Optimal PID-Control on First Order Plus Time Delay Systems & Verification of the SIMC Rules

Chriss Grimholt **Sigurd Skogestad***

Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway
**e-mail: skoge@ntnu.no*

Abstract: Optimal PID-settings are found for first-order with delay processes for specified levels of robustness (M_S-value) and compared with an extended SIMC-rule. Optimality (performance) is defined in terms of the integrated absolute error (IAE) for combined step changes in load output and input disturbances. The SIMC-rules gives a PI-controller for first order systems and no recommendation are given for tuning the derivative part. We propose an extended SIMC-rule where the the time delay is counteracted by introducing derivative action with $\tau_D = \theta/3$. The modification was found to give surprisingly good settings with near Pareto-optimal performance. However, to obtain the improvement over PI control τ_c should be reduced to about half of the recommended value $\tau_c = \theta$.

Keywords: Optimality, PI controllers, Pareto-optimal, robustness, performance evaluation, and closed-loop model approximation.

1. INTRODUCTION

We investigate optimal tunings for a first-order plus time delay process,

$$G(s) = \frac{k}{(\tau_1 s + 1)} \cdot e^{-\theta s} \tag{1}$$

where k is the process gain, τ_1 is the time constants, and θ is the time delay. We consider only the cascade form PID-controller

$$K(s) = K_c \left(\frac{\tau_I s + 1}{\tau_I s}\right)(\tau_D s + 1) \tag{2}$$

where K_c, τ_I and τ_D are the controller gain, integral time and derivative time. For other notation, see Figure 1.

In practice, the measurement is usually filtered. For example, by use of the controller

$$K(s) = K_c \left(\frac{\tau_I s + 1}{\tau_I s}\right)\left(\frac{\tau_D s + 1}{\tau_F s + 1}\right) \tag{3}$$

Measurement filtering is not included as a part of our tuning problem. For a PID-controller, the measurement filter time constant (τ_F) should anyway be selected such that the controller characteristics is not significantly changed, which for the controller (3) implies $\tau_F < \tau_D/3$, approximately. If the filter time constant (τ_F) is selected to enhance performance, then we are no longer talking about a PID-controller, but a PIDF-controller with four adjustable parameters.

Optimality is generally difficult to define as there are many issues to consider, including:

- Output performance
- Robustness
- Input usage
- Noise sensitivity

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

(1) Output performance
(2) Robustness, input usage and noise sensitivity

Pareto-optimality applies to multiobjective problems, and means that no further improvement can be made in objective 1 without sacrificing objective 2. The idea is

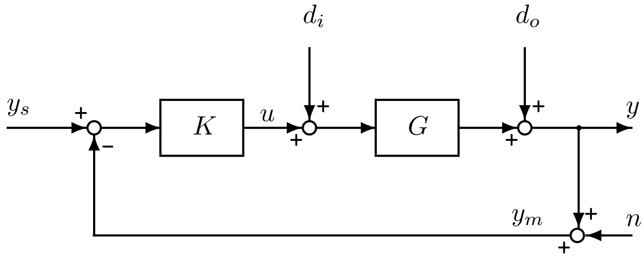

Fig. 1. Block diagram of the feedback control system. We may treat an output disturbance (d_o) as a special case of setpoint change (y_s)

then to find the Pareto-optimal controller, and compare with the SIMC-tuning.

The SIMC method for PID-controller tuning (Skogestad, 2003) has already found wide industrial usage. The SIMC-rules are analytically derived, and from a first or second order process model we can easily find PI- and PID-controller settings, respectively. The rules has one tuning parameter, the closed-loop time constant τ_c, which can be used to trade off between performance ("tight" control) and robustness ("smooth" control).

In a previous paper, we studied the optimal PI-controller on the same first order process (1), where we compared the SIMC-tuned PI-controller with the "optimal" PI-controller (Grimholt and Skogestad, 2012).

The SIMC rules do not cover the tuning of PID-controllers (τ_D) for first-order processes. In this work, we propose an extension of SIMC, and we find that adding $\tau_D = \theta/3$ gives a close-to optimal controller.

This paper is structured as follows. In Section 2 the performance/robustness trade-off is quantified. The optimization problem is defined in Section 3. Optimal PI- and PID-controllers are presented in Section 4, and the extended SIMC-rule is presented and analysed in Section 5.

2. EVALUATION OF PERFORMANCE AND ROBUSTNESS

2.1 Performance

Performance (objective 1) is related to the difference between the measurement $y(t)$ and the setpoint y_s (Figure 1), and can be quantified in several different ways. For a step load change we might quantify performance in term of rise time, overshoot, and settling time. To quantify performance in terms of a single scalar, we choose the integrated absolute error:

$$\mathrm{IAE} = \int_0^\infty |y(t) - y_s(t)| \, dt \tag{4}$$

Actually, in this paper we do not consider setpointchanges (y_s). That is, y_s may be considered constant. Hoverer, we do consider output disturbances d_o, which for the so-called one-degree of freedom controller in Figure 1 (where there is no filter on y_s) is equivalent to a setpoint change.

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

$$J(K) = 0.5 \left[\frac{\mathrm{IAE}_{do}(K)}{\mathrm{IAE}^\circ_{do}} + \frac{\mathrm{IAE}_{di}(K)}{\mathrm{IAE}^\circ_{di}} \right] \tag{5}$$

The weighting factors IAE°_{di} and IAE°_{do} are for a reference controller, which for the given process is the IAE-optimal PID-controller for a step load change on input and output, respectively. Note that two different controllers are used to obtain the reference IAE-values, whereas a single controller K is used to find $\mathrm{IAE}_{di}(K)$ and $\mathrm{IAE}_{do}(K)$ when evaluating the IAE-cost $J(K)$.

To ensure robust reference controllers, they are required to have $M_S = 1.59$ [1] , and the resulting weighting factors are given for four processes in Table 1. We could have used a truly IAE-optimal controller as reference (i.e. without the restrictions on M_S). However, this would not change the results much because the IAE-value is quite close to its optimum value at $M_S = 1.59$.

It may be argued that a two-degree of freedom controller with a setpoint filter can be used to enhance setpoint performance, and thus we only need to consider input disturbances. But note that although a step load change on the output d_o, as mentioned, is equivalent to a setpoint step-change y_s for the setup in Figure 1, it is not affected by the setpoint filter. In summary, we consider disturbance rejection which, can only be handled by the feedback controller K (Figure 1). The optimal controller will depend on the specific disturbance model, and we chose to consider disturbances at the plant output (d_o) and plant input (d_i). To get a good balance, we weigh the both equally as given in (5).

2.2 Robustness

Robustness (objective 2) can, like performance, be quantified in several ways. Some common measurements used are the sensitivity peak (M_S), complementarity sensitivity peak (M_T), gain margin (GM), phase margin (PM), and allowable time delay error ($\Delta\theta/\theta$). In this paper we have choose to quantify robustness with the peak of the sensitivity function M_S. The sensitivity function is defined as,

$$S = \frac{1}{1 + GK} \tag{6}$$

and the M_s-value is defined as its peak, or mathematically

$$M_S = \max_\omega |S(j\omega)| = \|S\|_\infty \tag{7}$$

where $\|\cdot\|_\infty$ is the H_∞-norm. 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 GK. For robustness, a small value of M_S is desired, and generally M_S should not exceed 2. A typical "good" value is about 1.6, and notice that $M_S < 1.6$ guarantees GM > 2.67 and PM $> 36.4°$ (Rivera et al., 1986).

3. PROBLEM FORMULATION

For a given first order plus time delay process, the Pareto-optimal curve for PI or PID control is generated by solving the the following optimization problem:

> Given the set of M_S-values $\mathcal{M}_S = \{1.1, \ldots, 3\}$, solve for all $m \in \mathcal{M}_S$:
>
> $$\min_K \quad J(K) = 0.5 \left[\frac{\mathrm{IAE}_{do}(K)}{\mathrm{IAE}^\circ_{do}} + \frac{\mathrm{IAE}_{di}(K)}{\mathrm{IAE}^\circ_{di}} \right]$$

[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 the process $G = \frac{e^{-s}}{s+1}$.

Table 1. Optimal PID-controllers ($M_s = 1.59$) and corresponding IAE-values for four processes.

	Output disturbance				Input disturbance				Optimal combined (minimize J)						
Process	K_c	τ_I	τ_D	IAE°_{do}	K_c	τ_I	τ_D	IAE°_{di}	K_c	τ_I	τ_D	IAE_{do}	IAE_{di}	J	M_s
e^{-s}	0.20	0.32	0	1.61	0.20	0.32	0	1.61	0.20	0.32	0	1.61	1.61	1	1.59
$\frac{e^{-s}}{s+1}$	0.43	0.62	0.62	1.56	0.42	0.59	0.59	1.46	0.42	0.61	0.61	1.57	1.47	1.01	1.59
$\frac{e^{-s}}{8s+1}$	4.97	8.00	0.32	1.61	3.75	1.56	0.59	0.58	4.35	2.52	0.48	2.35	0.63	1.27	1.59
$\frac{e^{-s}}{s}$	0.62	∞	0.32	1.61	0.51	2.33	0.53	6.37	0.54	3.24	0.48	3.02	6.81	1.47	1.59

IAE_{do} and IAE_{di} are for a unit step load change on output (y) and input (u), respectively.

$$\text{s.t.} \quad M_S = m$$

where $K(s)$ is a PI- or a PID-controller.

4. OPTIMAL PID-CONTROL ON FIRST ORDER PLUS TIME DELAY PROCESS

Four different first-order processes have been investigated,

- Pure time delay ($\tau_1/\theta = 0$)
- Small time constant ($\tau_1/\theta = 1$)
- Intermediate time constant ($\tau_1/\theta = 8$)
- Integrating ($\tau_1/\theta = \infty$)

Strictly speaking, the pure time delay process is a zeroth-order plus time delay process. Because a pure time delay process is a limiting case of a first-order process, with $\tau_1 = 0$, we will for clarity of the discussion also refer to this as a first order plus time delay process.

In Table 1, the resulting optimal PID-controllers and J-values are given for $M_S = 1.59$ for four processes; We note that $J = 1$ for a time delay process, because there is no trade-off between disturbances for this process, and because the reference controllers have $M_S = 1.59$. For the other cases we have $J > 1$ because there is a trade-off between input and output disturbances rejection. For example, for the integrating process, the optimal value of J is 1.47, mainly because we have to sacrifice output disturbance rejection.

4.1 Pareto-optimal controllers

A Pareto-optimal curve depicts the trade-off between two conflicting objectives. In our case, this is the trade-off between performance (J) and robustness (M_s). The Pareto-optimal curves for PI- and PID-control for the four processes are shown in Figure 3. Notice that we have only a real trade-off when there is a negative slope between the variables (left side of the plots). Here we have to select a compromise between the two objectives. That is, if we improve one objective, the other deteriorates. We never want to be in a region with zero or positive slope (right side of the plots), because we can both improve robustness and performance by just moving to the left. Therefore, the minimum point in the cure represent the largest M_S value we would like to use. The deterioration in performance at large M_S-values is cased by oscillating response which increases the IAE.

For a pure time delay process there is no advantage to add derivative action, and it is optimal to use simple PI-control (Figure 3, top right). As the time constant increases the benefit of using derivative action increases. For integrating processes, using derivative action improves performance a substantial 42% at $M_S = 1.59$, compared to optimal PI-control.

4.2 Optimal PID tuning parameters

The optimal tuning can be divided into to main regions: delay dominant ($\tau_1/\theta < 5$) and lag dominant ($\tau_1/\theta > 5$). For the lag dominant processes, the scaled controller

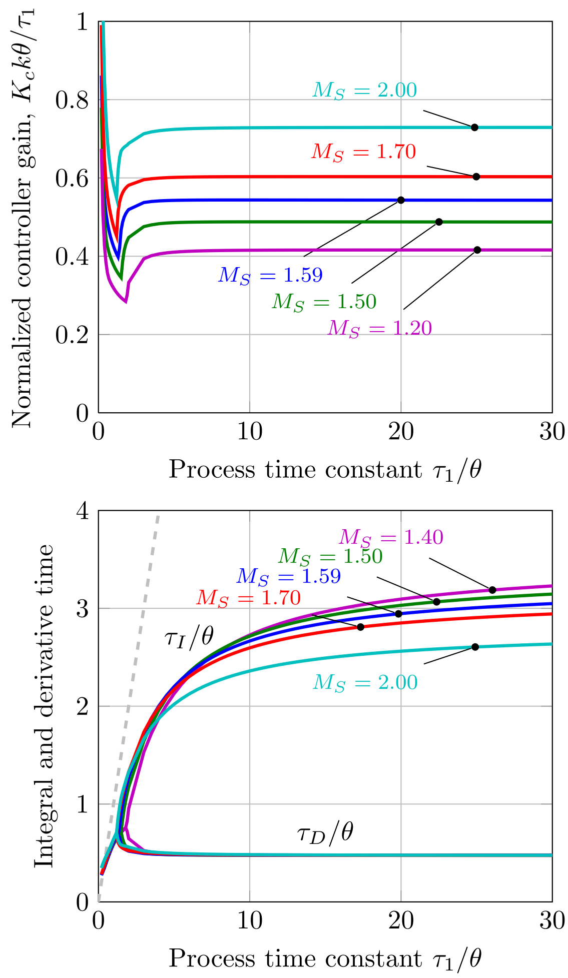

Fig. 2. Pareto-optimal PID settings for five given M_S-values (robustness) for the process $G(s) = \frac{ke^{-\theta s}}{\tau_1 s+1}$. For reference $\tau_I = \tau_1$ is also plotted (dashed line).

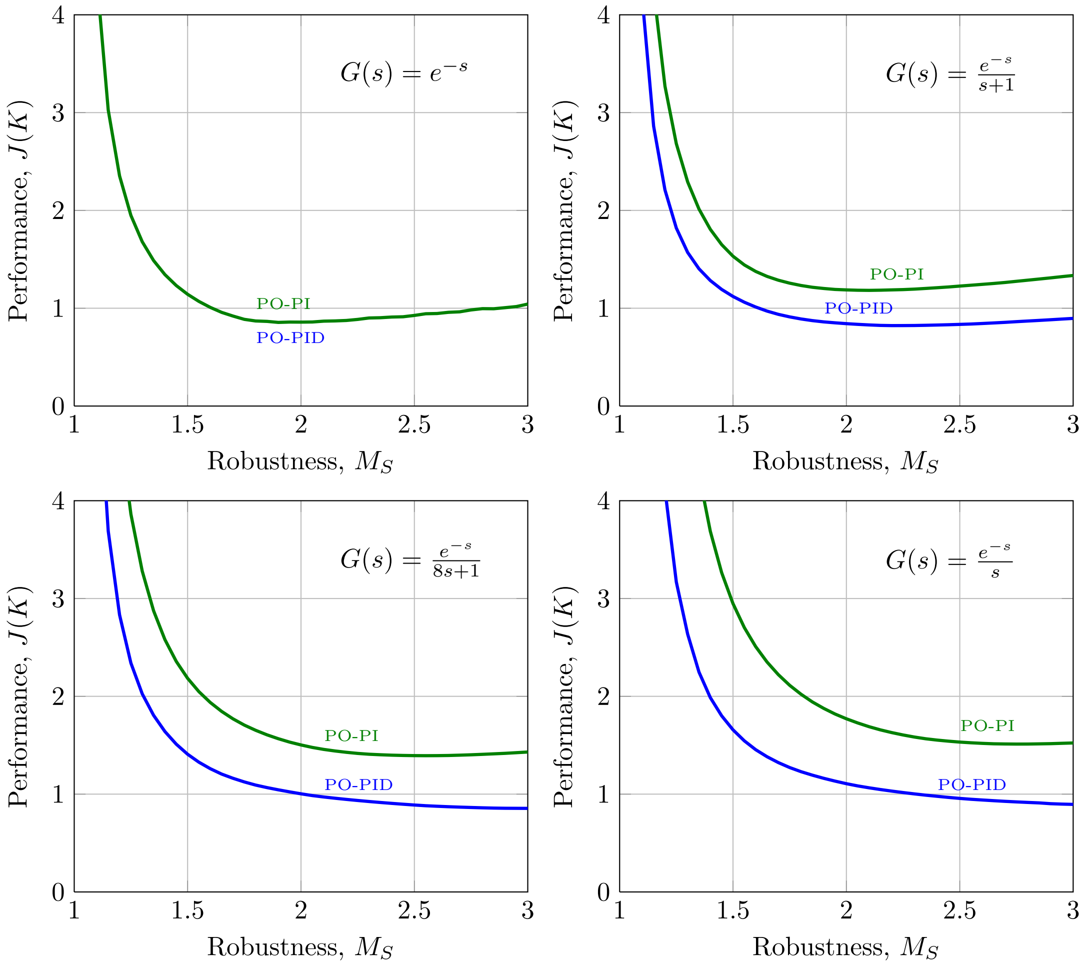

Fig. 3. Pareto-optimal trade-off between robustness (M_S) and performance (J) for Pareto-optimal PI- and PID-control for four processes

gain approaches the constant vale $K_c k\theta/\tau_1 = 0.54$ for $M_S = 1.59$ (Figure 2, top). The same can be observed for integral time and derivative time which approaches to $\tau_I\theta = 3.24$ and $\tau_D\theta = 0.48$ at the same M_S-value (Figure 2, bottom). Though, the integral time converges slower. For increasing M_S-values (lower robustness), the optimal controller gain increases and the optimal integral time decreases. Interestingly, the optimal derivative time seems to be independent to the selected robustness, resulting in $\tau_D\theta = 0.48$.

The delay dominated region can be subdivided into two additional regions based on the controller: Equal controller zeros ($\tau_I = \tau_D$). from approximately τ_1/θ smaller than 2, and two distinctive controllers zeros, from approximately τ_1/θ larger than 2. Setting the derivative time equal to the integral time concurs with the recommendation of Ziegler and Nichols (1942). However, this is only for a small range of first-order processes. In the upper part of the delay dominated region the integral time is close to the process time constant (indicated by dashed line) which is in agreement with the well-known IMC rule (Rivera et al., 1986).

4.3 Parallel (ideal) vs. cascade PID-controller

So far we have found the optimal cascade PID-controller in (2). A more general PID-controller is the parallel, or ideal, PID-controller,

$$K_{\text{parallel}} = K_c' \left(1 + \frac{1}{\tau_I' s} + \tau_D' s\right) \tag{8}$$

The cascade controller can always be translated into the parallel form by

$$K_c' = K_c f, \quad \tau_I' = \tau_I f, \quad \tau_D' = \tau_D / f \tag{9}$$

where $f = 1 + \tau_D/\tau_I$. The more general parallel form (8) can not be translated to the cascade form (2) if it has complex zeros.

The difference between the two forms are minor in our case. For three of the processes the cascade form is optimal, as the optimal parallel form has two real zeros. Only the small time constant process ($\tau_1/\theta = 1$) had optimal parallel PID-controller with complex zeros, as seen from Table 2. The optimal cascade controller (2) for this process is on the boarder between real an complex with to coinciding real zeros, $\tau_I = \tau_D$. This compares to $\tau_I' = 4\tau_D'$ for the parallel form controller (8). However, as seen from Table 4 and Figure 4, the difference between the cascade and the parallel controller very small even for this process.

Table 2. Optimal PID-controllers for $G(s) = \frac{e^{-s}}{s+1}$ for different M_s values.

Controller	Optimal parallel settings				
	K_c'	τ_I'	τ_D'	$J(K)$	M_s
$K^\star_{\text{cascade}}$	0.37	1.20	0.30	2.21	1.2
	1.16	1.37	0.32	0.84	2.0
	1.40	1.30	0.33	0.83	2.5
$K^\star_{\text{parallel}}$	0.37	1.01	0.44	1.89	1.2
	1.16	1.25	0.34	0.84	2.0
	1.37	1.12	0.40	0.81	2.5

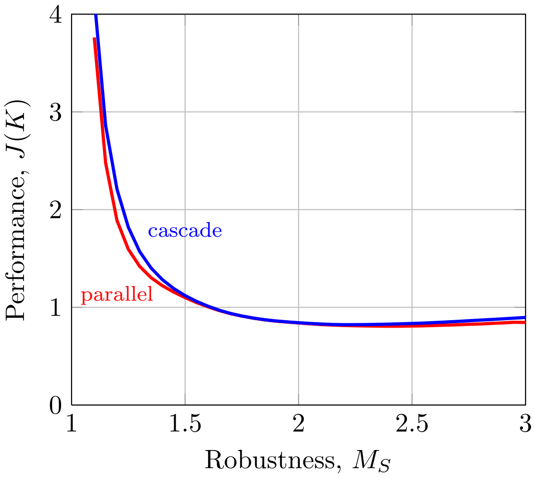

Fig. 4. Pareto-optimal cascade PID-control (blue line) and parallel PID-control (red line) on $G(s) = \frac{e^{-s}}{s+1}$.

4.4 Complimentary sensitivity peak M_T

In addition to the sensitivity peak, M_S, it is interesting to look at the complimentary sensitivity peak, M_T, which is defined as the resonance peak of the closed loop transfer function T,

$$T = \frac{GK}{1+GK}, \quad M_T = \max_\omega |T(j\omega)| = \|T\|_\infty$$

In robustness terms M_T represents the sensitivity to modelling errors of zeros and time delay. Since $y = -Tn$, it also gives the worst case noise amplification on the outputs. We know that M_T is bounded by the selected M_S-value (Skogestad and Postlethwaite, 2005),

$$M_T \leq M_S + 1$$

However, this bound can be rather very loose. For stable systems, the M_T-value is usually smaller than the M_S-value. For unstable systems, it is usually opposite. We usually want $M_T \leq 1.6$, which guarantees GM > 1.62 and PM $> 36.4°$

The Pareto-optimal controllers generally gave good M_T-values, which in most cases are lower than the M_S-values (Figure 6). Compared with PI, the PID-controllers gave higher M_T-values for smaller values of M_S. However, since the controller is anyway robust in this region, this deterioration in M_T is not significant.

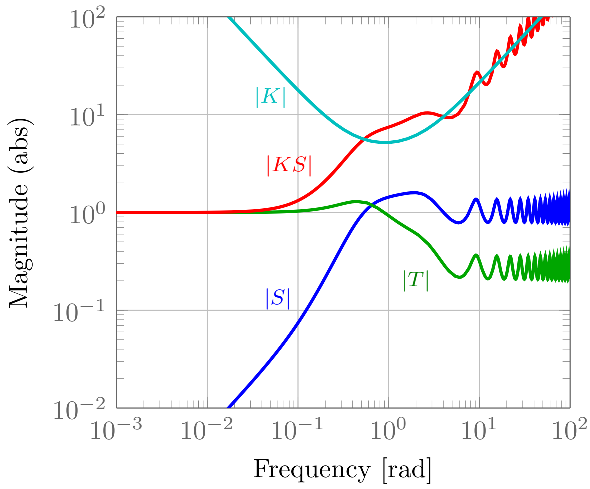

Fig. 5. Typical frequency plot of S, T, KS, and K. Example is for process $G(s) = \frac{e^{-s}}{8s+1}$ and PID-controller $K(s) = 4.35\left(\frac{2.52s+1}{2.52s}\right)(0.48s+1)$.

4.5 Input usage

Input usage is an important aspect for control. From Figure 1 we have

$$u = -Td_i + KS(d_o + n)$$

Thus, input usage is decided by the two transfer functions: T (from input disturbance) and KS (from output disturbance and noise), see Figure 5 for a typical example frequency plot. The input disturbances are not a problem because T is bound by M_T which is low for our cases.

KS has a peak at the intermediate frequencies which is approximately $|KS(j\omega)| \approx K_c M_S$ (Åström and Hägglund, 2006). Thus, with a given M_S-values, the optimal PID-controllers, which has a higher controller gain K_C, requires more input usage than the optimal PI-controller.

The product $K_c\tau_D$ is good indication for input usage in the high frequency range, where $|KS(j\omega)| \approx K_c\tau_D\omega$. For PID-controllers without a measurement filter (τ_F), the $|KS|$ peak goes to infinity, $\|KS\|_\infty = \infty$. Therefore, it is important to filter out the high frequency noise, and the resulting peak will depend heavily on the selected filter. It is important that the selected filter do not influence controller performance and robustness in a significant way. If so, we have a PIDF-controller where also the filter constant should be considered a degree of freedom in the optimization problem. For this reason we recommend that the filter constant should be selected no larger than $\tau_D/3$.

5. EXTENDED SIMC FOR PID-CONTROL OF FIRST-ORDER PROCESSES WITH TIME DELAY

The SIMC PI-settings for the first-order plus delay process (1) are

$$K_c = \frac{1}{k}\frac{\tau_1}{\tau_c + \theta}, \quad \tau_I = \min\{\tau_1,\ 4(\tau_c + \theta)\} \qquad (10)$$

where the desired first-order *closed-loop* time constant τ_c is the only tuning parameter. For a "fast and robust" setting, $\tau_c = \theta$ is recommended.

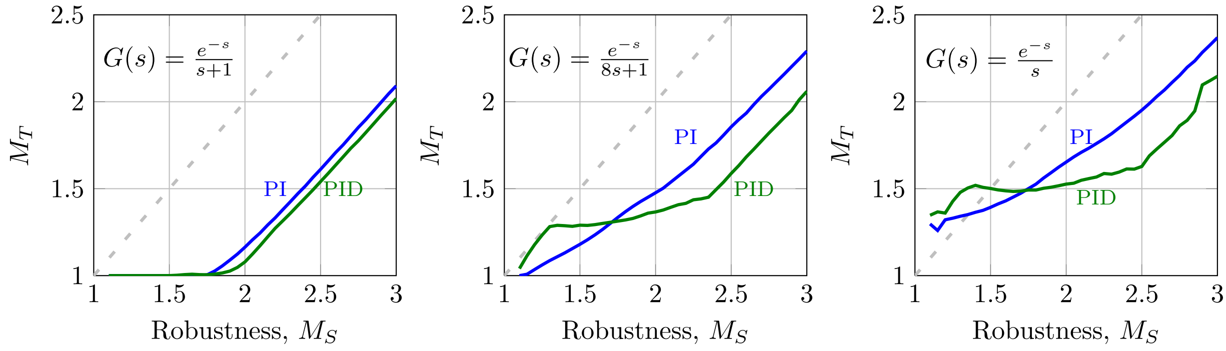

Fig. 6. M_T versus M_S for Pareto-optimal PI- and PID-controllers.

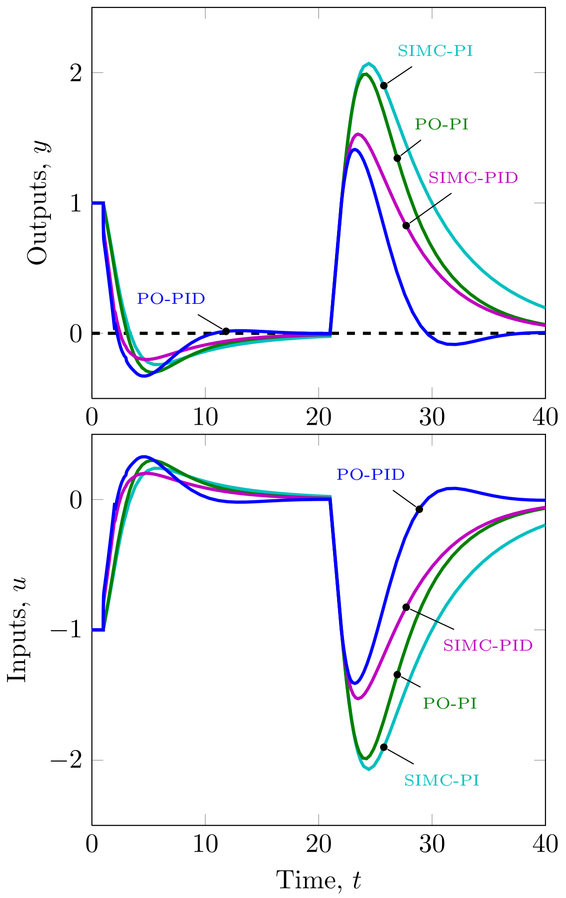

Fig. 7. Time response for optimal and SIMC PI- and PID-controllers ($M_S = 1.59$) for an input disturbance (at time 0) and an output disturbances (at time 20), on the integrating process $G(s) = e^{-s}/s$. To get a proper system, a PID-controller with measurement filter (3) is used with $\tau_F = 0.01$.

The trade-off curve for the SIMC controllers was generated by varying the tuning parameter τ_c from a large to a small value. The controllers corresponding to the three specific choices

- $\tau_c = 1.5\theta$ (smooth tuning)
- $\tau_c = \theta$ (default value)
- $\tau_c = 0.5\theta$ (more aggressive tuning)

are shown by circles. Except for the pure time delay process, the differences in performance (J) between SIMC-PI and optimal-PI are within 10%, which shows that the SIMC PI-rules are close to optimal (Figure 8). In other words, by adjusting τ_c we can generate the optimal controller for a given desired robustness (Grimholt and Skogestad, 2012).

When considering PID-control, is commonly proposed to introduce derivative action to improve performance for processes with time delay, e.g. $\tau_D = 0.5\theta$ (Rivera et al., 1986). Based on analytical derivations and simulations, Skogestad (2003) found that adding $\tau_D = 0.5\theta$ only marginally improved performance for load input disturbances compared with PI. However, he also noted that introducing derivative action improved the robustness margins somewhat. Because of the small improvements, increased complexity and increased noise sensitivity, Skogestad recommend to *not* use derivative action to counteract time delay for first-order plus delay processes.

This is somewhat conflicting to what we have found so far (Figure 2) where performance substantially improved by introducing derivative action (Figure 3).

We have found that setting the derivative time to:

$$\tau_D = \theta/3 \tag{11}$$

with the SIMC-rules gives good performance, as is discussed in more detail bellow. This value $\tau_D = \theta/3$ follows from our previous results (Grimholt and Skogestad, 2012) where we derived an "improved" SIMC PI-rule.

As Skogestad claimed, when comparing SIMC-PID with SIMC-PI when choosing $\tau_c = \theta$, the robustness is somewhat improved, but performance is only marginally improved (middle circle, Figure 8). For the integrating process, the M_S is improved from 1.70 for PI to 1.46 for PID, but there is only a 6% increase in performance. However, due to this added robustness for PID, we can reduce τ_c and significantly improve performance for a given M_S-value (36% increase for the integrating process) compared to the original PI-controller. A good value for the tuning constant would be $\tau_c = 0.5\theta$, as it give approximately the same robustness as the SIMC PI-rules with $\tau_c = \theta$.

Compared with the optimal PID-controller, the SIMC PID-controller have higher input usage in the intermediate frequency range and less input usage in the high frequency range, as can be seen from the K_c and $K_c\tau_D$ in Table 3.

The SIMC-rules settles slower than the optimal controller for both input and output disturbances (Figure 7). How-

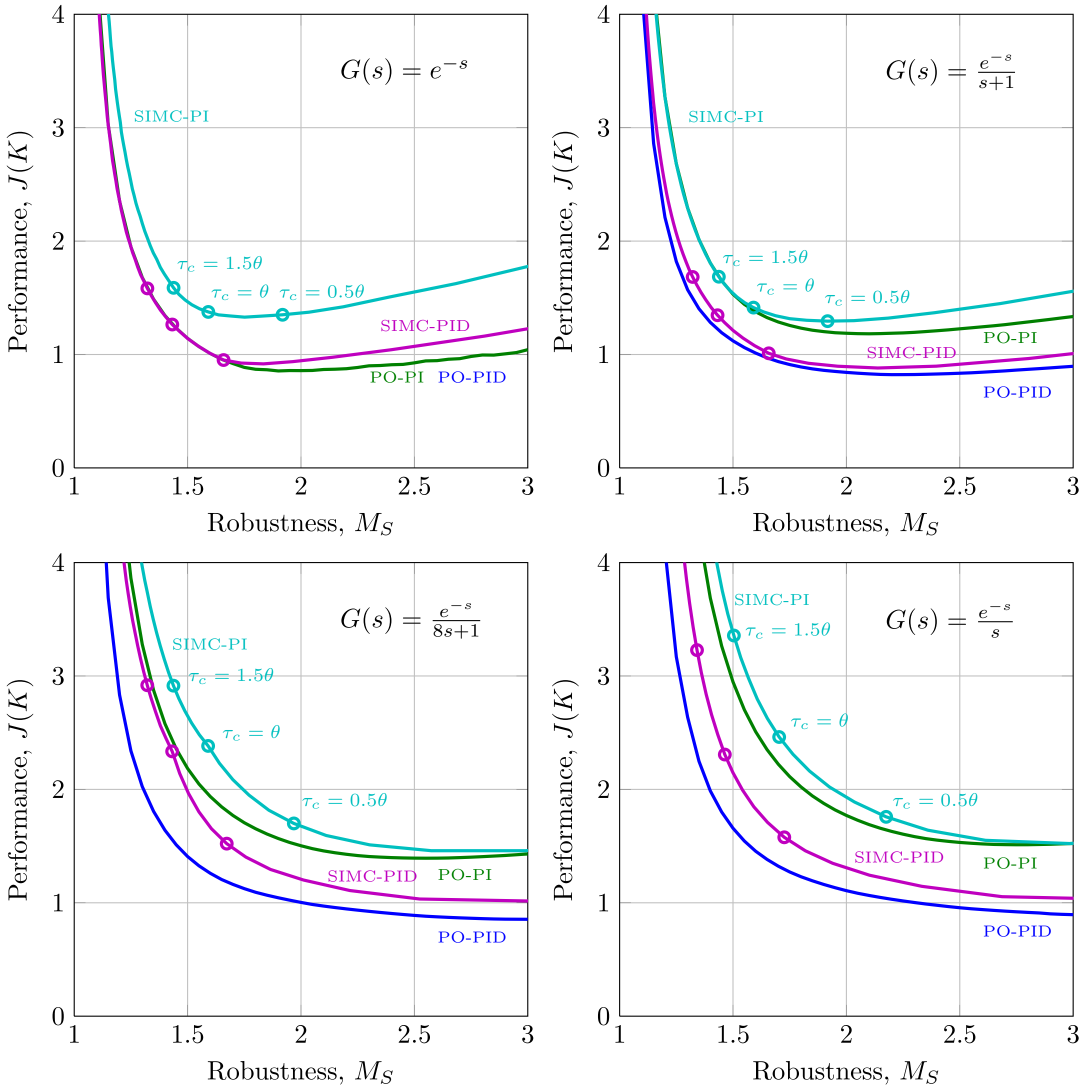

Fig. 8. Pareto-optimal trade-off between robustness (M_S) and performance (J) for optimal and SIMC PI- and PID-control for four processe. SIMC-PI and SIMC-PID have K_c and τ_I given by (10) (but the value of τ_c are not the the same for a given M_S), and SIMC-PID have $\tau_D = \theta/3$.

Table 3. Tuning for optimal and SIMC PID-controllers with $M_s = 1.59$ on four processes.

	Optimal PID						SIMC PID						
Process	K_c	τ_I	τ_D	$K_c\tau_D$	J	M_S	K_c	τ_I	τ_D	$K_c\tau_D$	J	τ_c	M_S
e^{-s}	0.20	0.32	0	–	1.00	1.59	0	0	$0.33^\star$	–	1.02	0.62	1.59
$\frac{e^{-s}}{s+1}$	0.43	0.61	0.61	0.26	1.01	1.59	0.62	1	0.33	0.21	1.08	0.61	1.59
$\frac{e^{-s}}{8s+1}$	4.39	2.54	0.48	2.11	1.26	1.59	5.92	6.5	0.33	1.64	1.69	0.63	1.59
$\frac{e^{-s}}{s}$	0.55	3.25	0.48	0.26	1.45	1.59	0.59	6.81	0.33	0.20	1.86	0.70	1.59

($\star$) Gives a ID-controller with $I = K_c/\tau_I = 1/(\tau_c + \theta)$. This is the same as a PI-controller.

ever, it is usually the maximum deviation that is of main concern in the industry. The SIMC-rule have roughly equal peak deviation for input disturbance, and a smaller peak deviation for output disturbances compared with the optimal. By using SIMC-PID the peak deviation is reduced by 26% for input disturbances, compared with SIMC-PI.

REFERENCES

Åström, K. and Hägglund, T. (2006). *Advanced PID Control*. ISA.

Grimholt, C. and Skogestad, S. (2012). Optimal PI-Control and Verification of the SIMC Tuning Rule. In *IFAC conference on Advances in PID control (PID'12)*, 2.

Rivera, D., Moriari, M., and Skogestad, S. (1986). Internal Model Control. 4. PID Controller Design. 252–265.

Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning . 13, 291–309.

Skogestad, S. and Postlethwaite, I. (2005). *Multivariable Feedback Control – Analysis and Design*. John Wiley & Sons, Ltd, 2 edition.

Ziegler, J.G. and Nichols, N.B. (1942). Optimum settings for automatic controllers. *Trans. ASME*, 64, 759–768.