IFAC Conference on Advances in PID Control
PID'12
Brescia (Italy), March 28-30, 2012

Optimal PI-Control and Verification of the SIMC Tuning Rule

Chriss Grimholt **Sigurd Skogestad***

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

Abstract: Optimal PI-settings are derived for first-order with delay processes for specified levels of robustness (M_s-value) and compared with the simple SIMC-rule. Optimality (performance) is defined in terms of the integrated absolute error (IAE) of the output for combined step changes in setpoints and input disturbances. With SIMC, the robustness level is adjusted by changing the tuning parameter τ_c, and the SIMC-rule was found to give surprisingly good setting with almost Pareto-optimal performance. The exception is a pure time delay processes where the SIMC-rule gives a pure integral controller with somewhat sluggish response. A simple modification to improve on this, is to increase the time constant in the rule by one third of the time delay.

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

1. INTRODUCTION

In this paper, we consider a PI-controller

$$c(s) = K_c \cdot \left(1 + \frac{1}{\tau_I s}\right) \tag{1}$$

where K_c is the controller gain and τ_I the integral time, applied to a first order with time delay process

$$g(s) = \frac{k}{\tau_1 s + 1} \cdot e^{-\theta s} \tag{2}$$

where k is the process gain, τ_1 is the process time constant and θ is the time delay.

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 the PI- and PID-controller setting, respectively. For a first-order process,

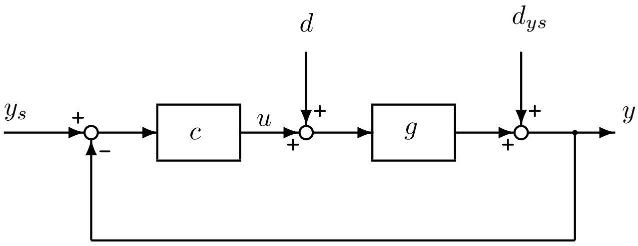


Fig. 1. Block diagram of feedback control system. We may treat an output disturbance (d_{ys}) as a special case of setpoint change (y_s)

the SIMC PI-settings are

$$K_c = \frac{1}{k}\frac{\tau_1}{\tau_c + \theta} \tag{3}$$

$$\tau_I = \min\{\tau_1, 4(\tau_c + \theta)\} \tag{4}$$

The SIMC PI-rule has one tuning parameter τ_c which can be used to trade off between performance ("tight" control) and robustness ("smooth" control). The rule was originally derived mainly with simplicity in mind, and although it was mentioned in the original paper that these are "probably the best simple PID tuning in the world", one may want to ask the questions: How good is the SIMC PI-rule? Can it be improved? Is the recommended value $\tau_c = \theta$ a good default value for the tuning parameter τ_c?

To study this, we need to compare the SIMC PI performance with the "optimal" PI-controller for a process on the from (2). However, "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 fortunately the trade-off space has only one main dimension, 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

The idea is then to compare the SIMC controller, with τ_c as a parameter, with the "Pareto-optimal" PI-controller. Pareto-optimality applies to multiobjective problems, and means that no further improvement can be made in objective 1 without sacrificing objective 2. We still need to define more exactly objectives 1 and 2.

Towards the end the paper, we consider how to find a first-order with delay model, which is required for the SIMC tuning rule. It can be difficult to obtain open-loop responses for model approximation and Shamsuzzoha and Skogestad (2010) presented a new method for using closed-loop data for tuning. We will modify this method to obtain a first-order with time delay model.

2. EVALUATION OF PERFORMANCE AND ROBUSTNESS

2.1 Output performance

Output performance (objective 1) is related to the difference between the measurement $y(t)$ and its setpoint y_s (Figure 1), and may be quantified in many different ways. For example, for a setpoint change, we might consider the rise time, overshoot and settling time. However, to quantify the output performance in terms of a single scalar, we choose to use the integrated absolute error:

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

The IAE-value depends on which disturbance or setpoint change we consider. We choose to consider a weighted average of IAE for a step change in the load disturbance d (IAE_d) and setpoint y_s (IAE_{y_s}):

$$J(c) = 0.5 \left[\frac{\mathrm{IAE}_{ys}(c)}{\mathrm{IAE}_{ys}^\circ} + \frac{\mathrm{IAE}_d(c)}{\mathrm{IAE}_d^\circ} \right] \tag{5}$$

Importantly, the weighted cost J is independent of the process gain k, the disturbance and setpoint magnitudes, and of the unit used for time.

The weighting factors IAE_{ys}° and IAE_d° are for reference PI-controllers, which for the given process are IAE-optimal for a setpoint change and a disturbance, respectively. To get robust reference controllers, they are required to have $M_s = 1.59$. Values for IAE_{ys}° and IAE_d°, are given for four different processes in Table 1. We could have used the truly optimal IAE-value as the reference (without the restriction $M_s = 1.59$), but this would not have changed the results much because the IAE-value is anyway quite close to its minimum at $M_s = 1.59$. Note that two different optimal PI-controllers are used to obtain the two reference values, whereas a single controller c is used to find $\mathrm{IAE}_{ys}(c)$ and $\mathrm{IAE}_d(c)$ when evaluating the weighted IAE-cost $J(c)$.

It may be argued that a two-degree of freedom controller with a setpoint filter may be used to improve the response for setpoints, but note that a setpoint change is equivalent to an output disturbance (d_{ys} in Figure 1) which is not affected by the filter. Thus, both setpoint changes (output disturbances) and input disturbances should be included when evaluating performance, and to get a good balance between the two, we weigh them about equally by using the cost in (5).

2.2 Robustness, input usage and noise sensitivity

There are many ways to quantify objective 2 related to the benefits of low gain. Robustness my be quantified in terms of sensitivity peak (M_s), complementarity sensitivity peak (M_t), gain margin (GM), phase margin (PM), allowable time delay error ($\Delta\theta/\theta$), etc. Input usage may also be quantified in many ways, but we have found total variation (TV) to be a good measure:

$$\mathrm{TV} = \int_0^\infty |\frac{du}{dt}| dt = \sum_0^\infty |u_i - u_{i-1}| \tag{6}$$

The noise sensitivity on the outputs (y) may be quantified by the complimentary sensitivity peak (M_t). However, even more important is usually the noise sensitivity on the inputs (u), which for a PI-controller is closely related to the magnitude of the controller gain, K_c, which should be low to have small noise sensitivity. This issue is not considered in this paper, partly because the noise frequency and magnitude shows large variation from case to case.

In this paper, we choose to quantify all the issues related to objective 2 in terms of the sensitivity peak, M_s:

$$M_s = \max_\omega \left| \frac{1}{1 + gc(j\omega)} \right|$$

which is the peak of the sensitivity function in the frequency domain. In robustness terms, M_s is the inverse of the closest distance between the critical point -1 and the loop transfer function gc in the Nyquist plot. 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 $> 36.4^o$ (Rivera *et al.*, 1986). As shown later, the M_s-value is also closely related to TV, at least for process and controllers used in this paper.

3. OPTIMAL PI TUNING

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

For a given set of M_s values $\mathcal{M}_s = \{1.1, ..., 1.59, ..., 3\}$, solve:

$$\min_c J(c) = 0.5 \left[\frac{\mathrm{IAE}_{ys}(c)}{\mathrm{IAE}_{ys}^\circ} + \frac{\mathrm{IAE}_d(c)}{\mathrm{IAE}_d^\circ} \right]$$
$$\text{s.t.} \quad M_s = m$$

for all $m \in \mathcal{M}_s$.

In Table 1, the resulting optimal PI-controllers and J-values are given for $M_s = 1.59$ for four processes;

- pure time delay ($\tau_1/\theta = 0$),
- small time constant ($\tau_1/\theta = 1$),
- intermediate time constant ($\tau_1/\theta = 8$),
- integrating ($\tau_1/\theta = \infty$).

We note that $J = 1$ for a time delay process, because there is no trade-off between disturbances and setpoints for this process, and because the reference controllers have $M_s = 1.59$. For the integrating process, the optimal value of J is 1.50, mainly because we have to sacrifice setpoint performance.

3.1 Optimal trade-off between robustness and performance

Figure 2 shows the optimal output response (left) and corresponding input usage (right) for three values of M_s

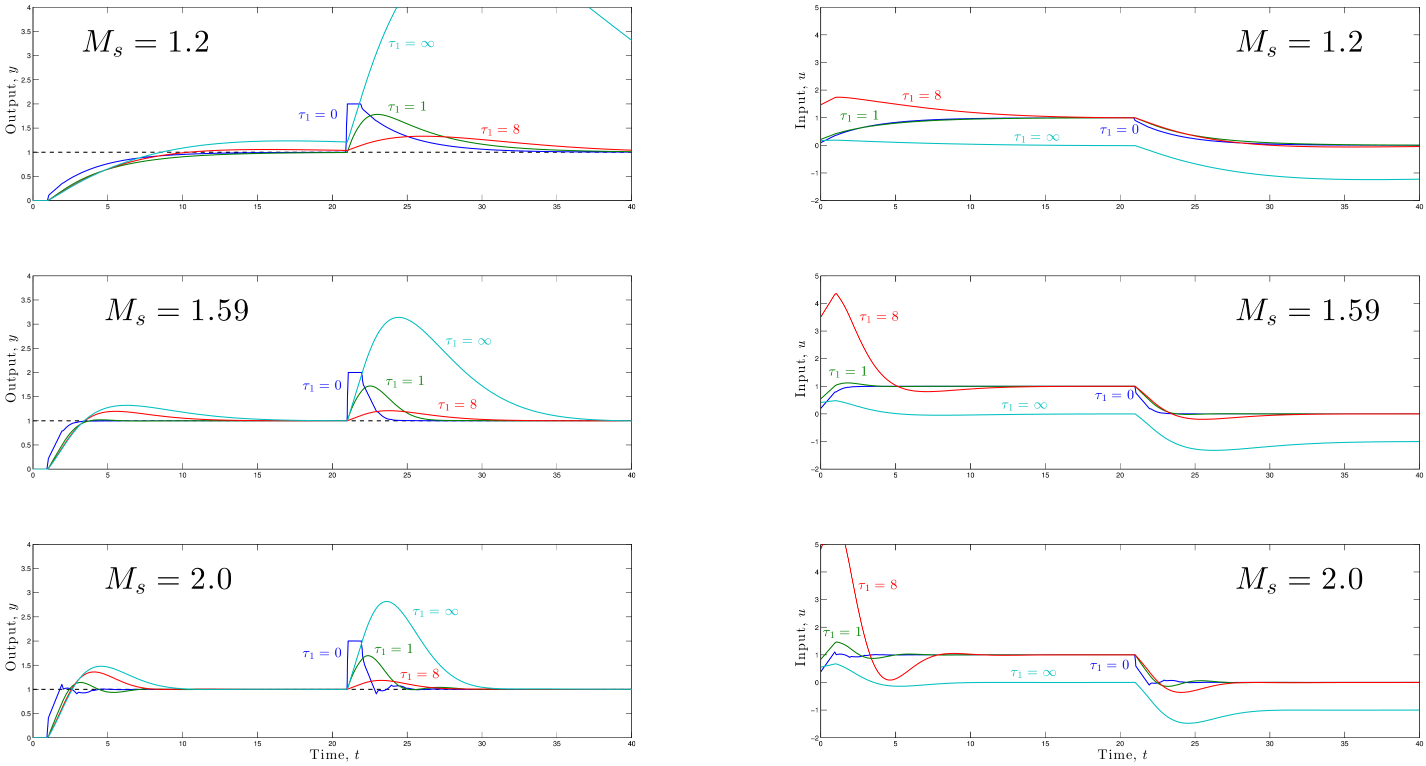


Fig. 2. Optimal PI control of first-order plus delay processes: Output response (left) and input usage (right) for $M_s = 1.20$ (top), $M_s = 1.59$ (middle) and $M_s = 2.00$ (bottom). Setpoint change at $t = 0$ and (input) disturbance at $t = 20$.

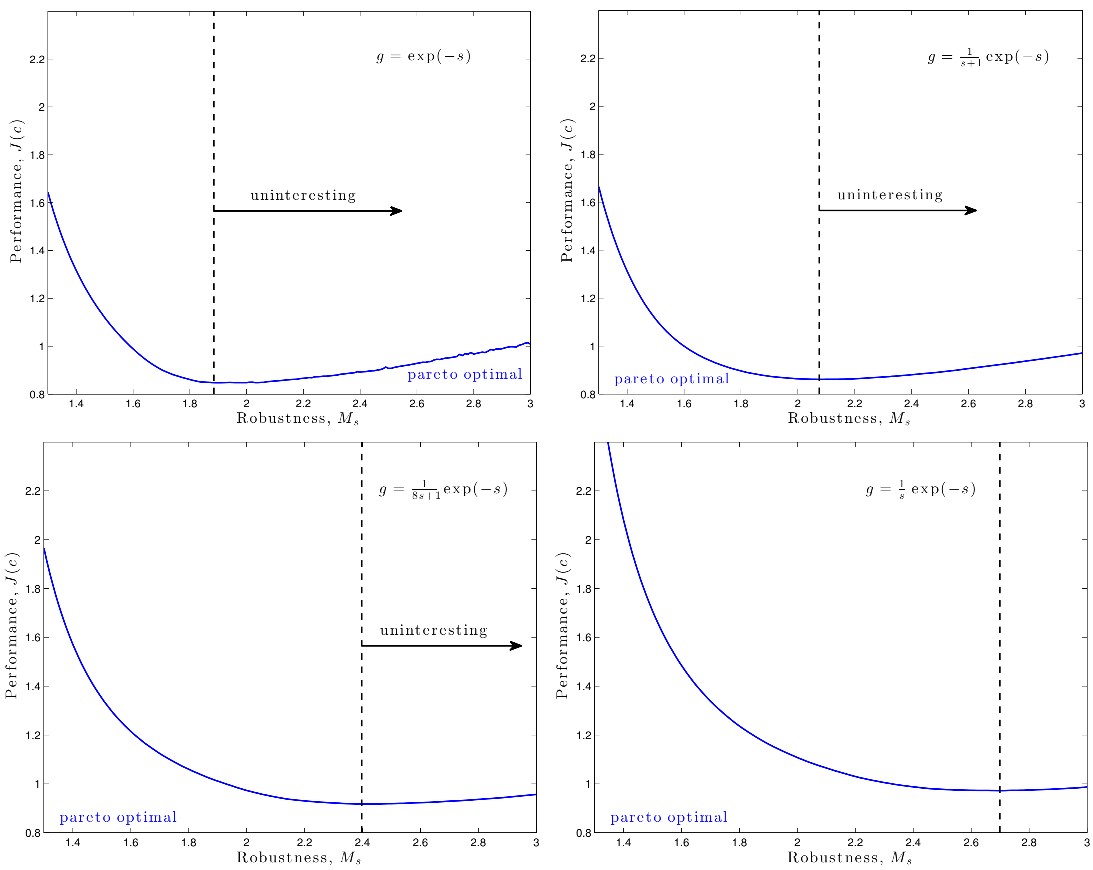


Fig. 3. Pareto-optimal trade-off between robustness (M_s) and performance (J) with PI control for four processes

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

	Setpoint			Input disturbance			Optimal combined (minimize J)					
Process	K_c	τ_I	$\mathrm{IAE}^{\circ}_{ys}$	K_c	τ_I	IAE°_{d}	K_c	τ_I	IAE_{ys}	IAE_d	J	M_s
e^{-s}	0.20	0.32	1.609	0.20	0.32	1.609	0.20	0.32	1.607	1.607	1	1.59
$\frac{e^{-s}}{s+1}$	0.54	1.10	2.073	0.50	1.0	2.016	0.54	1.10	2.087	2.038	1.00	1.59
$\frac{e^{-s}}{8s+1}$	4.0	8	2.171	3.34	3.7	1.134	3.46	4.0	3.096	1.164	1.23	1.59
$\frac{e^{-s}}{s}$	0.50	∞	2.174	0.40	5.8	15.09	0.41	6.3	4.318	15.38	1.50	1.59

IAE_{ys} is for a unit setpoint change. IAE_d is for a unit input disturbance.

(1.2, 1.59, and 2) for the four processes. We see that as M_s is increased and control gets more aggressive, the output performance is improved whereas the input usage gets worse.

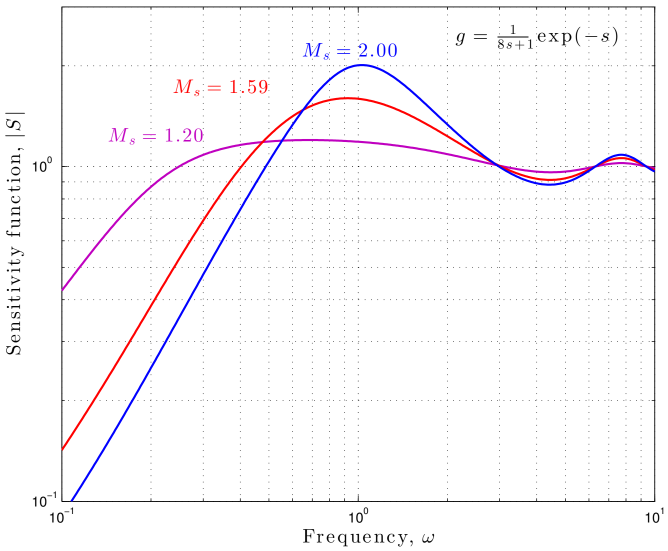


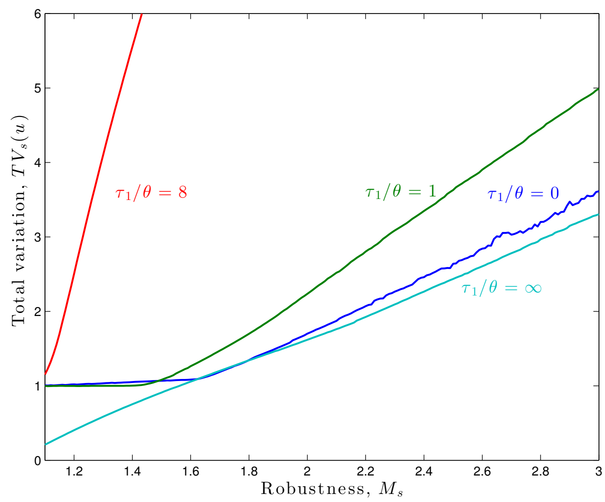
Fig. 4. Optimal PI control of first-order plus delay process: Magnitude of sensitivity function for three M_s-values

Figure 4 shows the corresponding optimal sensitivity function, $S = 1/(1 + gc)$, for the process $g(s) = \frac{1}{8s+1}\exp(-s)$ for the same three M_s-values. We see how tightening the control (increasing M_s) gives better performance (lower $|S|$) at lower frequencies, but worse performance at higher frequencies. The worst-case peak for $|S|$ is the M_s-value which occurs approximately at the frequency corresponding to the time delay, $\omega = 1/\theta = 1$ (the unit depends on the unit used for time).

The main results are given in Figure 3 which gives the Pareto-optimal trade-off between performance (J) and robustness (M_s) (blue curve) for the four processes. Note that we have a real trade-off between performance (J) and robustness (M_s) only when there is a negative slope between these variables, which is in the left region in the figures in Figure 3. Here, we have to compromise between performance and robustness. We never want to be in the region with a zero or positive slope (marked as "uninteresting"), because if we move to the right in this region, both performance and robustness deteriorate at the same time. The minimum point on the curve (dashed black line) represents the maximum M_s-value that we would want to use. This maximum M_s-value varies from about 1.9 for the delay-dominant process to about 2.7 for the integrating process. M_s-values higher than this give oscillating responses which increases the IAE-value.

3.2 Correlation between robustness and input usage

As mentioned earlier, we chose to not consider the input usage when finding the optimal controller. One reason is that this is not necessary, because the input usage in terms of total variation (TV) correlates very well with M_s. This is shown in Figure 5 for the four processes. We see that when M_s increases, TV increases for both setpoint change



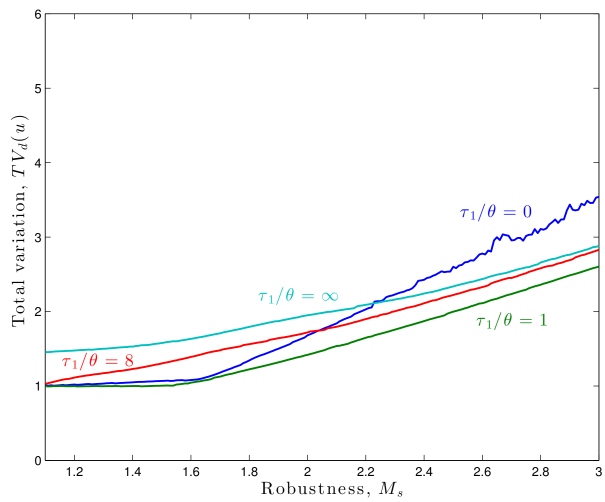


Fig. 5. Optimal PI control of first-order plus delay process: Input usage (TV) for setpoint change (upper) and input disturbance (lower) as a function of M_s.

and input disturbance. Because the system is subjected to unit step changes in setpoint and input disturbance, the minimum TV-value is 1, except for the integrating process $\tau_1/\theta = \infty$ where the minimum is 0. Also note the TV-value is relatively constant around 1 at low M_s-values. It is only when M_s is so large that the system starts oscillating, that TV starts increasing. This illustrates that M_s is better to use as a robustness measure (objective 2) than TV.

3.3 Optimal PI tuning parameters

The optimal PI controller settings are plotted in Figures 7 and 8 as a function of the scaled process time constant τ_1/θ for five values of M_s. The SIMC-settings for the choice $\tau_c = \theta$ are included as a reference (dashed line). The are two main regions: delay dominant ($\tau_1/\theta < 5$) and lag dominant ($\tau_1/\theta > 5$). For lag-dominant processes (Figure 4), the scaled controller gain $K_c k\theta/\tau_1$ approaches a constant value as we increase the lag τ_1/θ. For example, for $M_s = 1.59$, the optimal value is $K_c k\theta/\tau_1 = 0.4145$ for $\tau_1/\theta = 50$, and 0.4100 for $\tau_1/\theta = \infty$ (integrating process). The same can be observed for the integral time, which approaches $\tau_I = 6.3\theta$ for an integrating process. Note that when M_s increases (less robustness), the controller gain increases and the integral time decrease.

The delay-dominant region ($\tau_1/\theta < 5$) is shown more clearly in Figure 8. Interestingly, the optimal integral time is almost independent of the robustness (M_s) in this region. We note that $\tau_I \approx \tau_1$ (dashed line) for small values of τ_1/θ; more exactly for τ_1/θ between about 1 and 3. This agrees with the well-known IMC rule (Rivera *et al.*, 1986). However, τ_I does not approach zero (integrating controller) as τ_1/θ becomes small. Rather, for a pure time delay processes ($\tau_1/\theta = 0$), the integral time is approximately $\theta/3$ for all M_s-values of interest (which from Figure 3 is $M_s < 1.9$). On the other hand, the controller gain (K_c) increases with M_s also for small values of τ_1/θ).

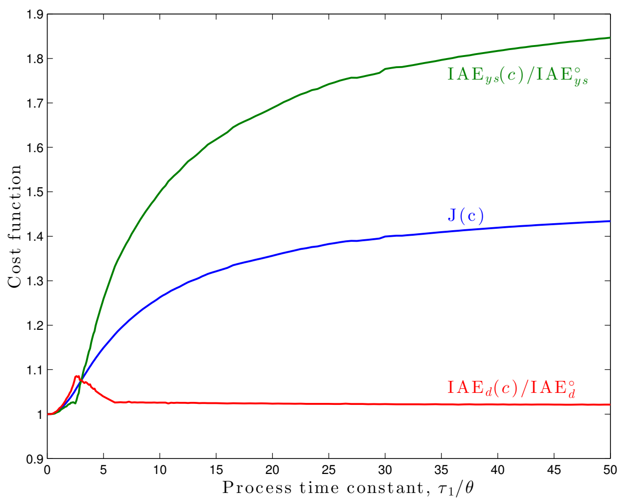


Fig. 6. Optimal PI control of first-order plus delay process: Setpoint and disturbance performance as a function of process time constant τ_1/θ ($M_s = 1.59$).

3.4 Output disturbance (setpoint) versus input disturbance trade-off

As already noted from Table 1, the optimal controller that minimizes the cost $J(c)$ in (5), puts more emphasis on the (input) disturbance rejection (IAE_d) than on the setpoint tracking (IAE_{ys}), especially for larger values of τ_1/θ. This is further illustrated in Figure 6, where we see that for τ_1/θ less than about 5, the optimal controller puts about equal weight on setpoints and disturbances, whereas for larger values it mostly emphasizes disturbances. For example, for an integrating process, we find $\mathrm{IAE}_d/\mathrm{IAE}^\circ_d = 1.02$ (close to optimal for input disturbance), whereas $\mathrm{IAE}_{ys}/\mathrm{IAE}^\circ_{ys} = 1.99$ (twice the optimal). This is a bit surprising since we have equal weights on the two terms in the cost function. The reason is that the IAE-optimal controller for setpoint (IAE°_{ys}) has no integral action ($\tau_I = \infty$), whereas integral action is required to bring the process back to its setpoint in reasonable time when there are input disturbances, and this consideration dominates when want to find a single optimal controller. Even if we increase the emphasis on setpoints, we find a similar pattern. For example, if we use a cost function with only a small weight (1%) on input disturbances,

$$J(c) = \left[0.99\frac{\mathrm{IAE}_{ys}(c)}{\mathrm{IAE}^\circ_{ys}} + 0.01\frac{\mathrm{IAE}_{d}(c)}{\mathrm{IAE}^\circ_{d}}\right]$$

we find for an integrating process the optimal settings $K_c = 0.467$ and $\tau_I = 13.3\theta$, with $\mathrm{IAE}_d/\mathrm{IAE}^\circ_d = 1.83$ and $\mathrm{IAE}_{ys}/\mathrm{IAE}^\circ_{ys} = 1.92$. We notice that there is only a minor improvement in setpoint performance ($\mathrm{IAE}_y s$ decreases about 8%), whereas disturbance rejection is much worse (IAE_d increases about 79%). The conclusion is this that we may put emphasis mainly on input disturbances when tuning PI controllers.

4. EVALUATION OF THE SIMC PI TUNING RULE

The SIMC PI-settings (Skogestad, 2003) for a first-order with delay model are

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

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. The corresponding SIMC tuning parameters for $\tau_c = \theta$ are given for the four processes in Table 2.

4.1 Effect of the tuning parameter τ_c on robustness

By adjusting the tuning parameter τ_c, we can shift the trade-off between performance and robustness. In particular, M_s decreases (more robust) as we increase τ_c. In Figure 10, we show for four processes, the effect of changing τ_c/θ in the region from 0 to 3, on (1) M_s-value, (2) gain margin, (3) phase margin and (4) allowed time delay error. We note that there is a direct correlation between τ_c and all four robustness measures, and the correlation does not depend strongly on the value of τ_1/θ. In particular, this applies to the gain margin (GM), which shows an almost one-to-one (and linear) relationship with

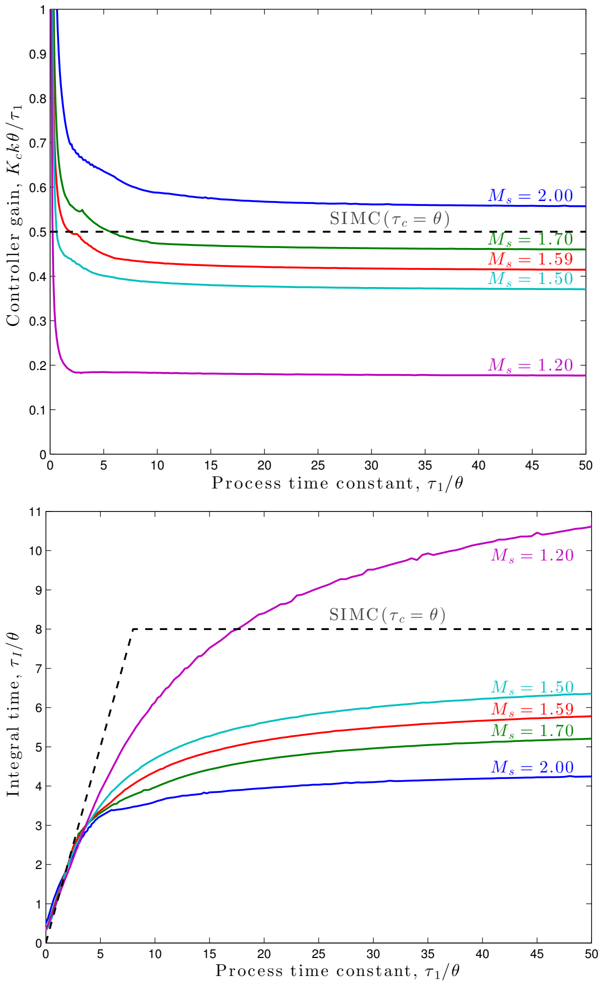


Fig. 7. Optimal PI-settings (K_c, τ_I) for first-order with delay processes as a function of τ_1/θ for five values of M_s.

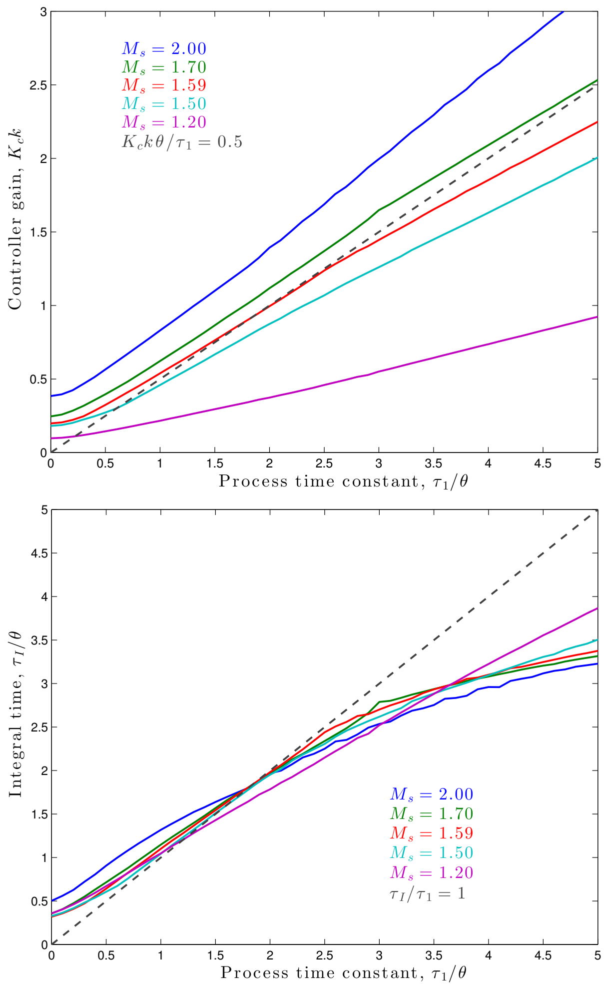


Fig. 8. Optimal PI-settings (K_c, τ_I) for first-order with delay processes for small values of τ_1/θ

Table 2. SIMC PI-controllers $(\tau_c = \theta)$ and corresponding J- and M_s-values for four processes.

	SIMC PI $(\tau_c = \theta)$						Improved SIMC PI $(\tau_c = \theta)$					
Process	K_c	τ_I	IAE$_{ys}$	IAE$_d$	J	M_s	K_c	τ_I	IAE$_{ys}$	IAE$_d$	J	M_s
e^{-s}	0	0 $^{(*)}$	2.17	2.17	1.35	1.59	0.17	0.33	1.95	1.95	1.21	1.45
$\frac{e^{-s}}{s+1}$	0.5	1	2.17	2.04	1.15	1.59	0.67	1.33	1.99	1.99	1.09	1.69
$\frac{e^{-s}}{8s+1}$	4	8	2.17	2.00	1.38	1.59	4.17	8	2.14	1.92	1.34	1.62
$\frac{e^{-s}}{s}$	0.5	8	3.92	16	1.43	1.70	0.5	8	3.92	16	1.43	1.70

$^{(*)}$ Pure integral controller with $K_I = K_c/\tau_I = 0.5$.

τ_c. Thus, for the SIMC-rule, specifying τ_c is essentially the same as specifying GM. This linear relationship can be proved mathematically for processes with relatively small values of τ_1, such that the SIMC-rule gives $\tau_I = \tau_1$ (see Appendix)

$$GM = \frac{\pi}{2}\left(\frac{\tau_c}{\theta} + 1\right) \tag{7}$$

and Figure 10 shows that this relationship holds well also for an integrating processes (where the SIMC-rule gives $\tau_I = 4(\tau_c + \theta)$).

For relatively small values of τ_1, such that $\tau_I = \tau_1$, a similar linear relationship also exists for the allowable time delay error (see Appendix):

$$\frac{\Delta\theta}{\theta} = \frac{\pi}{2}\left(\frac{\tau_c}{\theta} + 1\right) - 1 = GM - 1 \tag{8}$$

However, Figure 10 shows that this relationship does not quite hold for an integrating process where the allowable time delay error is somewhat smaller.

4.2 Optimality of the SIMC-rule

Figure 9 compares the Pareto-optimal performance (blue curve) with the SIMC PI-controller (green curve) for the four processes. The curve for the SIMC controller was generated by varying the tuning parameter τ_c from a large to a small value. The controllers corresponding to

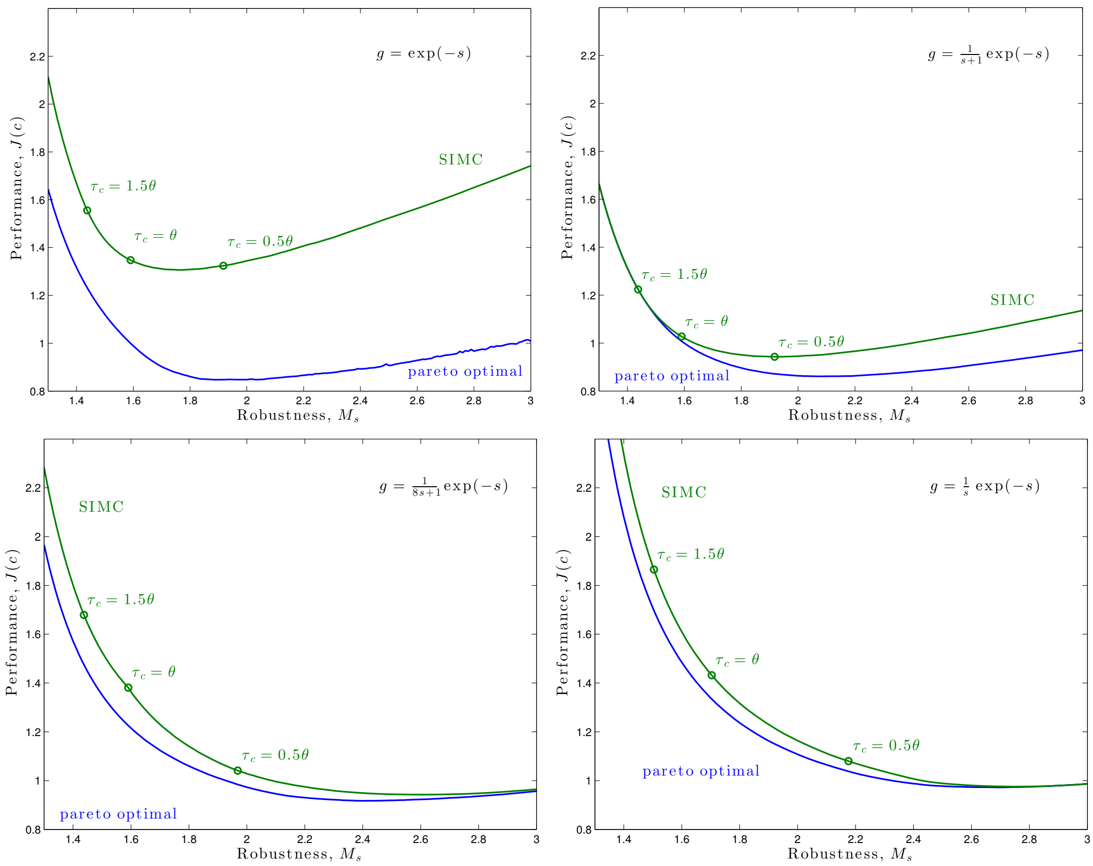


Fig. 9. Evaluation of optimality of SIMC PI tuning rule for four processes.

the choices $\tau_c = 1.5\theta$ (smooth tuning), $\tau_c = \theta$ (default value) and $\tau_c = 0.5\theta$ (more aggressive tuning) are shown by circles. Except for the pure time delay process, the differences between the IAE-values (J) for SIMC (green curve) and optimal (blue curve) are within 10%, which shows that the SIMC PI-rules are close to optimal. In other words, by adjusting τ_c we can generate the optimal controller for a given desired robustness.

Another important observation from Figure 9 is that the SIMC default recommendation $\tau_c = \theta$ for "tight" control (as given by middle of the three circles) in all cases is located in the desired trade-off region with a negative slope, well before we reach the minimum. Also, the recommended choice gives a fairly constant M_s-value in the region 1.59 to 1.7. From this we conclude that, except for the 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).

4.3 Evaluation of SIMC-rule for integral time

When comparing the optimal PI-settings with the SIMC-rule for an integrating process, it seems that the integral time given by the SIMC-rule is a bit too large (Figure 7, bottom). Specifically, for an integrating process and $\tau_c = \theta$ ($M_s = 1.70$), the SIMC-rule gives $\tau_I/\theta = 8$ whereas the optimal value for $M_s = 1.70$ is about 5.5. This indicates that in terms of the trade-off between setpoint and disturbance performance, the SIMC-rule is putting somewhat too much emphasis on setpoints. There has also been claims that the SIMC-rule results in sluggish disturbance rejection for integrating processes (Haugen, 2010; Di Ruscio, 2010). To be able to shift the trade-off between setpoint and disturbance performance, one may introduce an extra parameter in the tuning rule (Alcantara *et al.*, 2010). Haugen (2010) suggested to introduce an extra servo/regulator trade-off parameter c in the expression for the integral time,

$$\tau_I = \min(\tau_1, c(\tau_c + \theta)) \tag{9}$$

where $c = 4$ gives the original SIMC-rule. However, introducing an extra parameter adds complexity and from Figure 9 the potential benefit does not seem sufficiently large. Nevertheless, one may consider choosing another (lower) fixed value for c, and Haugen (2010) suggests using $c = 2$ to improve the input disturbance performance. In Figure 11 we check the optimality of choosing $c = 2$ (dashed pink curve) for an integrating process. We find indeed that performance (J) is improved compared to SIMC ($c = 4$) if we keep $\tau_c = \theta$. However, robustness is worse (with M_s close to 2, whereas SIMC gives 1.7). More importantly, by decreasing τ_c for SIMC so that we get the same robustness, the performance with SIMC ($c = 4$) is even better than with $c = 2$. In fact, we see that the SIMC performance curve (green) is closer to the Pareto-optimal curve in the entire optimal performance region (where J is small). In summary, we find that the value $c = 4$ in the original SIMC-rule provides a well-balanced servo/regulator trade-off, and to improve disturbance performance for an integrating process we

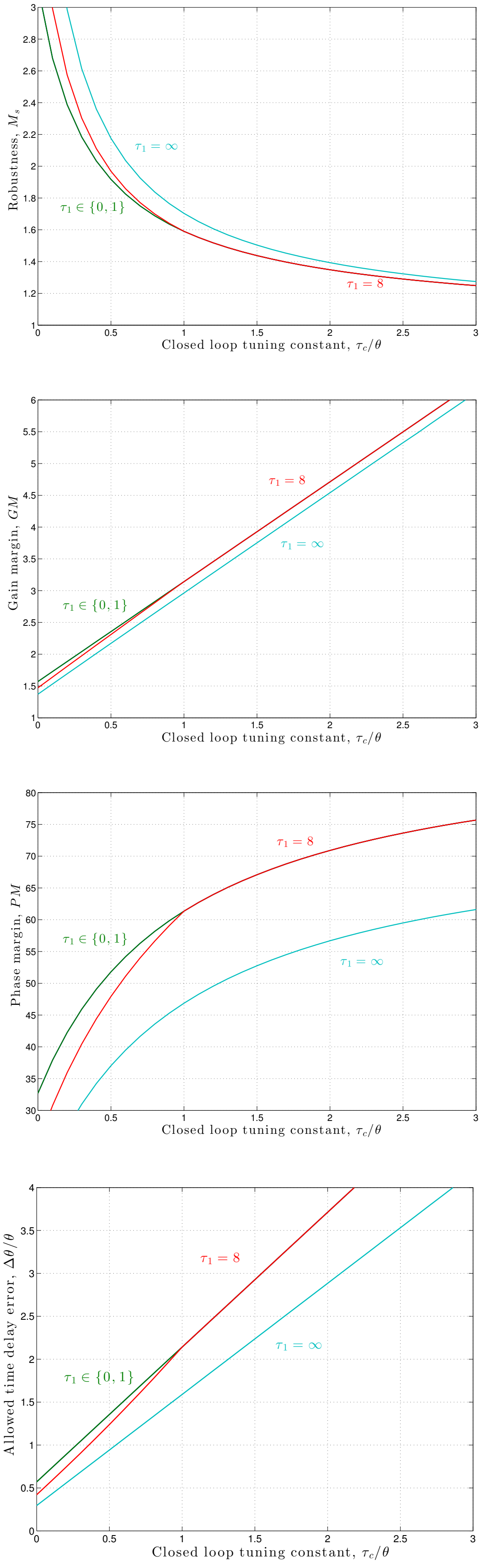


Fig. 10. SIMC: Effect of tuning parameter τ_c/θ on robustness parameters: (1) M_s, (2) GM, (3) PM and (4) allowable time delay error $\Delta\theta/\theta$

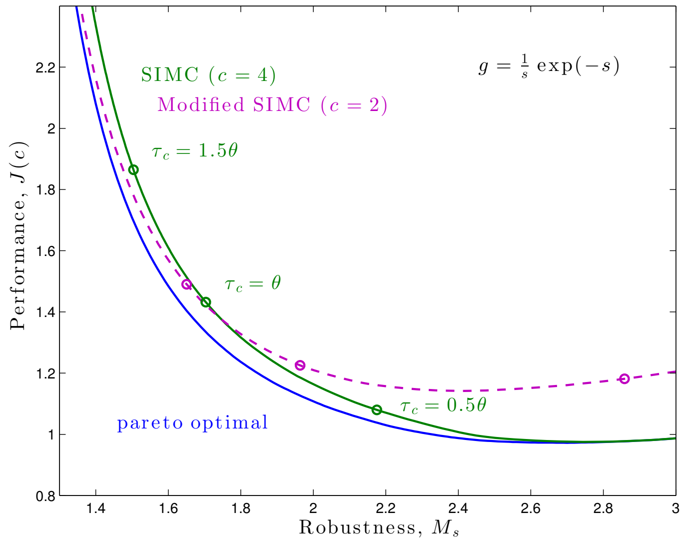


Fig. 11. Evaluation of suggested modified SIMC-rule ($c = 2$) for integrating process.

recommended decreasing the tuning constant τ_c/θ, say to around 0.7, rather than changing the value of c.

5. IMPROVED SIMC-RULE

From the results in Figure 9, the main "problem" with the SIMC-rule is for pure time delay processes, where we see that the IAE-value (J) for the SIMC controller is about 40% higher than the minimum with the same robustness (M_s). This is further illustrated by the closed-loop simulations in Figure 13. We see that the response with the SIMC PI-controller (denoted SIMC-original in the figure), although being nice and smooth, is somewhat sluggish initially, because it is actually a pure I-controller with $K_c = 0, \tau_I = 0$ and $K_I = K_c/\tau_I = 0.5$. On the other hand, the IAE-optimal PI-controller (with minimum J for $M_s = 1.59$) has K_c about 0.2 and τ_I about 0.32 (and $K_I = 0.62$).

Fortunately, as already observed, the optimal PI-controller for a pure time delay process (green line in Figure 3), has an almost fixed integral time of approximately $\theta/3$ for all values of M_s between 1.4 and 1.7. (Figure 8, bottom) Based on this fact, we propose a simple change to the SIMC-rule, namely to replace τ_1 by $\tau_1 + \theta/3$ in the rule (PI control), which markedly improved the responses for a pure time delay process. It is important that the change is simple because "simplicity" was one of the main objectives when originally deriving the SIMC-rule.

A similar change, but with $\theta/2$ rather than $\theta/3$, was originally proposed by Rivera *et al.* (1986) for their "improved PI" tuning rule, and the effectiveness of this modification is also clear from the paper of Foley *et al.* (2005). However, as seen in Figure 13, the response with this IMC PI controller also settles rather slowly towards the setpoint, indicating that the integral time $\theta/2$ is too large. The conclusion is that we recommend to replace τ_1 by $\tau_1 + \theta/3$ in the SIMC-rule to get the improved SIMC-rule:

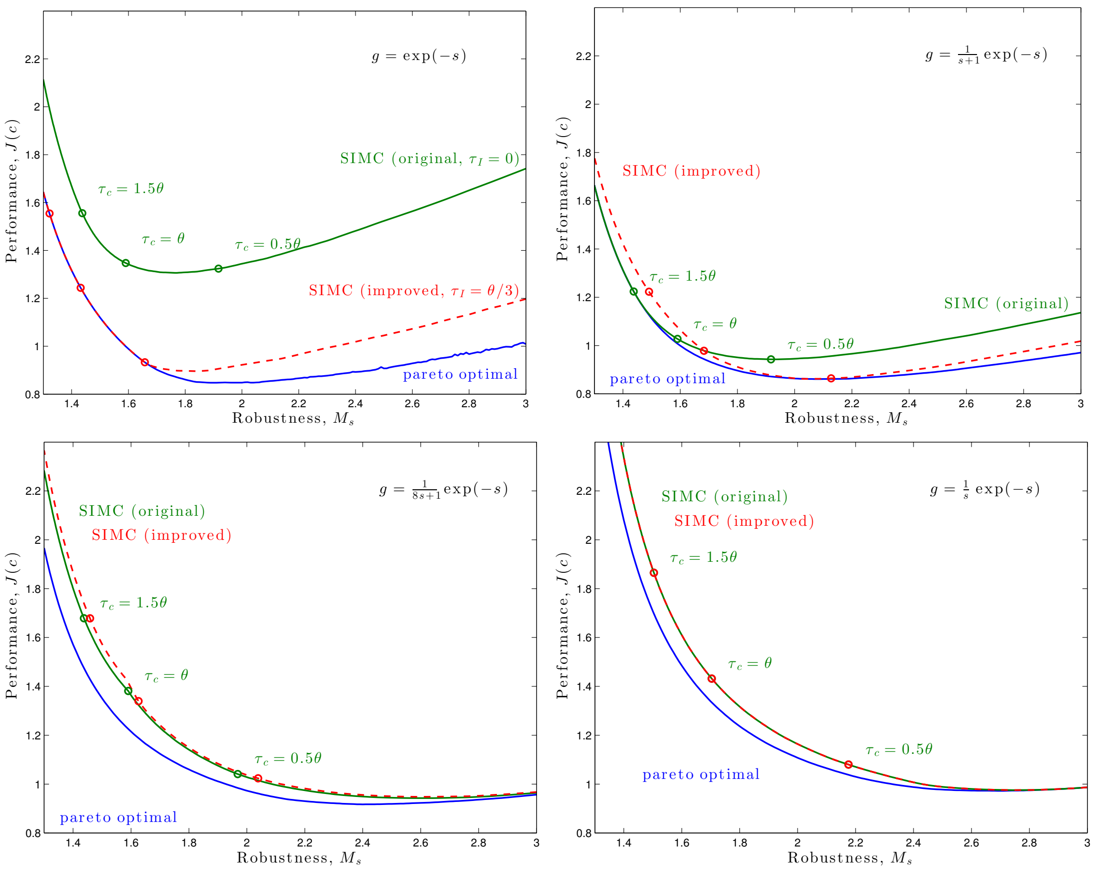


Fig. 12. Evaluation of optimality of improved SIMC PI tuning rule for four processes.

$$K_c = \frac{1}{k}\frac{\tau_1 + \frac{\theta}{3}}{\tau_c + \theta} \tag{10}$$

$$\tau_I = \min\{\tau_1 + \frac{\theta}{3}, 4(\tau_c + \theta)\} \tag{11}$$

The improvement of this rule for a pure time delay processes is clear from the red curves in Figures 12 (upper left) and 13; for small M_s-values the improved SIMC-controller is almost identical to the Pareto-optimal, which confirms that $\tau_I = \theta/3$ is close to optimal for a pure time delay process. One disadvantage with the improved rule, which is not considered in this analysis, is that the larger controller gain (K_c) makes the input and output responses more sensitive to measurement noise.

For the process with a small time constant ($\tau_1/\theta = 1$), the improved SIMC-rule (red curve in upper right plot in Figure 12) is slightly better than the "original" SIMC-rule (green curve) for higher M_s-values (where we get better performance) but slightly worse for lower M_s-values. For the two processes with a large time constant ($\tau_1/\theta = 8$ and $\tau_1/\theta = \infty$) there is, as expected, almost no difference between the original and improved SIMC-rules.

6. OBTAINING THE MODEL FROM A CLOSED-LOOP SETPOINT EXPERIMENT

In some cases, open-loop responses may be difficult to obtain, and using closed-loop data may be more effective. The most famous closed-loop experiment is the Ziegler-Nichols where the system is brought to sustained oscilla-

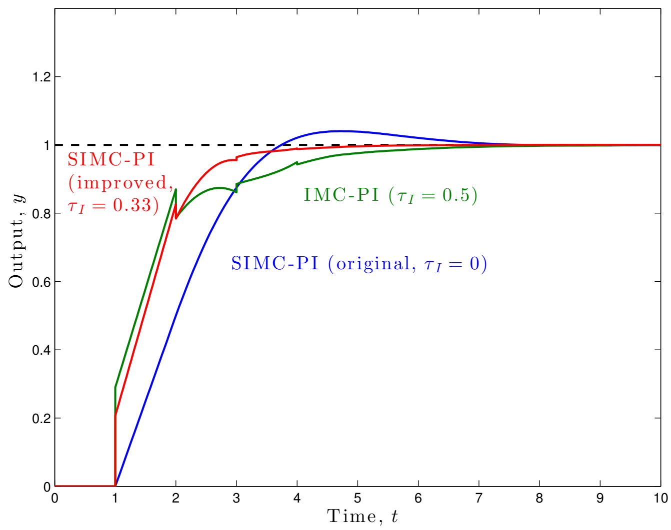


Fig. 13. Closed-loop setpoint responses for pure time delay process ($\theta = 1, k = 1, \tau_1 = 0$) with PI-control.
- IMC PI: $K_c = 0.29$, $\tau_I = 0.5$ ($K_I = K_c/\tau_I = 0.58$).
- SIMC PI original ($\tau_c = \theta$): $K_c = 0, \tau_I = 0 (K_I = 0.5)$.
- SIMC PI improved ($\tau_c = 0.61\theta$) : $K_c = 0.207, \tau_I = 0.333 (K_I = 0.62)$. All three controllers have the same robustness ($M_s = 1.59$). For a pure time delay process, the setpoint and disturbance responses are identical, and the input and output are identical.

tions by use of a P-only controller. One disadvantage with the method is that the system is brought to its instability

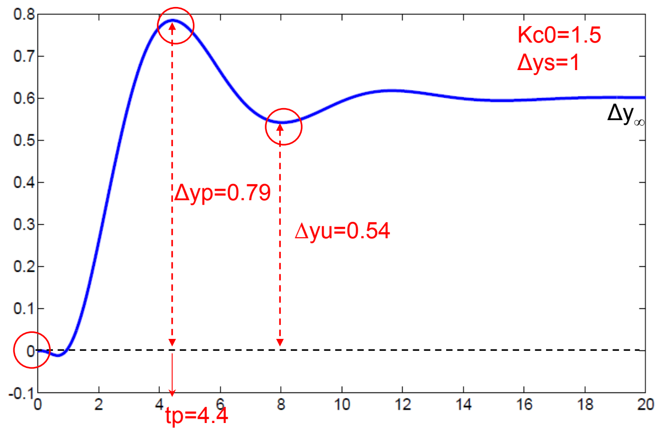


Fig. 14. Extracting information from closed-loop setpoint response with P-only controller.

limit. Another disadvantage is that it does not work for a simple second-order process. Finally, only two pieces of information are used (the controller gain K_u and the ultimate period P_u), so the method cannot possibly work on a wide range of first-order plus delay processes, which we know are described by three parameters (k, τ_1, θ).

Yuwana and Seborg (1982), and more recently Shamsuzzoha and Skogestad (2010), proposed a modification to the Ziegler-Nichols closed-loop experiment, which does not suffer from these three disadvantages. Instead of bringing the system to its limit of stability, one uses a P-controller with a gain that is about half this value, such that the resulting overshoot (D) to a step change in the setpoint is about 30% (that is, D is about 0.3).

We here describe the procedure proposed by Shamsuzzoha and Skogestad (2010). The system should be at steady-state initially, that is, before the setpoint change is applied. Then, from the closed-loop setpoint response one obtains the following parameters (see Figure 14):

- Controller gain used in experiment, K_{c0}
- Setpoint change, Δy_s
- Time from setpoint change to reach first (maximum) peak, t_p
- Corresponding maximum output change, Δy_p
- Output change at first undershoot, Δy_u

This seems to be the information that is most easy (and robust) to observe directly, without having to record and analyse all the data before finding the parameters. Also note that one may stop the experiment already at the first undershoot.

The undershoot Δy_u is used to estimate the steady-state output change (at infinite time) (Shamsuzzoha and Skogestad, 2010),

$$\Delta y_\infty = 0.45(\Delta y_p + \Delta y_u) \tag{12}$$

Alternatively, if one has time to wait for the experiment to settle, one may record Δy_∞ instead of Δy_u.

From this information one computes the relative overshoot and the absolute value of the relative steady-state offset, defined by:

- Overshoot, $D = \frac{\Delta y_p - \Delta y_\infty}{\Delta y_\infty}$.
- Steady-state offset, $B = \left|\frac{\Delta y_s - \Delta y_\infty}{\Delta y_\infty}\right|$.

Shamsuzzoha and Skogestad (2010) use this information to obtain directly the PI settings. Alternatively, we here propose a two-step procedure, where we first from K_{c0}, D, B and t_p obtain estimates for the parameters in a first-order plus delay model. We compute the parameters

$$A = 1.152D^2 - 1.607D + 1$$
$$r = 2A/B$$

and obtain the following first-order plus delay model (see Appendix for derivation):

$$k = 1/(K_{c0}B) \tag{13}$$
$$\theta = t_p \cdot (0.309 + 0.209e^{-0.61r}) \tag{14}$$
$$\tau_1 = r\theta \tag{15}$$

These values may subsequently be used with any tuning method, for example, the SIMC PI-rule. The closed-loop method may also be used for an unstable process, provided it can be approximated reasonably well by a stable first-order process. The extension to unstable processes is the reason for taking the absolute value when obtaining the steady-state offset B.

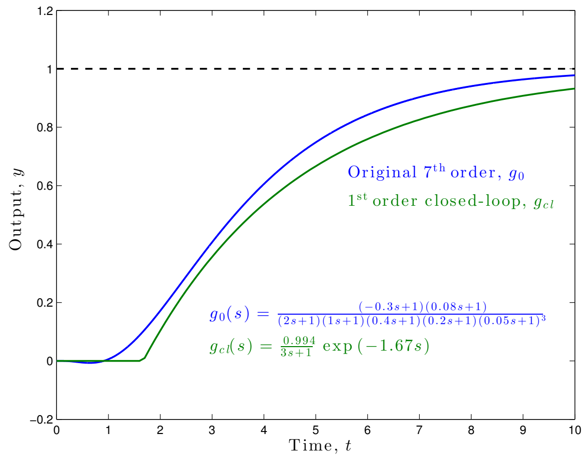


Fig. 15. Closed-loop setpoint identification method: Comparison of resulting open-loop step response with true model.

Example. *For the process (Skogestad, 2003)*

$$g_0(s) = \frac{(-0.3s+1)(0.08s+1)}{(2s+1)(1s+1)(0.4s+1)(0.2s+1)(0.05s+1)^3}$$

the closed-loop setpoint response with P-only controller with gain $K_{c0} = 1.5$ is shown in Figure 14. The following data is obtained from the closed-loop response

$$K_{c0} = 1.5, \Delta y_s = 1, \Delta y_p = 0.79, t_p = 4.4, \Delta y_u = 0.54$$

and we compute

$$\Delta y_\infty = 0.5985, D = 0.32, B = 0.67, A = 0.6038, r = 1.80$$

which using (13) - (15) gives the following first-order with delay model approximation,

$$k = 0.994, \theta = 1.67, \tau_1 = 3.00 \tag{16}$$

This gives a good approximation of the open-loop step response, as can seen by comparing the curves for g_0 (blue) and g_{cl} (green) in Figure 15. The approximation *is certainly not the best possible, but it should be noted that the objective is to use the model for tuning, and the resulting difference in the tuning, and thus closed-loop response, may be smaller than it appears by comparing the open-loop responses.*

7. CONCLUSION

Comparing the performance of the SIMC-rule with the optimal for a given robustness (M_s value) shows that the SIMC-rule give settings close to the Pareto-optimal (Figure 9). This means that the room for improving the SIMC PI-rule is limited, at least for the first-order plus delay processes considered in this paper, and with a good trade-off between rejecting input and output (setpoint) disturbances.

The tuning parameter τ_c should be chosen to get the desired trade-off between fast response (small IAE) on the one side, and smooth input usage and robustness (small M_s) on the other side. The recommended choice $\tau_c = \theta$ gives robust (M_s about 1.6 to 1.7) and somewhat conservative settings when compared with most other tuning rules. If it is desirable to get faster control one may consider reducing τ_c to about $\theta/2$ (see Figure 9). More commonly, one may want to have "smoother" control with $\tau_c > \theta$ and a smaller controller gain K_c.

To improve the performance for delay-dominant processes, one may replace τ_1 by $\tau_1 + \frac{\theta}{3}$ and use the "improved" SIMC PI-rule in (10)–(11).

REFERENCES

Alcantara, S., C. Pedret and R. Vilanova (2010). On the model matching approach to PID design: Analytical perspective for robust servo/regulator tradeoff tuning. *J. Process Control* **20**, 596–608.

Astrom, K.J. and T. Hagglund (1995). *PID controllers: Theory, design and tuning.* 2nd ed.. ISA-Instrument Society of America.

Di Ruscio, D. (2010). On Tuning PI Controllers for Integrating Plus Time Delay Systems. *Modeling, Identification and Control* **31**(4), 145-164.

Foley, M.W., N.R Ramharack and B.R Copeland (2005). Comparison of PI controller tuning methods. *Ind. Eng. Chem. Res.* **44**(17), 6741–6750.

Grimholt, C. (2010). Verification and improvements of the SIMC method for PI control. Technical report. 5th year project. Department of Chemical Engineering. Norwegian University of Science and Technology, Trondheim. http://www.nt.ntnu.no/users/skoge /diplom/prosjekt10/grimholt.

Haugen, F. (2010). Comparing PI tuning methods in a real benchmark temperature control system. *Modeling, Identifiation and Control* **31**, 79–91.

Rivera, D.E., M. Morari and S. Skogestad (1986). Internal model control. 4. PID controller design. *Ind. Eng. Chem. Res.* **25**(1), 252–265.

Seborg, D.E., T.F. Edgar and D.A. Mellichamp (1989). *Process Dynamics and Control.* John Wiley & Sons.

Shamsuzzoha, M. and S. Skogestad (2010). The setpoint overshoot method: A simple and fast method for closed-loop PID tuning. *J. Process Control* **20**, 1220–1234.

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

Skogestad, S. (2006). Tuning for smooth PID control with acceptable disturbance rejection. *Ind. Eng. Chem. Res.* **45**, 7817–7822.

Smith, O.J. (1957). Closer control of loops with dead time. *Chem. Eng. Prog.* **53**, 217.

Tyreus, B.D. and W.L. Luyben (1992). Tuning PI controllers for integrator/dead time processes. *Ind. Eng. Chem. Res.* pp. 2628–2631.

Yuwana, M. and D.E. Seborg (1982). A new method for on-line controller tuning. *AIChE Journal* **28**, 434–440.

Ziegler, J.G. and N.B. Nichols (1942). Optimum settings for automatic controllers. *Trans. of the A.S.M.E.* **64**, 759–768.

Appendix A. DERIVATION OF THE RELATIONSHIP BETWEEN τ_C AND GM, PM AND, $\Delta\theta/\theta$

This derivation applies to processes with a relatively small time constant, or specifically for processes where $\tau_1 < 4(\tau_c + \theta)$, such that we according the SIMC-rule have $\tau_I = \tau_1$. Consider a fist order model,

$$g = \frac{k}{\tau_1 s + 1} \exp(-\theta s)$$

and a PI controller

$$c = K_c \left(\frac{\tau_I s + 1}{\tau_I s} \right)$$

with the SIMC PI-settings

$$K_c = \frac{1}{k} \frac{\tau_1}{\tau_c + \theta}; \;\; \tau_I = \tau_1$$

This gives the loop transfer function

$$gc = \frac{1}{(\tau_c + \theta)s} \exp(-\theta s)$$

The magnitude and phase angles are

$$|gc(j\omega)| = \frac{1}{(\tau_c + \theta)\omega}$$

$$\angle gc(j\omega) = -\frac{\pi}{2} - \omega\theta$$

The gain crossover frequency is

$$|gc(j\omega_c)| = \frac{1}{(\tau_c + \theta)\omega_c} = 1; \;\; \omega_c = \frac{1}{\tau_c + \theta}$$

and phase crossover frequency is

$$\angle gc(j\omega_{180}) = -\frac{\pi}{2} - \omega_{180}\theta = -\pi; \;\; \omega_{180} = \frac{\pi}{2} \cdot \frac{1}{\theta}$$

The gain margin is then

$$GM = \frac{1}{|gc(j\omega_{180})|} = \frac{\pi}{2} \left(\frac{\tau_c}{\theta} + 1 \right)$$

The phase margin is

$$PM = \angle gc(j\omega_c) + \pi = \frac{\pi}{2} - \frac{\theta}{\tau_c + \theta}$$

and the allowable time delay error is

$$\frac{\Delta\theta}{\theta} = \frac{PM}{\omega_c} \cdot \frac{1}{\theta} = \frac{\pi}{2} \left(\frac{\tau_c}{\theta} + 1 \right) - 1 = GM - 1$$

Appendix B. ESTIMATION OF PARAMETERS τ_1 AND θ FROM CLOSED-LOOP STEP RESPONSE.

Shamsuzzoha and Skogestad (2010) discuss at the end of their paper a two-step closed-loop procedure, where the first step is to use closed-loop data and some expressions to obtain the parameters k, τ_1 and θ. We use this approach but have modified the expressions. Our expression for k in (13) is given by their equation (35) by noting that $B = |(1-b)/b|$ where $b = \Delta y_\infty/\Delta y_s$. However, our expressions for θ and τ_1 in (14)-(15) differ somewhat from their equations (36) and (37). The reason is that their equations (36) and (37) are not consistent in terms of the time delay estimate, because the expression for τ_1 in (36) is based on $\theta = 0.43 t_p$, whereas (37) uses $\theta = 0.305 t_p$. To correct for this, we first note from (19) in their paper (noting that $\tau_1 = \tau_I$ for the delay-dominant case), that τ_1 and θ are related by

$$\tau_1 = r\theta$$

where $r = 2A/B$, which is our expression in (15). Here, Shamsuzzoha and Skogestad (2010) recommend to use $\theta = 0.44 t_p$ for $\tau_1 < 8\theta$ and $\theta = 0.305 t_p$ for $\tau_1 > 8\theta$. However, to get better accuracy and a smooth transition, we fitted simulation data for θ/t_p as a function of τ_1/θ for a wide range of processes with an overshoot of 0.3, and obtained the correlation (Grimholt, 2010)

$$\theta = t_p \cdot (0.309 + 0.209 e^{-0.61(\tau_1/\theta)})$$

as given in (14). Note here that $(0.309 + 0.209 e^{-0.61(\tau_1/\theta)})$ is 0.518 for $r = \tau_1/\theta = 0$, and 0.309 for $r = \infty$.