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

A R T I C L E   I N F O
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

A B S T R A C T
The first-order plus delay process model with parameters $k$ (gain), $\tau$ (time constant) and $\theta$ (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 $\tau_d$ 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 $(k, \tau, \theta)$. 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 back-setting the controller parameters. Typically, a first-order response is specified with closed-loop time constant $\tau_c (\text{called } \lambda \text{ by Dahlin}).$ Importantly, $\tau_c$ (or $\lambda$) 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 $\tau$. 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 $\tau_i$ 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 $\theta$ is the effective time delay of the process.
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

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

where $k$ is the process gain, $\tau$ is the process time constant, and $\theta$ is the process time delay. We mainly consider the serial (cascade) form PID controller,

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

where $k_c$, $\tau_i$, and $\tau_d$ 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

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

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

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

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

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

Normally, we use is a first-order filter,

$$F = \frac{1}{\tau_fs + 1}.$$ (6)

Note that $\tau_f$ 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 $\tau_f = \tau_f / \alpha$ where $\alpha$ 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 $M_i$-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 ($M_i$). The resulting Pareto-optimal trade-off betwee performance and robustness is subsequently used to evaluate the SIMC PI and PID rules.

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 iSIMC and ISIMC-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),

$$\text{IAE} = \int_0^\infty |y(t) - y_s(t)|dt.$$ (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 $d_y$,

$$J(p) = 0.5 \left(\frac{\text{IAE}_{dy}(p)}{\text{IAE}_{dy}} + \frac{\text{IAE}_{du}(p)}{\text{IAE}_{du}}\right)$$ (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 $M_i = 1.59$, 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 $M_i = 1.59$, it is the resulting $M_i$-value for a SIMC-tuned PI-controller with $\tau = \theta$ on a first-order plus delay process with $\tau \leq 8\theta$. 

---

**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 ($d_u$).
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 $\theta$, 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],

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

(9)

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

$$M_s = \max_{w} S(jw) = \|S(jw)\|_{\infty},$$

$$M_T = \max_{w} T(jw) = \|T(jw)\|_{\infty},$$

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

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

(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].

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

(11)

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

### 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:

$$\min_p J(p) = 0.5 \frac{\text{IAE}_{dy}(p)}{\text{IAE}_{dy}} + \frac{\text{IAE}_{du}(p)}{\text{IAE}_{du}}$$

subject to: $M_s(p) \leq M^{ub}$

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

(12)

(13)

(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).
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 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 $\tau$ 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 $\tau/\theta$ 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 $\tau_i$:

- **Delay-dominant**
  - $\tau/\theta < 0.4$
  - $\tau_i \approx \theta/3$

- **Balanced**
  - $0.4 < \tau/\theta < 4$
  - $\tau_i \approx \tau$

- **Lag-dominant**
  - $\tau/\theta > 4$
  - $\tau_i \approx k\theta$

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 ($K(s) = K_i/s$, corresponding to $\tau_i \to 0$) 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 ISIMC – PID and 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, $k_{c\theta}/\tau$ approaches a constant value as $\tau$ goes to infinity (Fig. 4, upper right). For example, for $M_{ST}=1.59$, the optimal value is

![Fig. 4. PI control: IAE-optimal settings as a function of $\tau/\theta$ for five values of $M_{ST}$.](image)

### Table 2

<table>
<thead>
<tr>
<th>Process</th>
<th>Optimal PI</th>
<th>SIMC</th>
<th>$\delta$SIMC – PI</th>
<th>$M_{ST}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k_c$</td>
<td>$\tau_i$</td>
<td>$J$</td>
<td>$k_i$</td>
</tr>
<tr>
<td>$e^{-s}$</td>
<td>0.20</td>
<td>0.32</td>
<td>1.00</td>
<td>0</td>
</tr>
<tr>
<td>$\frac{s+1}{s+0.2}$</td>
<td>0.54</td>
<td>1.10</td>
<td>1.01</td>
<td>0.50</td>
</tr>
<tr>
<td>$\frac{s}{s+1}$</td>
<td>3.47</td>
<td>4.04</td>
<td>1.23</td>
<td>4.00</td>
</tr>
<tr>
<td>$\frac{s^2}{s+1}$</td>
<td>0.41</td>
<td>6.22</td>
<td>1.50</td>
<td>0.45</td>
</tr>
</tbody>
</table>

* This is an I controller with integral gain $k_i = k_c/\tau_i = 0.5$. 
However, \( k_{c}k_{0}/\tau = 0.414 \) for \( \tau/\theta = 50 \), and 0.409 for \( \tau/\theta = \infty \) (integrating process). This is close to the value \( k_{c}k_{0}/\tau = 0.5 \) 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 \( M_{ST} = 1.59 \) 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-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 \( \tau/\theta \). In the upper part, \( \tau_{i} \) increases with \( \tau/\theta \), whereas \( \tau_{d} \) 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 \( \tau/\theta \) between about 0.2 and 1.25.

From Fig. 5 we see that the integral time \( (\tau_{i}) \) is smaller than the process time constant \( (\tau) \) 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 \( \tau_{i} \)).

For lag-dominant processes (\( \tau/\theta > 4 \)), the normalized controller gain \( k_{c}k_{0}/\tau \) approaches a constant value as \( \tau \to \infty \). For example, for \( M_{ST} = 1.59 \) we have \( k_{c}k_{0}/\tau \approx 0.54 \). 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).

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.

### 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 \( \tau/\theta \) between 0.4 and 1.2, which is the region where \( \tau_{i} = \tau_{d} \) (two identical real zeroes) 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).
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]:

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

Here the closed-loop time constant \( \tau_c \) 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 (\( M_\eta \) 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 \( \tau_c \).

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 \( \tau_i \) as in the original PI-rule in (15), but derivative action is added to improve the performance for a time delay.

\[ \text{ISIMC} : \quad \tau_d = \theta / 3 \quad (16) \]

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 \( \tau_c \), the optimal value of \( \tau_d \) 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 \( M_\eta \)). However, the main reason for introducing derivative action is usually to improve performance, and to achieve this one should reduce \( \tau_c \). In the original SIMC rule [24] it was recommended to select \( \tau_c = \theta \) to achieve "tight control" with acceptable robustness (\( M_\eta \) 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 \( \tau_c \) and selecting \( \tau_c \geq \theta / 2 \).

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 \( (f) \) by about 30% compared to the original SIMC PI controller, while keeping about the same robustness level \( (M_\eta) \) 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 \( \tau_c \) 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 \( \tau_i = 0 \) 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]:

\[ k_c = \frac{1}{k} \frac{\tau + \theta / 3}{k (\tau_c + \theta)}, \quad \tau_i = \min \{ \tau + \theta / 3, \quad 4(\tau_c + \theta) \}. \quad (17) \]

Note that for a pure time delay process (\( \tau = 0 \)), the ISIMC PID-control 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.
5. Evaluation of the SIMC and iSIMC 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,

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

are shown by circles. For the SIMC controller (Fig. 7), the trade-off curves were generated by varying the tuning parameter $\tau_e$ 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 $\tau_e$ 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, $\tau_e = \theta$ (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 $M_I$-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 $M_I$).

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 iSIMC-PI rule in (17) rectifies this. As seen from Fig. 7 (upper left), the proposed iSIMC-PI rule is almost identical to the IAE-optimal controller when $\tau_e$ is adjusted 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 SIMC rule (16). We compare in Fig. 9 the IAE performance ($J$) of the iSIMC 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 $\tau_c$ unchanged at $\tau_e = \theta$, then adding derivative action mainly improves robustness. For example, for an integrating process and $\tau_e = \theta$, 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 $\tau_e$ we can significantly improve performance for a given $M_{ST}$ value. For the four processes, we see from Fig. 9 that $\tau_e = \theta/2$ (rightmost green circles) is a good choice for the tuning constant for the iSIMC PID-controller. Compared to SIMC PI-controller tuned with $\tau_e = \theta$,
this gives about 30% better IAE-performance and similar robustness ($M_S$ 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 $\delta$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 $\delta$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 $\delta$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

$$u = -T \ du - KS (d_f + n).$$

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 $M_I$.

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]:

$$\|KS\|_\infty^{PI} = k_c M_I.$$  \hspace{1cm} (18)

Here, $M_I$ 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-
ing peak will depend heavily on the selected filter time constant. With a first or second order measurement filter
\[ F_1(s) = \frac{1}{\tau_f s + 1} \] or
\[ F_2(s) = \frac{1}{(\tau_f s)^2 + \sqrt{2}\tau_f s + 1}, \tag{19} \]
the high-frequency peak can be approximated by
\[ ||K||_{\infty}^{PID} \approx ak_c ||K||_{\infty}^{PI} \]  \tag{20}
where \( a = \tau_d / \tau_f \). Note that \( \tau_d \) here is for the cascade PID-controller in (2) and not for the ideal form in (3). Typically, \( a \) 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
\[ \frac{||K||_{\infty}^{SIMC(PID)}}{||K||_{\infty}^{SIMC(PI)}} \approx \frac{k_c^{PID}}{k_c^{PI}} \frac{\alpha}{M_f^{PI}} \]  \tag{21}
With the recommended tight tuning (\( \tau_c = \theta \) for SIMC PI and \( \tau_c = \theta / 2 \) for SIMC PID), we get
\[ \frac{k_c^{PID}}{k_c^{PI}} = \left( \frac{\theta + \theta}{(\theta / 2 + \theta) = 1.33 \text{ and the ratio in input usage can be expressed as} \right. \]
\[ \frac{||K||_{\infty}^{SIMC(PID)}}{||K||_{\infty}^{SIMC(PI)}} \approx 1.33 \frac{\alpha}{1.6} \approx 0.8a \]  \tag{22}
where we have assumed \( M_f^{PI} \) to be 1.6. Since \( a \) 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 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 \( \tau_c \) 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 \( \alpha \) 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

\[ G(s) = e^{-s} / (8s + 1) \]
\[ F(s) = 1 / (\tau_f s + 1) \]
\[ \tau_f = \tau_d / \alpha \]
\[ \|K\|_{\infty}^{PID} \approx ak_c \|K\|_{\infty}^{PI} \]
\[ \frac{\|K\|_{\infty}^{SIMC(PID)}}{\|K\|_{\infty}^{SIMC(PI)}} \approx \frac{k_{c}^{PID}}{k_{c}^{PI}} \frac{\alpha}{M_{f}^{PI}} \]
\[ \frac{\|K\|_{\infty}^{SIMC(PID)}}{\|K\|_{\infty}^{SIMC(PI)}} \approx 1.33 \frac{\alpha}{1.6} \approx 0.8a \]

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 \( \tau_f \) should be considered a degree of freedom in the optimization problem.

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 \( (\text{IAE}_{dx}) \) than at the output \( (\text{IAE}_{dy}) \), especially for larger values of the process time constant. For example, for an integrating process we find \( \text{IAE}_{dx} = 1.02 \) (close to optimal for input disturbance), whereas \( \text{IAE}_{dy} = 1.99 \) (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,

\[ \text{controller balance} = \frac{\text{IAE}_{dx}}{\text{IAE}_{dy}} \]
Interestingly, if we use a cost function with only a small weight (1%) on input disturbances

\[ J(p) = \left( 0.99 \frac{IAE_{dy}(p)}{IAE_{dy}} + 0.01 \frac{IAE_{du}(p)}{IAE_{du}} \right) \]  

we find for an integrating process the optimal settings \( k_c = 0.462 \) and \( \tau_i = 12.2\theta \), with \( IAE_{du} = 1.72 \) and \( IAE_{dy} = 1.91 \). We notice that there is only a minor improvement in setpoint performance (IAE\(_{dy}\) decreases about 4%), whereas disturbance rejection is much worse (IAE\(_{du}\) increases about 69%). The conclusion is this that we may put emphasis primarily 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 rule, 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 \cite{ZN1952, SSM1987} suggested to introduce an extra servo/regulator trade-off parameter \( c \) in the expression for the integral time,

\[ \tau_i = \min(\tau_c, c(\tau_c + \theta)), \]  

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 \cite{SSM1987} 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 \( \tau_c \) 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 \( \tau_c/\theta \), say to around 0.7, rather than changing the value of \( c \).

### 6.4. SIMC for second-order plus delay process

For a second-order plus delay process,

\[ G(s) = \frac{k e^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \]  

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

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

The direct extension of the iSIMC rule would be to add another derivative term, \( \frac{2}{s^2} + 1 \), 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\(^4\)

\[ iSIMC : \tau_d = \tau_2 + \theta/3 \]  

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 \( \tau_c \) may be reduced towards \( \theta/2 \) for processes where \( \tau_2 \) 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 \cite{ZN1952} 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

\[^4\text{If } \tau_d \text{ is very large, specifically if } \tau_2 > 4(\tau_c + \theta), \text{ then one should approximate the process as a double integrating process, } G(s) = k e^{-\theta s}/s^2 \text{ with } k' = k/(\tau_1 \tau_2), \text{ and use the PID-tunings for a double integrating process } (24).\]
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 $M_r$-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 $\tau_2$ to get a desired robustness, we can closely track the Pareto-optimal trade-off curve between performance and robustness. The only exception is for delay-dominant first order plus time delay processes, where the SIMC proportional gain is too small, but this can be corrected for by using the $iSIMC – PI$ rule in (17).

For PID-control, we propose the $iSIMC$ 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 $SIMC$ PID-controller is an ID-controller which can be rewritten to give the $iSIMC – PI$ rule in (17).

The improved performance/robustness trade-off of the $iSIMC – PI$ and $iSIMC$ 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