sensitivity trade-off curve for PID controllers with $M_s = M_t = 1.4$, see Fig. S.3. Control signal noise responses with several of these controllers were plotted to get visual feedback of their behavior. Two of them, with filter time constants $T_f = 0.2$ and $T_f = 1.0$, are shown in Fig. S.4. The blue curves show the outer-loop control signal, and the red curves show the resulting inner-loop control signal. Given this information, we choose the filter with $T_f = 1.0$, which results in slow and relatively low magnitude variations of the tool rotation rate, even though the performance loss is significant

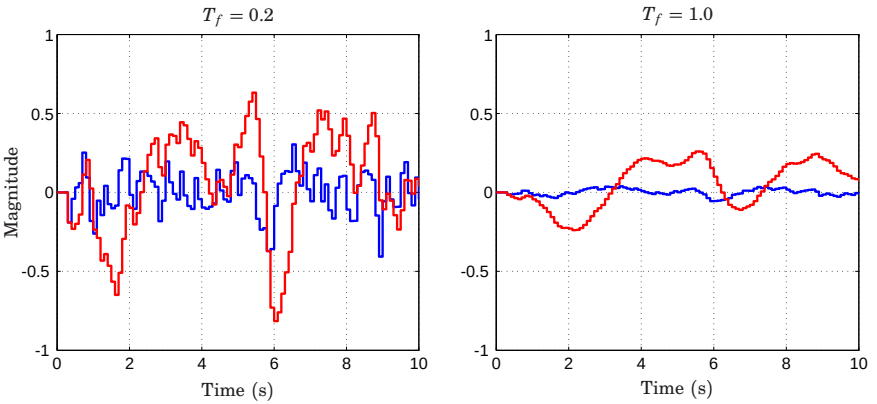

Figure S.4 Simulated control signal noise responses using real temperature noise data from the FSW process. Red signals show the inner-loop control signal (rpm) and the blue signals show the outer-loop control signal (kW). The responses to the left correspond to a low-pass filter with $T_f = 0.2$, while the right hand plot shows the same with $T_f = 1.0$.

in comparison to controllers with $\sigma_u \approx 0.1$. This controller,

$$C_2(s)C_{f_2}(s) = 0.071\left(1 + \frac{1}{10.55s} + 5.13s\right) \cdot \frac{1}{0.5s^2 + s + 1}, \tag{S.2}$$

gives IAE = 184.4, i.e. 16% worse performance than the best possible PID controller,

$$C_2(s) = 0.081\left(1 + \frac{1}{10.26s} + 4.8s\right).$$

The FSW process is especially sensitive to high power input values during start-up, see Section 5.3 in Paper V. It is thus important that the final cascade controller does not give significant temperature overshoots during the first weld sequences. Closed-loop reference step responses were simulated on the system in Fig. 9, Paper V, using the PID controller from Eq. (S.2). Even if both the proportional and derivative set-point weights are set to zero (i.e. $b = c = 0$, see Eqs. (1.8) and (1.10) in the introductory chapters), the simulations still show an 11% temperature overshoot. Since there are no other options for set-point weighting on the FSW process, we choose to instead manipulate the overshoot using the M_t-value. Choosing $M_t = 1.002$ and $T_f = 1.0$ in SWORD gives the PID controller and low-pass filter,

$$C_2(s)C_{f_2}(s) = 0.071\left(1 + \frac{1}{12.63s} + 4.63s\right) \cdot \frac{1}{0.5s^2 + s + 1}, \tag{S.3}$$

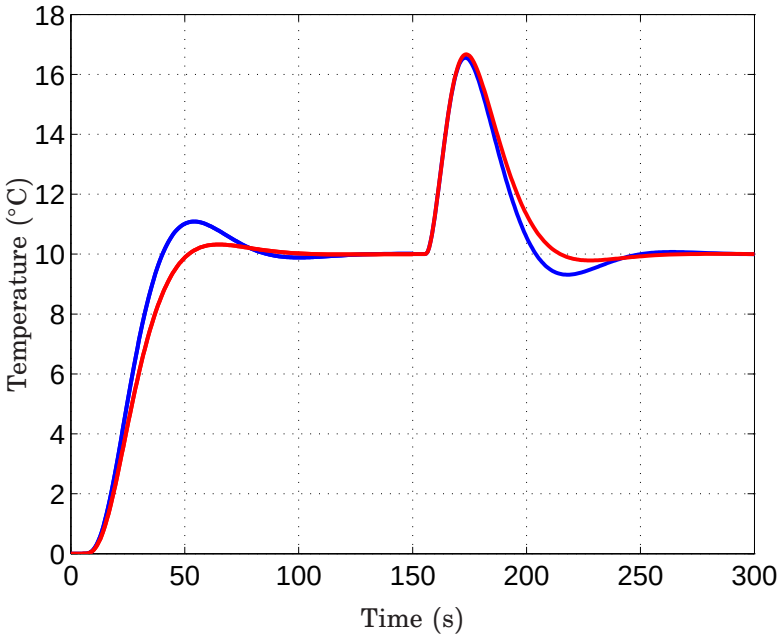

Figure S.5 Step set-point and load disturbance responses using outer PID controllers designed with $M_s = M_t = 1.4$ (blue) and $M_s = 1.4$, $M_t = 1.002$ (red).

which results in the red reference and load disturbance responses shown in Fig. S.5. These can be compared to the responses in blue, given with the previous controller (S.2). The overshoot has been decreased 3.2% at the same time as the load disturbance IAE is just 2.8% worse. The PID controller (S.3) was finally chosen to the control the FSW process together with the I controller in (S.1).

For comparison, the old PI controllers from Paper V have 79% ($M_s = M_t = 1.4$) and 17% ($M_s = M_t = 1.8$) worse IAE than (S.3). The best possible, unfiltered, PI and I controllers with $M_s = M_t = 1.4$ give 58% and 189% worse performance respectively.

S.3 Results on the FSW process

So far, all plots have shown simulated data. But the improved cascade controller has also been tested on full welds. Probe temperatures of two full welds during the joint line sequences are shown in Fig. S.6. The blue

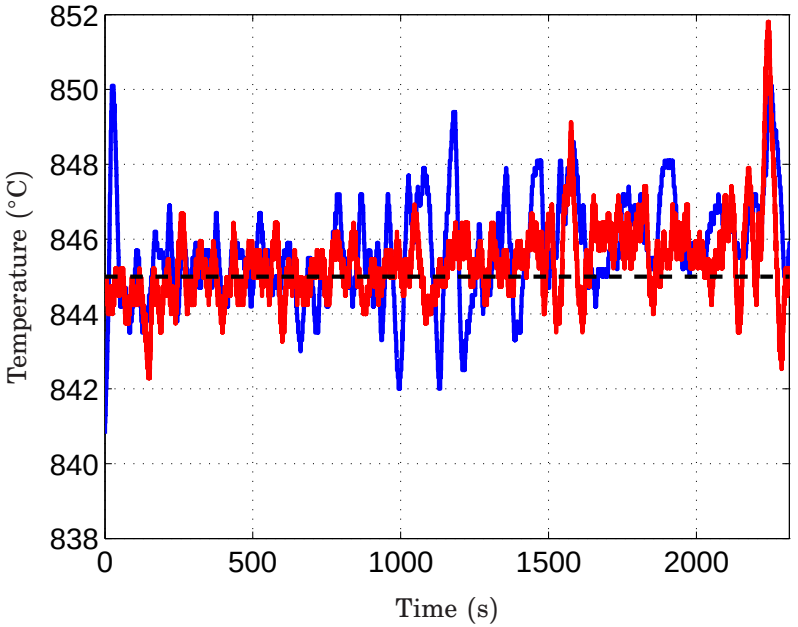

Figure S.6 Probe temperatures during 60 seconds of the joint line sequences of two full welds on the FSW process. The blue line shows the weld using the old cascade controller from Paper V, while the red line shows the weld using the proposed cascade controller.

curve indicates a weld made with the two old PI controllers, and the red curve shows the result with the new cascade controller. The temperature reference was set to 845°C in both cases. The old cascade controller results in an IAE that is 58% higher than that of the new controller, i.e. fairly close to the previously predicted value of 79%. An example of the control signals from the two welds is plotted in Fig. S.7, showing 60 seconds of weld data. As expected, the control signal of the new controller is slightly smoother than the old one. This agrees with the prediction from Section S.1.

S.4 Conclusions

It has been shown that the overall performance of the FSW temperature control can be significantly improved by using the new controller design method. We could probably have improved the cascade control even further

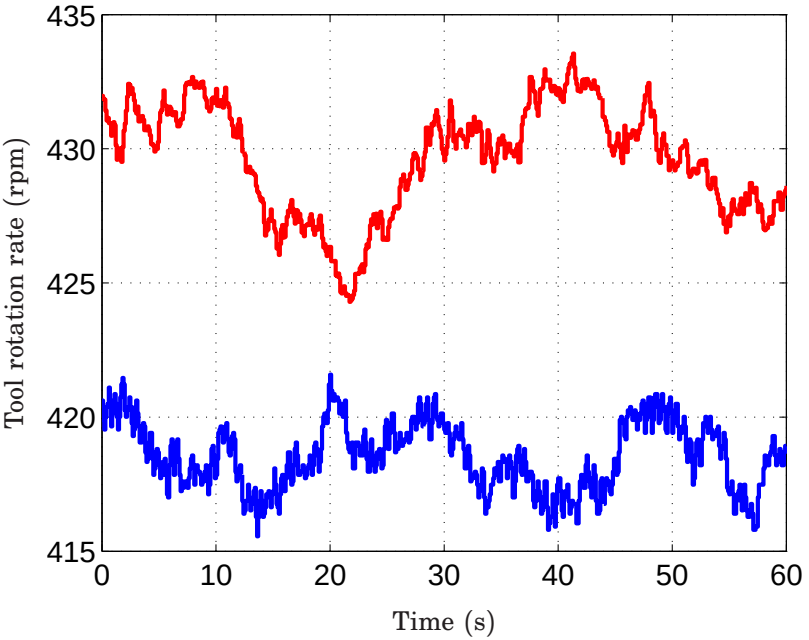

Figure S.7 Tool rotation rate during part of the joint line sequences from two full welds. The blue line corresponds to the old cascade controller from Paper V and the red line to the proposed controller.

by using more realistic disturbance profiles in the SWORD software. But, this supplement was rather made to show how well the general optimal PID design method presented in Paper IV would work on the FSW process.