Journal of Process Control 70 (2018) 36–46

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

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

Chriss Grimholt, Sigurd Skogestad *

Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

ARTICLE INFO

Article history:
Received 10 September 2017
Received in revised form 16 May 2018
Accepted 27 June 2018

Keywords:
Process control
PID control
Robustness
Tuning rules

ABSTRACT

The first-order plus delay process model with parameters k (gain), τ (time constant) and θ (delay) is the most used representation of process dynamics. This paper has three objectives. First, we derive optimal PI- and PID-settings for this process. Optimality is here defined as the minimum Integrated Absolute Error (IAE) to disturbances for a given robustness level. The robustness level, which is here defined as the sensitivity peak (M_s), may be regarded as a tuning parameter. Second, we compare the optimal IAE-performance with the simple SIMC-rules, where the SIMC tuning parameter τ_c is adjusted to get a given robustness. The "original" SIMC-rules give a PI-controller for a first-order with delay process, and we find that this SIMC PI-controller is close to the optimal PI-controller for most values of the process parameters (k, τ, θ). The only exception is for delay-dominant processes where the SIMC-rule gives a pure integrating controller. The third objective of this paper is to propose and study a very simple modification to the original SIMC-rule, which is to add a derivative time $\tau_d = \theta/3$ (for the serial PID-form). This gives performance close to the IAE-optimal PID also for delay-dominant processes. We call this the "improved" SIMC-rule, but we put "improved" in quotes, because this controller requires more input usage, so in practice the original SIMC-rule, which gives a PI-controller, may be preferred.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

The PID controller is by far the most common controller in industrial practice. However, although it has only three parameters, it is not easy to tune unless one uses a systematic approach. The first PID tuning rules were introduced by Ziegler and Nichols [25]. Although some other empirical rules were later suggested, the Ziegler-Nichols (ZN) rules remained for about 50 years as the best and most commonly used rules. However, there are at least three problems with the ZN-rules:

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

For many years there was almost no academic interest in revisiting the PID controller to obtain better tuning rules. Dahlin [7] considered discrete-time controllers and introduced the idea of specifying the desired closed-loop response and from this backing out the controller parameters. Typically, a first-order response is specified with closed-loop time constant τ_c (called λ by Dahlin). Importantly, τ_c (or λ) is a single tuning parameter which the engineer can use to specify how aggressive the controller should be. For first or second-order plus delay processes, the resulting controller can be approximated by a PID controller. This idea is also the basis of the IMC PID-controller of [22] which results in similar PID tuning rules. The IMC PI-tuning rules, also known as lambda tuning, became widely used in the pulp and paper industry around 1990 [5].

However, the Dahlin and IMC rules set the controller integral time equal to the dominant process time constant ($\tau_i = \tau$) and this means that integral action is effectively turned off for "slow" or "integrating" processes with a large value of τ. This may be acceptable for setpoint tracking, but not for load disturbances, that is, for disturbances entering at the plant input. This led [24] to suggest the SIMC rule where τ_i is reduced for processes with large time constants. However, to avoid slow oscillations it should not be reduced too much, and this led to the SIMC-rule $\tau_i = \min\{\tau, 4(\tau_c + \theta)\}$, where θ is the effective time delay of the process.

* Corresponding author.
E-mail address: skoge@ntnu.no (S. Skogestad).

https://doi.org/10.1016/j.jprocont.2018.06.011
0959-1524/© 2018 Elsevier Ltd. All rights reserved.

Since about 2000, partly inspired by the work of Åström (e.g. [2,4]), there has been a surge in academic papers on PID control as can be seen by the Handbook on PID rules by [21] which lists hundreds of tuning rules.

In particular, the very simple SIMC PID tuning rules [24] have found widespread industrial acceptance. However, there has also been suggestions to improve the SIMC rules [16,20]. One question then naturally arises: Is there any point in searching for better PID rules for first-order plus delay processes, or are the SIMC rules good enough? To answer this question, we want in this paper to answer the following three more detailed questions: 1. What are the optimal PI and PID settings for a first-order with delay process? 2. How close are the simple SIMC rules to these optimal settings? 3. Can the SIMC rules be improved in a simple manner?

We consider the stable first-order plus time delay process

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

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

$$K_{PID}(s) = \frac{k_c(\tau_i s + 1)(\tau_d s + 1)}{\tau_i s}, \tag{2}$$

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

$$K_{PID}^{parallel}(s) = k_c'\left(1 + \frac{1}{\tau_i' s} + \tau_d' s\right), \tag{3}$$

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

$$k_c' = k_c f, \quad \tau_i' = \tau_i f, \quad \text{and} \quad \tau_d' = \tau_d / f. \tag{4}$$

For PI-control, $f = 0$ the two forms are identical. In addition, a filter F is added, at least when there is derivative action, so the overall controller is

$$K(s) = K_{PID}(s)F(s). \tag{5}$$

Normally, we use is a first-order filter,

$$F = \frac{1}{\tau_f s + 1}. \tag{6}$$

Note that τ_f is not considered a tuning parameter in this paper, but rather set at a fixed small value, depending on the case. The filter is generally needed when we have derivative action, and we may write $\tau_f = \tau_d/\alpha$ where α often is in the range from 5 to 10. For other notation, see Fig. 1.

The main trade-off in controller design is between the benefits of high controller gains (performance) and the disadvantages of high controller gain (robustness and input usage) e.g. [6,19]. In this paper, we focus on the trade-off between IAE-performance and M_s-robustness. More precisely, we use the integrated absolute error (IAE) for combined input and output disturbances as the performance measure and obtain optimal PI and PID settings for various robustness levels, where robustness is measured in terms of the peak sensitivity (M_s). The resulting Pareto-optimal trade-off between performance and robustness is subsequently used to evaluate the SIMC PI and PID rules.

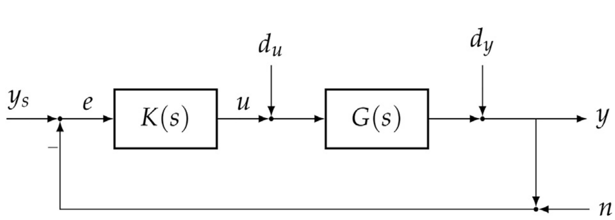

Fig. 1. Block diagram of one degree-of-freedom feedback control system. We may treat a setpoint change (y_s) as a special case of an output disturbance (d_y).

The paper is structured as follows. In Section 2 we define the measures used to quantify the performance/robustness trade-off. Based on this, optimal PI and PID controllers are presented in Section 3. In Section 4 we present the SIMC rules and propose "improved" SIMC rules, referred to as *i*SIMC and *i*SIMC-PI in this paper. In Section 5, the SIMC and improved SIMC rules evaluated. In Section 6, we discuss input usage and some other issues.

Preliminary versions of some of the results were presented in [11,13].

2. Quantifying the optimal controller

The first authors to use the terms "optimal settings" for PID-control where Ziegler and Nichols in their classical paper [25]. Generally, it is difficult to define "optimality" of a controller, as there are many important aspects to take into consideration, including set-point response, disturbance rejection, robustness, input usage, and noise sensitivity. Often a control loop is evaluated solely on the basis of its response to a setpoint change, but in process control, disturbance rejection is usually the major concern. Another important aspect is robustness, which often is completely omitted. [3] emphasize the need of including all the behaviours of the control loop.

2.1. Performance

In this paper, we quantify performance in terms of the integrated absolute error (IAE),

$$\text{IAE} = \int_0^\infty |y(t) - y_s(t)| \mathrm{d}t. \tag{7}$$

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

$$J(p) = 0.5\left(\frac{\text{IAE}_{dy}(p)}{\text{IAE}_{dy}^{\circ}} + \frac{\text{IAE}_{du}(p)}{\text{IAE}_{du}^{\circ}}\right) \tag{8}$$

where IAE_{dy}° and IAE_{du}° are weighting factors, and p denotes the controller parameters. Note that we do not consider setpoint responses, but instead output disturbances. For the system in Fig. 1, the closed-loop responses in the error $e = y_s - y$ to an output disturbance d_y and to a setpoint change y_s are identical, except for the sign. The difference is that since the setpoint is known we could further enhance the setpoint performance using a two-degrees-of freedom controller (which is not considered in this paper), whereas the unmeasured output disturbance can only be handled by the feedback controller K (which is the focus of this paper). Of course, we may consider other disturbance dynamics, but step disturbances at the plant input and output are believed to be representative for most cases.

The two weighting factors IAE° for input and output disturbances, respectively, are selected as the optimal IAE values when using PI control (as recommended by [6]). To ensure robust reference PI controllers, they are required to have $M_s = 1.59$,[1] and the resulting weighting factors are given for four processes in Table 1.

[1] For those that are curious about the origin of this specific value $M_s = 1.59$, it is the resulting M_s-value for a SIMC-tuned PI-controller with $\tau_c = \theta$ on a first-order plus delay process with $\tau \leq 8\theta$.

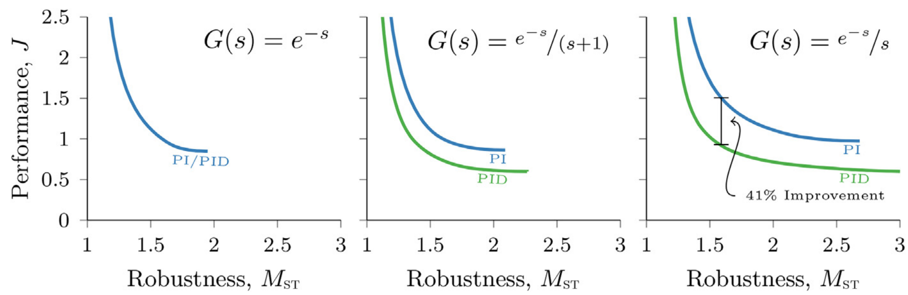

Fig. 2. Pareto-optimal IAE-performance (J) for PI and PID control.

Table 1
Reference PI-controllers and resulting weighting factors for four processes.

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

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

Note that two different reference PI controllers are used to obtain the weighting factors.

2.2. Robustness

Robustness may be defined in many ways, for example, using the classical gain and phase margins, which are related to robustness with respect to the model parameters k and θ, respectively. However, as a single robustness measure, we in this paper quantify robustness in terms of M_{ST}, defined as the largest value of M_s and M_T [9],

$$M_{ST} = \max\{M_s, M_T\}. \tag{9}$$

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

$$\begin{aligned} M_s &= \max_w |S(jw)| = \|S(jw)\|_\infty, \\ M_T &= \max_w |T(jw)| = \|T(jw)\|_\infty, \end{aligned}$$

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

$$S(s) = \frac{1}{(1+G(s)K(s))} \quad \text{and} \quad T(s) = 1 - S(s). \tag{10}$$

For most stable processes, $M_s \geq M_T$. In the frequency domain (Nyquist plot), M_s is the inverse of the closest distance between the critical point -1 and the loop transfer function $G(s)K(s)$. For robustness, small M_s and M_T values are desired, and generally M_s should not exceed 2. For a given M_s we are guaranteed the following GM and PM, [22].

$$\mathrm{GM} \geq \frac{M_s}{M_s - 1} \quad \text{and} \quad \mathrm{PM} \geq 2\arcsin\left(\frac{1}{2M_s}\right) \geq \frac{1}{M_s}. \tag{11}$$

For example, $M_s = 1.6$ guarantees $\mathrm{GM} \geq 2.67$ and $\mathrm{PM} \geq 36.4^\circ = 0.64$ rad.

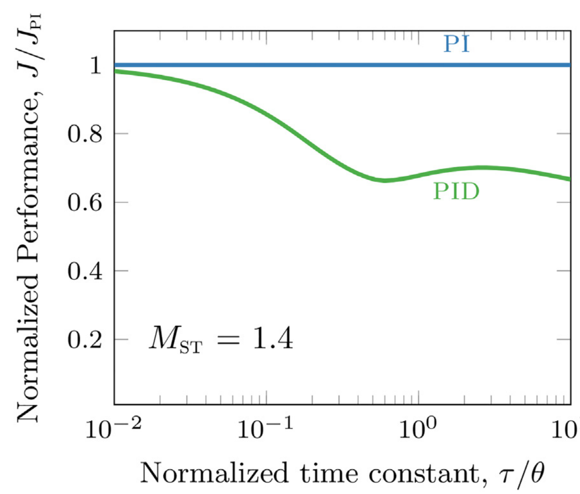

Fig. 3. PID versus PI control: Normalized IAE-optimal performance for $M_{ST} = 1.4$ as a function of the normalized time constant τ/θ.

2.3. Optimal controller

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

$$\min_p \quad J(p) = 0.5\left(\frac{\mathrm{IAE}_{dy}(p)}{\mathrm{IAE}^{\circ}_{dy}} + \frac{\mathrm{IAE}_{du}(p)}{\mathrm{IAE}^{\circ}_{du}}\right) \tag{12}$$

$$\text{subject to:} \quad M_s(p) \leq M^{ub} \tag{13}$$

$$M_T(p) \leq M^{ub} \tag{14}$$

where in this paper the parameter vector p is for a PI or PID controller. For more details on how to solve the optimization problem, see [14]. The problem is solved repeatedly for different values of M^{ub}. One of the constraints in (13) or (14) will be active if there is a trade-off between robustness and performance. This is the case for values of M^{ub} less than about 2 to 3. Usually the M_s-bound is active, except for integrating processes with a small M^{ub} (less than about 1.3), where the M_T-bound is active (see Fig. 4, later).

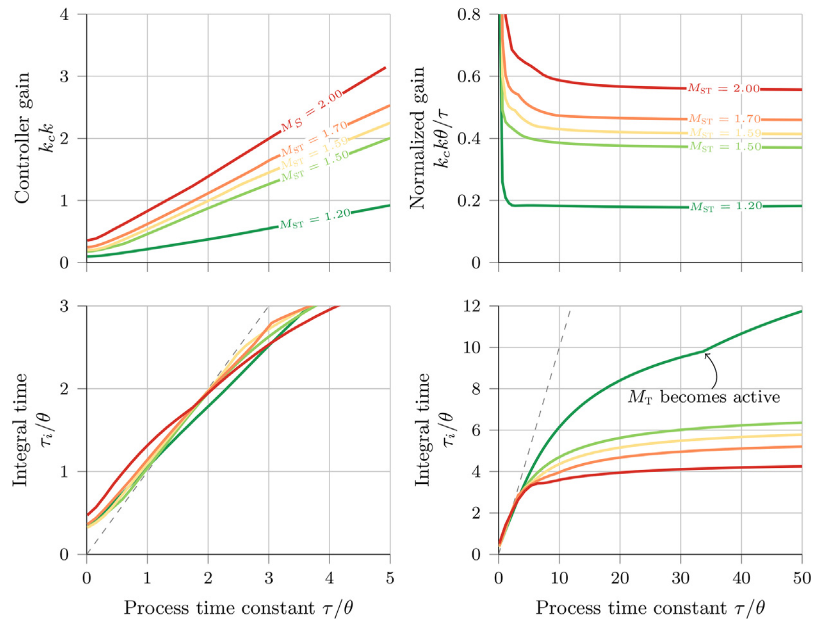

Fig. 4. PI control: IAE-optimal settings as a function of τ/θ for five values of M_{ST}.

Table 2
PI control: comparison of optimal, SIMC, and *i*SIMC-PI controllers with $M_{ST} = 1.59$.

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

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

In retrospect, looking at the results of this paper, we would have obtained similar optimal PI- and PID-settings for the process (1) if we only considered input disturbances for performance and only used M_s for robustness.

3. Optimal PI and PID control

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

In this section, we present IAE-optimal (J) settings for PI and PID controllers as a function of the robustness level (M_{ST}). However, before presenting the optimal settings, we show in Fig. 2 the Pareto-optimal IAE-performance (J) as a function of robustness (M_{ST}) for the optimal PI and PID controllers for three processes. Note that the curves in Fig. 2 stop when M_{ST} is between 2 and 3. This is because performance (J) actually gets worse and the curve for J bends upwards [11] when M_{ST} increases beyond this value. Thus, there is no trade-off and the region with M_{ST} larger than about 2 should be avoided.

We see from Fig. 2 that for a pure time delay process there is no advantage in adding derivative action; and it is optimal to use simple PI control. As the time constant τ increases, the benefit of using derivative action also increases. For an integrating process, derivative action improves IAE-performance by about 40%, compared to optimal PI control. This is emphasised again in Fig. 3, where performance is shown as a function of the normalized time constant for robust controllers with $M_{ST} = 1.4$.

3.2. Optimal PI control

The IAE-optimal PI settings are shown graphically in Fig. 4 as a function of τ/θ for different robustness levels (M_{ST}) and are also given for $M_s = 1.59$ for four processes in Table 2. For PI control we observe three main regions (Fig. 4) in terms of the optimal integral time τ_i:

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

These regions match well the classification of first-order plus time delay processes in [10].

In contrast with the IMC rules [22] and the SIMC rules [24], the optimal PI controller does not converge to a pure integral controller ($K_I(s) = k_i/s$, corresponding to $\tau_i \to 0$) as as we approach a pure time delay process ($\tau/\theta \to 0$) (Fig. 4). Rather, for a pure time delay processes, the optimal integral time is approximately $\theta/3$, which we will use in the proposed *i*SIMC – PID and *i*SIMC – PI rules (see below). The optimal integral time of about $\theta/3$ is almost independent of the robustness level (M_{ST}-values). For balanced processes, the integral time is similar to the time constant ($\tau_i \approx \tau$, see dashed line), and also almost independent of the robustness level. This value agrees with the IMC and SIMC rules.

For lag-dominant processes (with $\tau > 4\theta$), the integral time for $M_{ST} = 1.59$ approaches $\tau_i = 6.22\theta$ for $\tau/\theta = \infty$ (integrating process) (Fig. 4, lower right). This is somewhat smaller than the value $\tau_i = 8\theta$ obtained from the SIMC rule. Also the normalized controller gain, $k_c k\theta/\tau$ approaches a constant value as τ goes to infinity (Fig. 4, upper right). For example, for $M_{ST} = 1.59$, the optimal value is

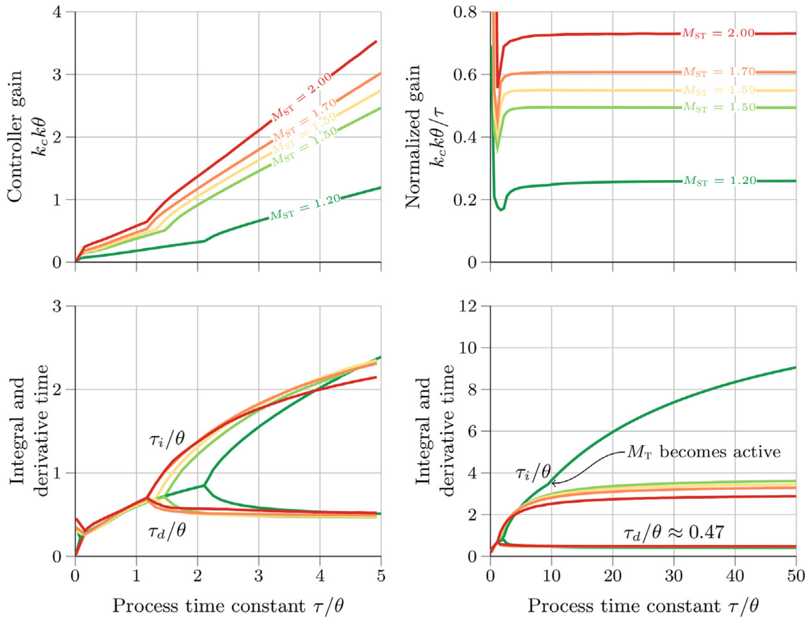

Fig. 5. PID control: IAE-optimal settings as a function of τ/θ for five values of M_{ST}.

$k_c k\theta/\tau = 0.414$ for $\tau/\theta = 50$, and 0.409 for $\tau/\theta = \infty$ (integrating process). This is close to the value $k_c k\theta/\tau = 0.5$ obtained with the SIMC-rule with $M_{ST} = 1.59$.

3.3. Optimal PID control

The IAE-optimal PID settings are shown graphically in Fig. 5 and are also given for $M_s = 1.59$ for four processes in Table 3. The optimal PID settings can be divided into the same regions as for PI control. Note that for a pure time delay process, it is optimal with PI control, that is, it is optimal to have $\tau_d = 0$ and $\tau_i = \theta/3$. Actually, if we allow for having the derivative time larger than the integral time, then we could interpret it differently, and say that for a pure time delay process, the optimal PID-controller is an ID-controller with $\tau_i = 0$ and $\tau_d = \theta/3$. We will see that this latter interpretation is consistent with the proposed improved SIMC PID-rule, whereas the first interpretation is consistent with the improved SIMC PI-rule.

For PID-control, the balanced region ($0.2 < \tau/\theta < 4$) can be divided in two. In the lower part ($\tau/\theta < 1.25$), the optimal derivative and integral time are the same, $\tau_i = \tau_d$, and increase with τ/θ. In the upper part, τ_i increases with τ/θ, whereas τ_d remains approximately constant. Note that the region with $\tau_i = \tau_d$ agrees with the recommendation of [25].[2] However, we see from Fig. 5 that $\tau_i = \tau_d$ is optimal only for a fairly small range of first-order plus time delay processes with τ/θ between about 0.2 and 1.25.

From Fig. 5 we see that the integral time (τ_i) is smaller than the process time constant (τ) for all processes with $\tau/\theta > 4$, whereas we found that $\tau_i \approx \tau$ was optimal in the balanced region for PI control. For given values of M_{ST}, the optimal PID controller gain is slightly larger than the optimal PI controller gain, and the integral action is also larger (with a lower value of τ_i).

For lag-dominant processes ($\tau/\theta > 4$), the normalized controller gain $k_c k\theta/\tau$ approaches a constant value as $\tau \to \infty$. For example, for $M_{ST} = 1.59$ we have $k_c k\theta/\tau \to 0.54$. The same can be observed for the integral and derivative times which for $M_{ST} = 1.59$ approach $\tau_i/\theta = 3.24$ and $\tau_d/\theta = 0.48$, respectively (Fig. 5, bottom right). For increasing M_{ST} values (less robustness), the optimal controller gain increases and the optimal integral time decreases. Interestingly, for all lag-dominant processes the optimal derivative time is $\tau_d \approx 0.47\theta$ almost independent of the M_{ST}-value.

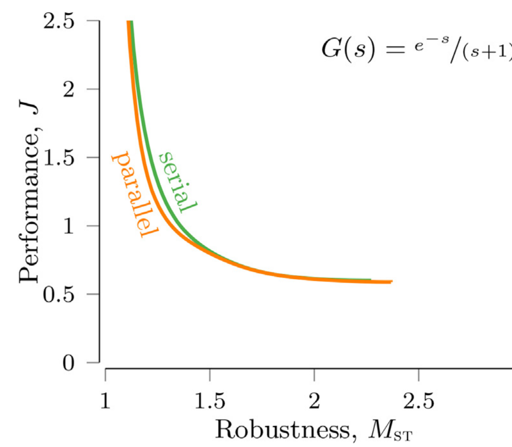

Fig. 6. PID control: Comparison of IAE-optimal performance (J) for serial PID control (2) and parallel PID control (3) for a process with $\tau/\theta = 1$.

3.4. Parallel vs. serial PID controller

The above optimization was for the serial PID controller in (2). A more general PID controller is the parallel, or ideal, PID controller in (3), which allows for complex zeroes. The parallel PID controller is better only for processes with τ/θ between 0.4 and 1.2, which is the region where $\tau_i = \tau_d$ (two identical real zeros) for the serial PID controller. Furthermore, the improvement with the parallel (ideal) PID form is very minor as illustrated in Fig. 6, which compares the IAE performance for a "balanced" process with $\tau/\theta = 1$. Therefore, the serial PID implementation in (2) is sufficient for first-order plus time delay processes.

[2] [25] recommend the integral time to be 4 times the derivative time for the parallel (ideal) PID controller, which for the serial (cascade) PID form corresponds to $\tau_i = \tau_d$, see (4).

Table 3
PID control: Comparison of optimal and SIMC controllers with M_{ST} = 1.59.

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

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

Table 4
SIMC PI controller and iSIMC PID controller with tuning constant $\tau_c = \theta$ and $\tau_c = \theta/2$, respectively.

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

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

4. The original and improved SIMC rules

4.1. Original SIMC rule

We consider the first-order with delay process in (1). The original SIMC PID tunings for this process give a PI controller [24]:

$$k_c = \frac{1}{k}\frac{\tau}{(\tau_c + \theta)}, \quad \tau_i = \min\left\{\tau, 4(\tau_c + \theta)\right\}. \tag{15}$$

Here the closed-loop time constant τ_c is an adjustable tuning parameter which is used to get the desired trade-off between output performance, robustness and input usage. For "tight control" (good performance) with acceptable robustness (M_s about 1.6 to 1.7), [24] recommends selecting $\tau_c = \theta$. However, in many cases "smooth control" is desired and we should use a larger value for τ_c.

4.2. "Improved" SIMC rule with derivative action (iSIMC)

In this paper, we propose the "improved" SIMC PID-rule for a first-order with delay process. Since an important feature of the SIMC rules is simplicity, we keep the same expressions for k_c and τ_i as in the original PI-rule in (15), but derivative action is added to improve the performance for a time delay,

$$\text{iSIMC}: \quad \tau_d = \theta/3 \tag{16}$$

Note that the iSIMC tunings are for the serial PID implementation in (2). For the more common parallel (ideal) PID implementation in (3), one must compute the factor $f = 1 + \tau_d/\tau_i$ and use the values in (4).

As seen from Fig. 5, the value $\tau_d = \theta/3$ is close to the optimal for a pure time delay process (with $\tau = 0$). For larger values of process time constant τ, the optimal value of τ_d is closer to $\theta/2$. However, we chose to use the smaller value $\tau_d = \theta/3$ in order to reduce possible other disadvantages of adding derivative action.

If we use the same value for the tuning constant (e.g. $\tau_c = \theta$) as for the original SIMC PI controller in (15), then the addition of the derivative action in (16) mainly improves robustness (lower M_S). However, the main reason for introducing derivative action is usually to improve performance, and to achieve this one should reduce τ_c. In the original SIMC rule [24] it was recommended to select $\tau_c = \theta$ to achieve "tight control" with acceptable robustness (M_s about 1.6 to 1.7). However, as will become clearer from the results below, for the iSIMC_PID rule we recommend reducing the value of τ_c and select $\tau_c \geq \theta/2$.

The SIMC PI-tunings parameters with $\tau_c = \theta$ and the iSIMC PID-tunings with $\tau_c = \theta/2$ are given for four first-order plus delay processes in Table 4. As seen from the table, iSIMC-PID-improves the IAE performance (J) by about 30% compared to the original SIMC PI controller, while keeping about the same robustness level (M_{ST} about 1.7).

Note that we have put "improved" in quotes for the iSIMC rule. Indeed, in the original SIMC paper, [24] considered adding the derivative time $\tau_d = \theta/2$ to counteract time delay, but concluded that it was probably not worth the increased complexity of the controller and the increased sensitivity to measurement noise and input usage. Therefore, in most practical situations in industry, the original SIMC PI-rule is most likely preferable. Nevertheless, if performance is important and τ_c is adjusted as mentioned above, then the results of this paper show (e.g. see Fig. 9) that significant improvements in performance may be achieved with the iSIMC rule. Additionally, we have found that PID control with iSIMC tunings is better in almost all respects than a well-tuned Smith Predictor controller [15].

4.3. Alternative improved SIMC rule without derivative action (iSIMC-PI) for delay dominant processes

Note that for a pure time delay process ($\tau = 0$), the iSIMC PID-controller in ((15–16)) is actually a ID-controller, since $k_c = 0$. As noted earlier this ID-controller (with $\tau_d = \theta/3$ and $\tau_i = 0$) may be realized instead as a PI-controller (with $\tau_i = \theta/3$ and $\tau_d = 0$). This is the basis for the following "improved" SIMC PI rule for a first-order plus delay process, denoted iSIMC-PI [12]:

$$\text{iSIMC} - \text{PI}: \quad k_c = \frac{1}{k}\frac{\tau + \theta/3}{(\tau_c + \theta)}, \quad \tau_i = \min\{\tau + \theta/3, \quad 4(\tau_c + \theta)\}. \tag{17}$$

Note that for a pure time delay process ($\tau = 0$), the iSIMC PID-controller in ((15–16)) and the iSIMC-PI PI-controller in (17) are identical. The iSIMC-PI tunings in (17) may give significant performance improvements benefits compared to the original SIMC PI-tunings for delay-dominant processes, but at the expense of larger input usage. However, for processes with $\tau > \theta/2$, approximately, we find that the benefits are marginal or even negative.

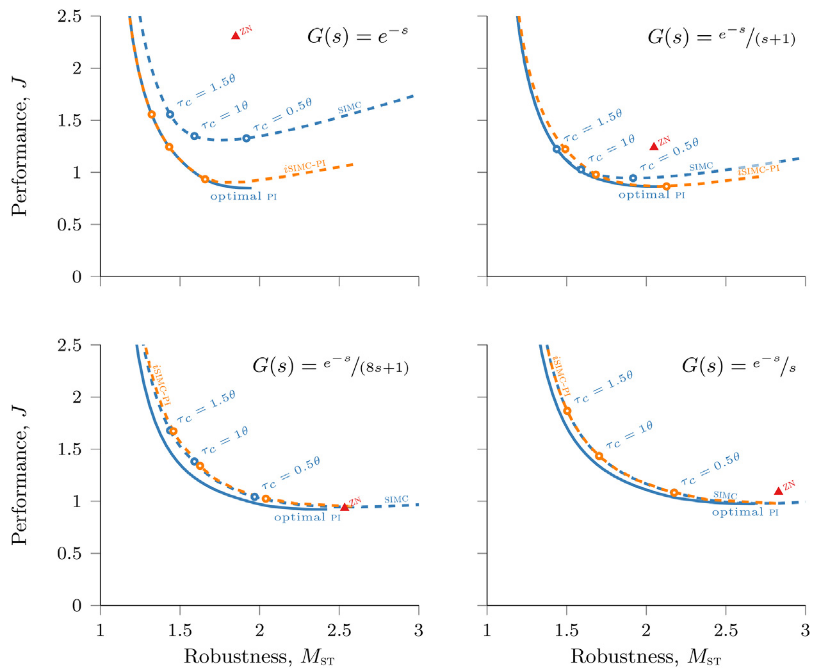

Fig. 7. PI control: Pareto optimal IAE-performance (J) for optimal-PI, SIMC (15), and *i*SIMC-PI (17) control for four processes. The trade-off for SIMC and *i*SIMC-PI is generated by changing the value of the closed-loop tuning constant τ_c.

5. Evaluation of the SIMC and *i*SIMC rules

5.1. SIMC PI-rule (original)

We compare in Fig. 7 the Pareto-optimal IAE performance (J) of the SIMC PI controller with the IAE-optimal PI controller for four different processes. The PI settings for $M_{ST} = 1.59$ are given in Table 2. In addition, the SIMC controllers for three specific choices of the tuning parameter,

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

are shown by circles. For the SIMC controller (Fig. 7), the trade-off curves were generated by varying the tuning parameter τ_c from a large to a small value. Except for the pure time delay process, the IAE-performance SIMC PI-controller is very close (within 10%) to the IAE-optimal PI controller for all robustness levels. In other words, by adjusting τ_c we can generate a close-to-optimal PI-optimal controller with a given desired robustness. Another important observation is that the default PID-recommendation for "tight" control, $\tau_c = \theta$ (as given by middle of the three circles), in all cases is located in a desired part of the trade-off region, well before we reach the minimum. Also, the recommended choice gives a fairly constant M_s-value, in the range from 1.59 to 1.7. From this we conclude that, except for the pure time delay process, there is little room to improve on the SIMC PI-rule, at least when performance and robustness are as defined above (J and M_s).

5.2. Improved SIMC PI rule (iSIMC-PI)

The main "problem" with the original SIMC rule is for pure time delay processes, where the IAE-performance (J) is about 40% higher than the optimal (Fig. 7). The proposed *i*SIMC-PI rule in (17) rectifies this. As seen from Fig. 7 (upper left), the proposed *i*SIMC-PI rule is almost identical to the IAE-optimal controller when τ_c is adjusted

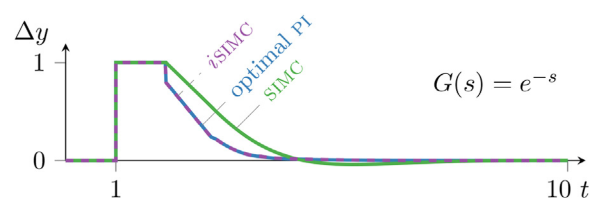

Fig. 8. PI-control:Time response for optimal, SIMC and *i*SIMC controllers ($M_{ST} = 1.59$) for an input/output disturbance (at time 0/1) for the pure time delay process $G(s) = e^{-s}$. For pure time delay process the input u response is equal to the output y response time shifted one time delay earlier. Also the response is the same for an input or output disturbance (but shifted). Note that for a pure time delay process there is no benefit of adding derivative action ($\tau_d = 0$ is optimal).

to give the same robustness (M_{ST}). This is further illustrated by the simulation in Fig. 8.

5.3. Improved SIMC PID rule (iSIMC)

Next, we consider PID control, that is, the addition of derivative action using $\tau_d = \theta/3$, as proposed with the *i*SIMC rule (16). We compare in Fig. 9 the IAE performance (J) of the *i*SIMC PID controller with the IAE optimal PID controller with the same robustness (M_{ST}) for four different processes (green curves). PID settings for $M_{ST} = 1.59$ are given in Table 3. To illustrate the benefits of derivative action we also show in Fig. 9 the curves with PI control (blue curves).

We see from Fig. 9 that the SIMC PID-controller (dashed green curve) is close to the optimal PID-controller (solid green line) for all four processes and all robustness levels. By considering the location of the middle green circles, we see that if we keep the value of τ_c unchanged at $\tau_c = \theta$, then adding derivative action mainly improves robustness. For example, for an integrating process and $\tau_c = \theta$, the value of M_{ST} is improved from 1.70 for PI to 1.46 for PID, but there is only a 6% improvement in performance. However, by reducing τ_c we can significantly improve performance for a given M_{ST} value. For the four processes, we see from Fig. 9 that $\tau_c = \theta/2$ (rightmost green circles) is a good choice for the tuning constant for the *i*SIMC PID-controller. Compared to SIMC PI-controller tuned with $\tau_c = \theta$,

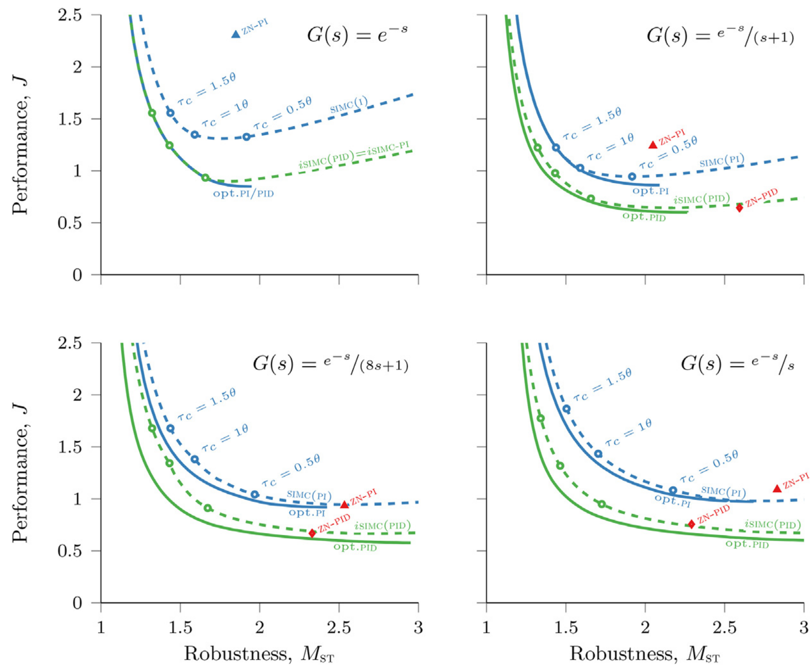

Fig. 9. Pareto-optimal IAE-performance (J) as a function of robustness M_{ST} for optimal PI and PID, and SIMC (15), *i*SIMC (16) and ZN for four processes. The trade-off for the various SIMC rules are generated by changing the value of the closed-loop tuning constant τ_c. PI controllers are shown with blue and PID controllers with green. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

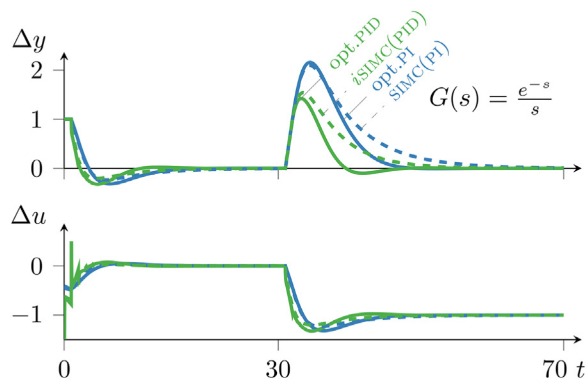

Fig. 10. Time response for optimal PI and PID compared with SIMC and *i*SIMC controllers (M_{ST} = 1.59) for an output disturbance (at time 0) and an input disturbances (at time 20), for an integrating process $G(s) = e^{-s}/s$. To get a proper system, a first order measurement filter with $\tau_f = 0.02$ was applied to the PID controllers.

this gives about 30% better IAE-performance and similar robustness (M_{ST} about 1.7).

We said that the SIMC PID-controller is close to the optimal PID-controller. However, we see from the two lower plots in Fig. 9 that the performance loss is somewhat larger for processes with large time constants. To study this further, we compare in Fig. 10 the step responses for various PI and PID controllers for an integrating process. Because of a larger integral time, the SIMC and *i*SIMC controllers settle more slowly than the optimal PI and PID controllers for both input and output disturbances. This results in a 22% higher average IAE-performance (J) for the *i*SIMC PID-controller when compared with the optimal PID controller. However, it is usually the maximum deviation that is of main concern in industry. Because of a larger controller gain, the SIMC controllers have roughly the same peak deviation as the optimal PI and PID controllers for input disturbance (see Fig. 10), and a smaller overshoot for output disturbances (setpoint) than the optimal. Thus, we conclude that the performance of the SIMC controllers are better than indicated from the IAE-value (J), when we take into account other aspects of performance. In conclusion, also for PID control, we conclude that *i*SIMC is close to the optimal PID-controller, so the benefit of looking for even more "improved" rules for first-order plus time delay processes is limited.

6. Discussion

6.1. Input usage and filtering

Input usage is an important aspect for control, but have not been explicitly treated in this work. From Fig. 1 we have

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

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

The important new transfer function is therefore KS, and by limiting its peak one can adjust input usage related to measurement noise and output disturbances [18]. For PI control, KS has a peak at intermediate frequencies which is approximately [3]:

$$\|KS\|_{\infty}^{\text{PI}} \approx k_c M_s. \tag{18}$$

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

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

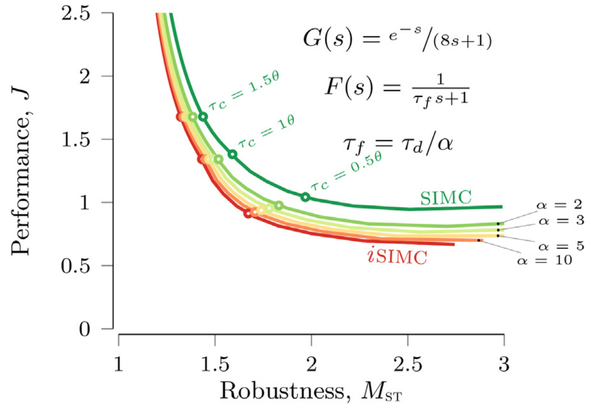

Fig. 11. Performance/robustness trade-off curve for the *i*SIMC (PID) controller with measurement filtering. The SIMC (PI) controller corresponds to the case with $\alpha = 1$.

ing peak will depend heavily on the selected filter time constant. With a first or second order measurement filter

$$F_1(s) = \frac{1}{\tau_f s + 1} \quad \text{or} \quad F_2(s) = \frac{1}{(\tau_f s)^2 + \sqrt{2}\tau_f s + 1}, \tag{19}$$

the high-frequency peak can be approximated by

$$\|KS\|_\infty^{\text{PID}} \approx \alpha k_c \tag{20}$$

where $\alpha = \tau_d/\tau_f$. Note that τ_d here is for the cascade PID-controller in (2) and not for the ideal form in (3). Typically, α is between 5 and 10. The ratio in input magnitude between PI and PID related to measurement noise and output disturbances can then be expressed as

$$\frac{\|KS\|_\infty^{\text{PID}}}{\|KS\|_\infty^{\text{PI}}} \approx \frac{k_c[\text{PID}]}{k_c[\text{PI}]} \frac{\alpha}{M_s[\text{PI}]}. \tag{21}$$

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

$$\frac{\|KS\|_\infty^{\text{iSIMC(PID)}}}{\|KS\|_\infty^{\text{SIMC(PI)}}} \approx \frac{1.33}{1.6}\alpha \approx 0.8\alpha \tag{22}$$

where we have assumed M_s[PI] to be 1.6. Since α is typically larger than 5, this means that the improved IAE performance of PID control may require an input magnitude related to measurement noise and output disturbances which is at least 4 times larger than for PI control.[3]

Trade-off curves for *i*SIMC with different first-order measurement filters are shown in Fig. 11. For the recommended PID tuning, $\tau_c = \theta/2$, performance and robustness will deteriorate with increased measurement filtering. With $\tau_c = \theta/2$ and $\alpha = 3$, the robustness is quite low, and a retuning of the controller by selecting a larger τ_c might be necessary. With $\alpha = 1$, we recover the original SIMC PI-controller for which we recommend $\tau_c = \theta$ to get a good trade-off between performance and robustness.

Based on Fig. 11, we recommend for PID-control to choose α in the range from 5 to 10, which gives most of the benefit of the D-action. The high-frequency input usage may then increase by a factor 4 to 8 compared to PI-control. This increase in input usage may be undesirable, so for many real process applications where performance is not the key issue, the original SIMC rule, which gives a PI-controller, is the best choice.

The above approach assumes that noise filtering can be handled with the parameter α, after the PID parameters have been slected. If noise filtering is an important factor, an iterative design approach can be used in combination with SIMC to ensure that the controller has both good robustness and low noise sensitivity [23]. Indeed, our approach can easily be extended to handle noise filtering more explicitly by using an appropriate constraint, e.g. on KS. This is treated in a separate paper [NEW] . Note that if the measurement filter actually enhances noise-less performance, then we are no longer talking about a PID controller, but rather a PIDF controller with four adjustable parameters.

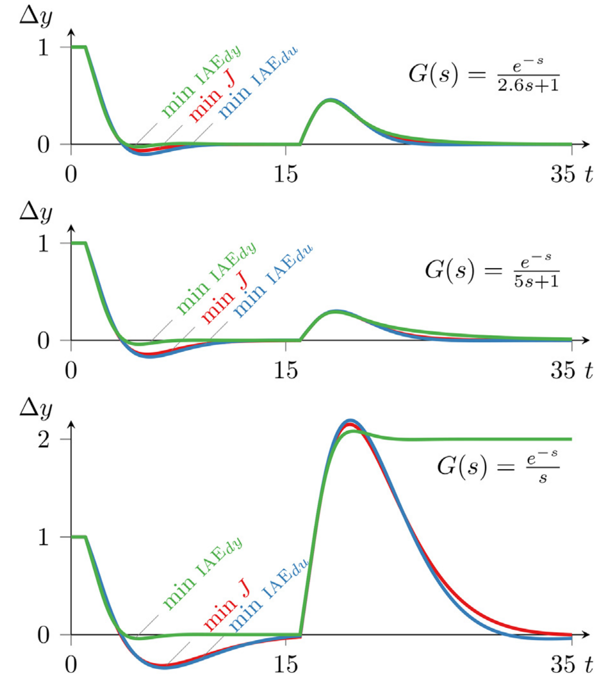

Fig. 12. PI-control: Comparison of step disturbance responses of IAE optimal controller which minimizes J, with the two reference controllers that consider disturbances only at the output (IAE°_{dy}) and input (IAE°_{du}), respectively. All controllers have $M_{ST} = 1.59$ (Table 1).

6.2. Trade-off between input and output disturbance response

As already noted from Table 1 and observed from the simulations in Fig. 12, the optimal controller that minimizes the average IAE-performance (J) in (12), puts more emphasis on disturbance rejection at the input (IAE_{du}) than at the output (IAE_{dy}), especially for larger values of the process time constant. For example, for an integrating process we find $\tilde{IAE}_{du} = 1.02$ (close to optimal for input disturbance), whereas $\tilde{IAE}_{dy} = 1.99$ (twice the optimal). This is further illustrated in Fig. 13, where we show ratio between the two terms in the IAE-performance index J [17], which in this paper we term *controller balance*,

$$\text{controller balance} = \frac{\text{IAE}_{du}}{\text{IAE}^\circ_{du}} \Big/ \frac{\text{IAE}_{dy}}{\text{IAE}^\circ_{dy}}, \tag{23}$$

as a function of τ/θ. From Fig. 13, we also note that for time constants less than about 3θ, the optimal controller has roughly equal weight on input and output (also seen in Fig. 12, top). For larger time constants, the emphasis shifts towards input disturbances.

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

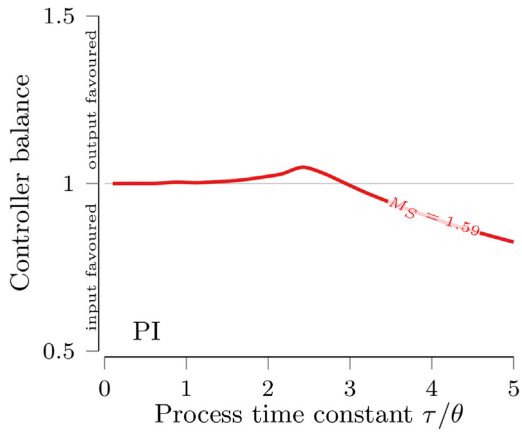

Fig. 13. PI-control: Controller balance between input and output disturbances, defined as $(\mathrm{IAE}_{du}/\mathrm{IAE}^\circ_{du})/(\mathrm{IAE}_{dy}/\mathrm{IAE}^\circ_{dy})$, for optimal PI control as a function of the time constant τ/θ ($M_{ST}=1.59$).

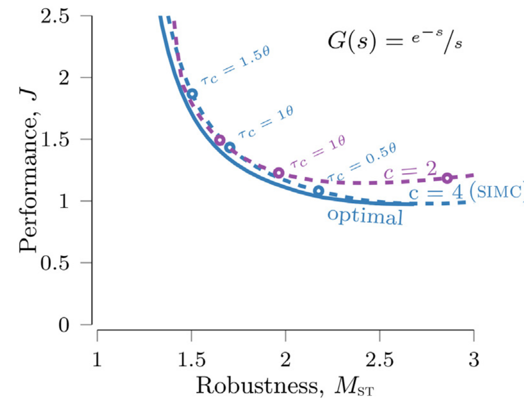

Fig. 14. PI control: evaluation of suggested modification $c=2$ in (24) for integrating process.

Interestingly, if we use a cost function with only a small weight (1%) on input disturbances

$$J(p) = \left(0.99\,\frac{\mathrm{IAE}_{dy}(p)}{\mathrm{IAE}^\circ_{dy}} + 0.01\,\frac{\mathrm{IAE}_{du}(p)}{\mathrm{IAE}^\circ_{du}}\right), \tag{24}$$

we find for an integrating process the optimal settings $k_c=0.462$ and $\tau_i=12.2\theta$, with $I\tilde{A}E_{du}=1.72$ and $I\tilde{A}E_{dy}=1.91$. We notice that there is only a minor improvement in setpoint performance (IAE_{dy} decreases about 4%), whereas disturbance rejection is much worse (IAE_{du} increases about 69%). The conclusion is this that we may put emphasis mainly on input disturbances when tuning PI-controllers for lag-dominated processes.

6.3. Further evaluation of SIMC PI rule for integrating processes

When comparing the optimal PI-settings with the original SIMC rule for an integrating process, we find that the SIMC integral time is larger than the optimal (Fig. 4, bottom). Specifically, for an integrating process with $\tau_c=\theta$ (giving $M_{ST}=1.70$), the SIMC rule gives $\tau_i/\theta=8$, whereas the optimal performance (J) for the same robustness is with $\tau_i/\theta=5.6$. This indicates that the SIMC rule puts more emphasis on output disturbances than input disturbances, than for the IAE-optimal controller with equal weighting. To shift the trade-off between output (setpoint) and input disturbance, one may introduce an extra parameter in the tuning rule [1,8]. [16] suggested to introduce an extra servo/regulator trade-off parameter c in the expression for the integral time,

$$\tau_i = \min\{\tau,\ c(\tau_c+\theta)\}, \tag{25}$$

where $c=4$ gives the original SIMC rule. However, introducing an extra parameter adds complexity, and the potential performance benefit of approximately 10% (see Fig. 7) does not seem sufficiently large to justify it. Nevertheless, one may consider choosing another (lower) fixed value for c, and [16] suggests using $c=2$ to improve performance for input disturbances. If we use the recommended tuning $\tau_c=\theta$, we find indeed that IAE performance J is improved compared to SIMC (see Fig. 14). However, robustness is worse, with M_{ST} close to 2 (where the SIMC rule gives M_{ST} close to 1.7). More importantly, as seen from Fig. 14 the SIMC performance is better if we decrease τ_c to get the same robustness in terms of M_{ST}. In fact, SIMC is closer to the Pareto optimal curve for most values of M_{ST}. Actually, a better fixed value would be $c=3$. However, changing the parameter c causes the recommended tuning $\tau_c=\theta$ to shift to the less robust region. In summary, we find that the value $c=4$ in the original SIMC rule provides a well balanced servo/regulator trade-off. To improve performance for input disturbances on an integrating process, we recommend decreasing the tuning constant τ_c/θ, say to around 0.7, rather than changing the value of c.

6.4. iSIMC for second-order plus delay process

For a second-order plus delay process,

$$G = \frac{ke^{-\theta s}}{(\tau_1 s+1)(\tau_2 s+1)} \tag{26}$$

where $\tau_1 > \tau_2$, the original SIMC rule gives a PID controller on the series form (2) with

$$k_c = \frac{1}{k}\frac{\tau}{(\tau_c+\theta)},\quad \tau_i = \min\left\{\tau,\ \ 4(\tau_c+\theta)\right\},\quad \tau_d = \tau_2. \tag{27}$$

The direct extension of the *i*SIMC rule would be to add another derivative term, $\left(\frac{\theta}{3}s+1\right)$, to the numerator of the PID controller in (2). First, this would not be a standard industrial controller and, second, it would give even more aggressive input usage. Thus, to get a PID controller, the following modified derivative time is recommended[4]

$$i\mathrm{SIMC}: \tau_d = \tau_2 + \theta/3 \tag{28}$$

with the controller gain and integral time as given in (27). To get a good trade-off between performance and robustness, we may select $\tau_c=\theta$, but τ_c may be reduced towards $\theta/2$ for processes where τ_2 is smaller than $\theta/3$. Again, to get settings for the parallel (ideal) PID-controller in (3) one must compute the factor $f=1+\tau_d/\tau_i$, and use (4).

6.5. Ziegler-Nichols tuning rule

We also show in Fig. 9 by red triangles the location of the classical [25] Ziegler-Nichols (ZN) PI- and PID-controllers. Since the ZN rules have no tuning parameter we get a single point in Fig. 9. With exception of the pure time delay process (where ZN-PID is unstable and ZN-PI has very poor performance), the IAE-performance for ZN is very good, but the ZN controllers are located in the "flat" trade-off region with poor robustness (large M_{ST} value).

The Ziegler-Nichols PID tuning rules were the by far most used rules for about 50 years, up to about 1990. The very poor performance of the ZN rules for pure time delay processes may then partly explain the myth that "time delay compensators", such as the Smith

[4] If τ_2 is very large, specifically if $\tau_2 > 4(\tau_c+\theta)$, then one should approximate the process as a double integrating process, $G(s)\approx k''e^{-\theta s}/s^2$ with $k''=k/(\tau_1\tau_2)$, and use the PID-tunings for a double integrating process [24].

Predictor, may give significant performance benefits compared to PI- or PID-control for processes with large time delays [15].

7. Conclusion

The IAE-optimal PI- and PID-settings for a first-order plus delay process (1) are shown for various robustness levels (as expressed by the M_s-value) in Figs. 4 and 5, respectively. However, in practice, we recommend using the SIMC-rules for PI- and PID-tuning.

For PI-control, Fig. 7 shows that the "original" SIMC rule [24] in (15) gives close-to optimal PI-performance. That is, by adjusting the tuning constant τ_c to get a desired robustnes, we can closely track the Pareto-optimal trade-off curve between performance and robustness. The only exception is for delay-dominant first order plus time delay processes, where the SIMC proportional gain is too small, but this can be corrected for by using the *i*SIMC – PI rule in (17).

For PID-control, we propose the *i*SIMC rule where derivative action with $\tau_d = \theta/3$ (16) is added. Note that this is for the cascade PID-controller in (2). Fig. 9 shows that this rule gives close-to optimal PID-performance, even for delay-dominant processes. For a pure time delay process, the *i*SIMC PID-controller is an ID-controller which can by rewritten to give the *i*SIMC – PI rule in (17).

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

References

[1] S. Alcántara, C. Pedret, R. Vilanova, On the model matching approach to PID design: analytical perspective for robust servo/regulator tradeoff tuning, J. Process Control 20 (5) (2010) 596–608.
[2] K.J. Åström, T. Hägglund, Automatic Tuning of PID Controllers, Instrument Society of America (ISA), 1988.
[3] K.J. Åström, T. Hägglund, Advanced PID Control, The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, 2006.
[4] K.J. Åström, C.C. Hang, P. Persson, W.K. Ho, Towards Intelligent PID Control, vol. 28, 1992.
[5] W.L. Bialkowski, Control of the pulp and paper making process, in: W.S. Levine (Ed.), The Control Handbook, CRC Press and IEEE Press, 1996, pp. 1219–1243 (chapter 72).
[6] S.P. Boyd, C.H. Barratt, Linear Controller Design: Limits of Performance, Prentice Hall, Englewood Cliffs, NJ, 1991.
[7] E.B. Dahlin, Designing and tuning digital controllers, Instrum. Control Syst. 41 (6) (1968) 77–83.
[8] D. Di Ruscio, On tuning PI controllers for integrating plus time delay systems, Model. Identif. Control 31 (4) (2010) 145–164.
[9] O. Garpinger, T. Hägglund, A software tool for robust PID design., in: Proc. 17th IFAC World Congress, Seoul, Korea, 2008.
[10] O. Garpinger, T. Hägglund, K.J. Åström, Performance and robustness trade-offs in PID control, J. Process Control 24 (5) (2014) 568–577.
[11] C. Grimholt, S. Skogestad, Optimal PI-control and verification of the SIMC tuning rule, in: IFAC Conference on Advances in PID Control (PID'12), The International Federation of Automatic Control, March, 2012.
[12] C. Grimholt, S. Skogestad, The SIMC method for smooth PID controller tuning, in: R. Vilanova, A. Visioli (Eds.), PID Control in the Third Millennium – Lessons Learned and New Approaches, Springer, 2012.
[13] C. Grimholt, S. Skogestad, Optimal PID-control on first order plus time delay systems & verification of the SIMC rules, 10th IFAC International Symposium on Dynamics and Control of Process Systems (2013).
[14] C. Grimholt, S. Skogestad, Optimization of fixed-order controllers using exact gradients, J. Process Control (2018) (Unpublished).
[15] C. Grimholt, S. Skogestad, Should we forget the Smith predictor? in: 3rd IFAC Conference on Advances in PID Control, Ghent, Belgium, 9–11 May 2018, 2018.
[16] F. Haugen, Comparing PI tuning methods in a real benchmark temperature control system Model. Identif. Control 31 (3) (2010) 79–91.
[17] M. Huba, Performance measures, performance limits and optimal PI control for the IPDT plant, J. Process Control 23 (4) (2013) 500–515.
[18] B. Kristiansson, B. Lennartson, Robust and optimal tuning of PI and PID controllers, IEE Proc.-Control Theory Appl. 149 (1) (2002) 17–25.
[19] B. Kristiansson, B. Lennartson, Evaluation and simple tuning of PID controllers with high-frequency robustness, J. Process Control 16 (2) (2006) 91–102.
[20] J. Lee, W. Cho, T.F. Edgar, Simple analytic PID controller tuning rules revisited, Ind. Eng. Chem. Res. 53 (13) (2014) 5038–5047.
[21] A. O'Dwyer, Handbook of PI and PID Controller Tuning Rules, 2nd ed., Imperial College Press, 2006.
[22] D.E. Rivera, M. Morarl, S. Skogestad, Internal model control. 4. PID controller design, Ind. Eng. Chem. Process Des. Dev. 25 (1986) 252–256.
[23] V.R. Segovia, T. Hägglund, K.J. Åström, Design of measurement noise filters for PID control, IFAC World Congress 2014, vol. 19 (2014) 8359–8364.
[24] S. Skogestad, Simple analytic rules for model reduction and PID controller tuning, J. Process Control 13 (4) (2003) 291–309.
[25] J.G. Ziegler, N.B. Nichols, Optimum settings for automatic controllers, Trans. ASME 64 (11) (1942) 759–768.