The SIMC method for smooth PID controller tuning

Sigurd Skogestad and Chriss Grimholt

Abstract The SIMC method for PID controller tuning (Skogestad 2003) has already found widespread industrial usage in Norway. This chapter gives an updated overview of the method, mainly from a user's point of view. The basis for the SIMC method is a first-order plus time delay model, and we present a new effective method to obtain the model from a simple closed-loop experiment. An important advantage of the SIMC rule is that there is a single tuning parameter (τ_c) that gives a good balance between the PID parameters (K_c, τ_I, τ_D), and which can be adjusted to get a desired trade-off between performance ("tight" control) and robustness ("smooth" control). Compared to the original paper of Skogestad (2003), the choice of the tuning parameter τ_c is discussed in more detail, and lower and upper limits are presented for tight and smooth tuning, respectively. Finally, the optimality of the SIMC PI rules is studied by comparing the performance (IAE) versus robustness (M_s) trade-off with the Pareto-optimal curve. The difference is small which leads to the conclusion that the SIMC rules are close to optimal. The only exception is for pure time delay processes, so we introduce the "improved" SIMC rule to improve the performance for this case.

Chapter for PID book (planned: Springer, 2011, Editor: R. Vilanova)
This version: September 7, 2011

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

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

1 Introduction

Although the proportional-integral-derivative (PID) controller has only three parameters, it is not easy, without a systematic procedure, to find good values (settings) for them. In fact, a visit to a process plant will usually show that a large number of the PID controllers are poorly tuned. The tuning rules presented in this chapter have developed mainly as a result of teaching this material, where there are several objectives:

1. The tuning rules should be well motivated, and preferably model-based and analytically derived.
2. They should be simple and easy to memorize.
3. They should work well on a wide range of processes.

In this paper the simple two-step SIMC procedure (Skogestad 2003) that satisfies these objectives is summarized:

Step 1. Obtain a first- or second-order plus delay model.
Step 2. Derive model-based controller settings. PI-settings result if we start from a first-order model, whereas PID-settings result from a second-order model.

The SIMC method is based on classical ideas presented earlier by Ziegler and Nichols (1942), the IMC PID-tuning paper by Rivera *et al.* (1986), and the closely related direct synthesis tuning rules in the book by Smith and Corripio (1985). The Ziegler-Nichols settings result in a very good disturbance response for integrating processes, but are otherwise known to result in rather aggressive settings (Tyreus and Luyben 1992) (Astrom and Hagglund 1995), and also give poor performance for processes with a dominant delay. On the other hand, the analytically derived IMC-settings of Rivera *et al.* (1986) are known to result in poor disturbance response for integrating processes (Chien and Fruehauf 1990), (Horn *et al.* 1996), but are robust and generally give very good responses for setpoint changes. The SIMC tuning rule presented in this chapter works well for both integrating and pure time delay processes, and for both setpoints and load disturbances.

This chapter provides a summary of the original SIMC method and provides some new results on obtaining the model from closed-loop data, and on the Pareto-optimality of the SIMC method. There is some room for improvement for delay-dominant processes, and at the end of the chapter "improved" SIMC rules are presented.

Notation. The notation is summarized in Figure 1. Here u is the manipulated input (controller output), d the disturbance, y the controlled output, and y_s the setpoint (reference) for the controlled output. $g(s) = \frac{\Delta y}{\Delta u}$ denotes the process transfer function and $c(s)$ is the feedback part of the controller. Note that all the variables u, d and y are deviations from the initial steady state, but the Δ used to indicate deviation variables is usually omitted. Similarly, the Laplace variable s is often omitted to simplify notation. The settings given in this chapter are for the series (cascade, "interacting") form PID controller:

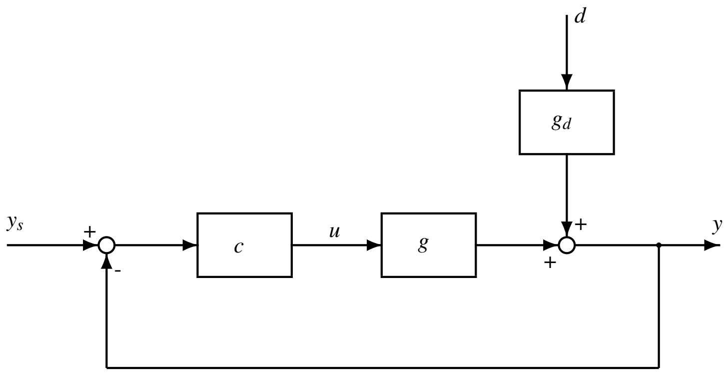

Fig. 1 Block diagram of feedback control system.
In this chapter we consider an input ("load") disturbance ($g_d = g$).

$$\text{Series PID}: \quad c(s) = K_c \cdot \left(\frac{\tau_I s+1}{\tau_I s}\right) \cdot (\tau_D s+1) = \frac{K_c}{\tau_I s}\left(\tau_I \tau_D s^2 + (\tau_I + \tau_D)s + 1\right) \tag{1}$$

where K_c is the controller gain, τ_I the integral time, and τ_D the derivative time. The reason for using the series form is that the PID rules with derivative action are then much simpler. The corresponding settings for the ideal (parallel form) PID controller are easily obtained using (30).

Simulations. The following practical PID controller (series form) is used in the simulations:

$$u(s) = K_c \left(\frac{\tau_I s+1}{\tau_I s}\right)\left(y_s(s) - \frac{\tau_D s+1}{(\tau_D/N)s+1} y(s)\right) \tag{2}$$

with $N = 10$. Note that we in order to avoid "derivative kick" do not differentiate the setpoint in (2). In most cases we use PI-control, i.e. $\tau_D = 0$, and the above implementation issues and differences between series and ideal form do not apply.

2 Model approximation (Step 1)

The first step in the SIMC design procedure is to obtain an approximate first- or second-order time delay model on the form

$$g_1(s) = \frac{k}{\tau_1 s+1} e^{-\theta s} = \frac{k'}{s+1/\tau_1} e^{-\theta s} \tag{3}$$

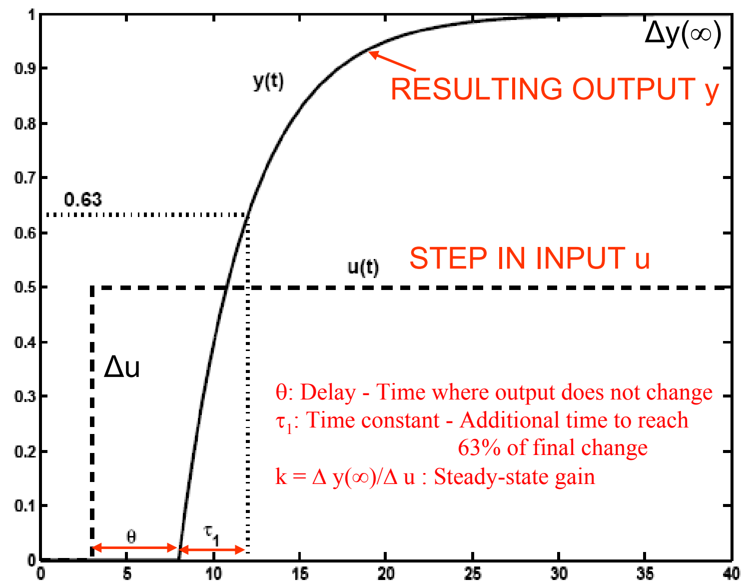

Fig. 2 Open-loop step response experiment to obtain parameters k, τ_1 and θ in first-order model (3)

$$g_2(s) = \frac{k}{(\tau_1 s+1)(\tau_2 s+1)} e^{-\theta s} \tag{4}$$

Thus, we need to estimate the following model information

- Plant gain, k
- Dominant lag time constant, τ_1
- (Effective) time delay (dead time), θ
- Optional: Second-order lag time constant, τ_2 (for dominant second-order process for which $\tau_2 > \theta$, approximately)

Such data may be obtained in many ways, three of which are discussed below.

1. From open-loop step response
2. From closed-loop setpoint response with P-controller
3. From detailed model: Approximation of effective delay using the half rule

2.1 Model from open-loop step response

In practice, the model parameters for a first-order model are commonly obtained from a step response experiment as shown in Figure 2. From a theoretical point of view this may not be the most effective method, but it has the advantage of being very simple to use and interpret.

For plants with a large time constant τ_1, one has to wait a long time for the process to settle. Fortunately, it is generally not necessary to run the experiment for longer than about 10 times the effective delay (θ). At this time, one may simply stop the experiment and either extend the response "by hand" towards settling, or approximate it as an integrating process (see Figure 3),

$$\frac{ke^{-\theta s}}{\tau_1 s+1} \approx \frac{k' e^{-\theta s}}{s} \tag{5}$$

where

- Slope, $k' \stackrel{\text{def}}{=} k/\tau_1$

is the slope of the integrating response. The reason is that for lag-dominant processes, i.e. for $\tau_1 > 8\theta$ approximately, the individual values of the time constant τ_1 and the gain k are not very important for controller design. Rather, their ratio k' determines the PI-settings, as is clear from the SIMC tuning rules presented below.

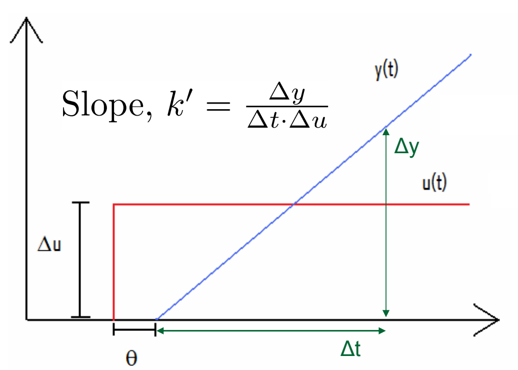

Fig. 3 Open-loop step response experiment to obtain parameters k' and θ in integrating model (5).

2.2 Model from closed-loop setpoint response

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 oscillations by use of a P-only controller. One disadvantage with the method is that the system is brought to its instability 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) which seems to use the most easily available parameters from the closed-loop response. 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 4):

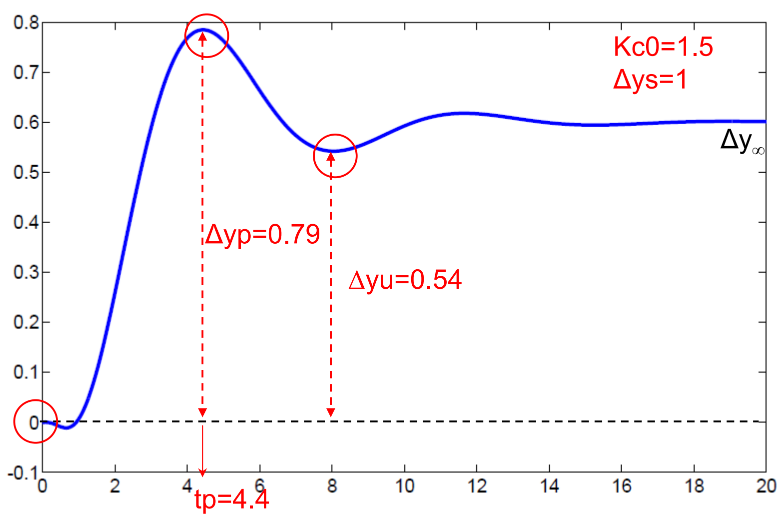

Fig. 4 Extracting information from closed-loop setpoint response with P-only controller.

- 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 analyze 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),

$$\Delta y_\infty = 0.45(\Delta y_p + \Delta y_u) \tag{6}$$

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 may use 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 (see the Appendix for details). We compute the parameters

$$A = 1.152D^2 - 1.607D + 1$$

$$r = 2A/B$$

and we obtain the following first-order plus delay model parameters from the closed-loop setpoint response (Figure 4):

$$k = 1/(K_{c0}B) \tag{7}$$

$$\theta = t_p \cdot (0.309 + 0.209e^{-0.61r}) \tag{8}$$

$$\tau_1 = r\theta \tag{9}$$

These values may subsequently be used with any tuning method, for example, the SIMC PI rules. 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.

Example E2(Skogestad 2003). *For the process*

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

the closed-loop setpoint response with P-only controller with gain $K_{c0} = 1.5$ *is shown in Figure 4. 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 (7) - (9) gives the following first-order with delay model approximation,

$$k = 0.994, \theta = 1.67, \tau_1 = 3.00 \tag{10}$$

This gives a good approximation of the open-loop step response, as can seen by comparing the curves for g_0 and g_{cl} in Figure 5. 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.

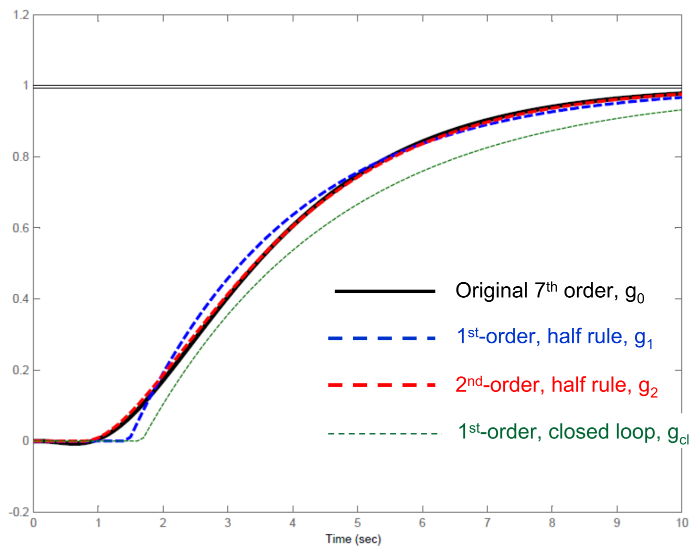

Fig. 5 Open-loop response to step change in input u for process E2, $g_0(s) = \frac{(-0.3s+1)(0.08s+1)}{(2s+1)(1s+1)(0.4s+1)(0.2s+1)(0.05s+1)^3}$ (solid line), and comparison with various approximations.

2.3 Approximation of detailed model using half rule

Assume that we have a given detailed transfer function model in the form

$$g_0(s) = \frac{\prod_j(-T_{j0}^{\text{inv}} s+1)}{\prod_i(\tau_{i0}s+1)} e^{-\theta_0 s} \tag{11}$$

where all the given parameters are positive and the time constants are ordered according to their magnitudes. To approximate this with a first or second-order time delay model, (3) or (4), Skogestad (2003) recommends that the "effective delay" θ is taken as the "true" delay θ_0, plus the inverse response (negative numerator) time constant(s) T^{inv}, plus half of the largest neglected time constant (half rule), plus all smaller time constant τ_{i0}. The "other half" of the largest neglected time constant is added to get at larger time constant τ_1 (or τ_2 for a second-order model).

Half rule: The largest neglected (denominator) time constant (lag) is distributed evenly to the effective delay (θ) and the smallest retained time constant (τ_1 or τ_2).

In summary, for a model in the form (11), to obtain a first-order model (3) we use

$$\tau_1 = \tau_{10} + \frac{\tau_{20}}{2}; \quad \theta = \theta_0 + \frac{\tau_{20}}{2} + \sum_{i\geq 3} \tau_{i0} + \sum_j T_{j0}^{\text{inv}} + \frac{h}{2} \tag{12}$$

and, to obtain a second-order model (4), we use

$$\tau_1 = \tau_{10}; \quad \tau_2 = \tau_{20} + \frac{\tau_{30}}{2}; \quad \theta = \theta_0 + \frac{\tau_{30}}{2} + \sum_{i\geq 4} \tau_{i0} + \sum_j T_{j0}^{\text{inv}} + \frac{h}{2} \tag{13}$$

where h is the sampling period (for cases with digital implementation).

Example E1. *Using the half rule, the process*

$$g_0(s) = \frac{1}{(s+1)(0.2s+1)}$$

is approximated as a first-order time delay process, $g(s) = ke^{-\theta s+1}/(\tau_1 s+1)$*, with* $k = 1, \theta = 0.2/2 = 0.1$ *and* $\tau_1 = 1 + 0.2/2 = 1.1$.

Example E2 (continued). *Using the half rule, the process*

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

is approximated as a first-order time delay process (3) with

$$\tau_1 = 2 + 1/2 = 2.5$$

$$\theta = 1/2 + 0.4 + 0.2 + 3 \cdot 0.05 + 0.3 - 0.08 = 1.47$$

or a second-order time delay process (4) with

$$\tau_1 = 2$$

$$\tau_2 = 1 + 0.4/2 = 1.2$$

$$\theta = 0.4/2 + 0.2 + 3 \cdot 0.05 + 0.3 - 0.08 = 0.77$$

The small positive numerator time constant $T_0 = 0.08$ was subtracted from the effective time delay according to rule T3 (see below). Both approximations, and in particular the second-order model, are very good as can be seen by from the open-loop step responses in Figure 5. Note that with the SIMC tuning rules, a first-order model yields a PI-controller, whereas a second-order model yields a PID controller.

Comment: In this case, we have $\tau_2 > \theta (1.2 > 0.77)$ for the second-order model, and the use of PID control is expected to yield a significant performance improvement compared to PI control (see below for details). However, adding derivative action has disadvantages, such as increased input usage and increased noise sensitivity.

2.4 Approximation of positive numerator time constants

A process model can also contain positive numerator time constants T_0 as the following process:

$$g(s) = g_0(s) \frac{T_0 s + 1}{\tau_0 s + 1} \tag{14}$$

Skogestad (2003) propose to cancel out the numerator time constant T_0 against a "neighboring" lag time constant τ_0 by the following rules: [1]

$$\frac{T_0 s + 1}{\tau_0 s + 1} \approx \begin{cases} T_0/\tau_0 & \text{for } T_0 \geq \tau_0 \geq \tau_c & \text{(Rule T1)} \\ T_0/\tau_c & \text{for } T_0 \geq \tau_c \geq \tau_0 & \text{(Rule T1a)} \\ 1 & \text{for } \tau_c \geq T_0 \geq \tau_0 & \text{(Rule T1b)} \\ T_0/\tau_0 & \text{for } \tau_0 \geq T_0 \geq 5\tau_c & \text{(Rule T2)} \\ \frac{(\tilde{\tau}_0/\tau_0)}{(\tilde{\tau}_0 - T_0)s + 1} & \text{for } \tilde{\tau}_0 \stackrel{\text{def}}{=} \min(\tau_0, 5\tau_c) \geq T_0 & \text{(Rule T3)} \end{cases} \tag{15}$$

Here τ_c is the desired closed-loop time constant, which appears as the tuning parameter in the SIMC PID rules. Because the tuning parameter is normally chosen after obtaining the effective time delay (the recommended value for "tight control" is $\tau_c = \theta$), one may not know this value before the model is approximated. Therefore, one may initially have to guess the value τ_c and iterate.

We normally select τ_0 as the closest *larger* denominator time constant ($\tau_0 > T_0$) and use Rules T2 or T3. Note that an integrating process corresponds to a process with an infinitely large time constant, $\tau_0 = \infty$. For example, for an integrating-pole-

[1] The rules are slightly generalized compared to Skogestad (2003) by replacing θ (effective time delay in final model) by τ_c (desired closed-loop time constant). This makes the rules applicable also to cases where τ_c is selected to be different from θ.

zero (IPZ) process on the form $k' \frac{e^{-\theta s}}{s} \frac{Ts+1}{\tau_2 s+1}$, we get $\frac{Ts+1}{s} \approx T$ (Rule T2 with $\tau_0 = \infty > T$). However, if T is smaller than τ_2 then we may use the approximation $\frac{Ts+1}{\tau_2 s+1} \approx \frac{T}{\tau_2}$ (Rule T2 with $\tau_2 > T > 5\theta$). Rule T3 would apply if T was even smaller.

However, if there exists no larger τ_0, or if there is smaller denominator time constant "close to" T_0, then we select τ_0 as the closest *smaller* denominator time constant ($\tau_0 < T_0$) and use rules T1, T1a or T1b. To define "close to" more precisely, let τ_{0a} (large) and τ_{0b} (small) denote the two neighboring denominator constants to T_0. Then, we select $\tau_0 = \tau_{0b}$ (small) if $T_0/\tau_{0b} < \tau_{0a}/T_0$ and $T_0/\tau_{0b} < 1.6$ (both conditions must be satisfied).

Derivations of the above rules and additional examples are given in (Skogestad 2003).

3 SIMC PI and PID tuning rules (step 2)

In step 2, we use the model parameters $(k, \theta, \tau_1, \tau_2)$ to tune the PID controller. We here derive the SIMC rules and apply them to some typical processes.

3.1 Derivation of SIMC rules

The SIMC rules may be derived using the method of direct synthesis for setpoints (Smith and Corripio 1985), or equivalently the Internal Model Control approach for setpoints (Rivera *et al.* 1986). For the system in Figure 1, the closed-loop setpoint response is

$$\frac{y}{y_s} = \frac{g(s)c(s)}{g(s)c(s)+1} \tag{16}$$

where we have assumed that the measurement of the output y is perfect. The idea of direct synthesis is to specify the desired closed-loop response and solve for the corresponding controller. From (16) we get

$$c(s) = \frac{1}{g(s)} \frac{1}{\frac{1}{(y/y_s)_{\text{desired}}} - 1} \tag{17}$$

We here consider the second-order time delay model $g(s)$ in (4), and specify that we, following the delay, desire a "smooth" first-order response with time constant τ_c

$$\left(\frac{y}{y_s}\right)_{\text{desired}} = \frac{1}{\tau_c s+1} e^{-\theta s} \tag{18}$$

The delay θ is kept in the "desired" response because it is unavoidable. Substituting (18) and (4) into (17) gives a "Smith Predictor" controller (Smith 1957):

$$c(s) = \frac{(\tau_1 s+1)(\tau_2 s+1)}{k} \frac{1}{(\tau_c s+1-e^{-\theta s})} \tag{19}$$

τ_c is the desired closed-loop time constant, and is the sole tuning parameter for the controller. To derive PID settings, we introduce in (19) a first-order Taylor series approximation of the delay, $e^{-\theta s} \approx 1-\theta s$. This gives

$$c(s) = \frac{(\tau_1 s+1)(\tau_2 s+1)}{k} \frac{1}{(\tau_c+\theta)s} \tag{20}$$

which is a series form PID-controller (1) with (Smith and Corripio 1985) (Rivera *et al.* 1986)

$$K_c = \frac{1}{k}\frac{\tau_1}{\tau_c+\theta} = \frac{1}{k'}\frac{1}{\tau_c+\theta}; \quad \tau_I = \tau_1; \quad \tau_D = \tau_2 \tag{21}$$

These settings are derived by considering the setpoint response. However, it is well known that for lag dominant processes with $\tau_1 \gg \theta$ (e.g. integrating processes), the choice $\tau_I = \tau_1$ results in a long settling time for *input ("load") disturbances* (Chien and Fruehauf 1990). To improve the load disturbance response, one may reduce the integral time, but not by too much, because otherwise we get slow oscillations and robustness problems. Skogestad (2003) suggests that a good trade-off between disturbance response and robustness is obtained by selecting the integral time such that we just avoid the slow oscillations, which with the controller gain given in (21) corresponds to

$$\tau_I = 4(\tau_c+\theta) \tag{22}$$

3.2 Summary of SIMC rules (original)

For a **first-order model**

$$g_1(s) = \frac{k}{(\tau_1 s+1)} e^{-\theta s} \tag{23}$$

the SIMC method results in a PI controller with settings

$$K_c = \frac{1}{k}\frac{\tau_1}{\tau_c+\theta} = \frac{1}{k'}\frac{1}{\tau_c+\theta} \tag{24}$$

$$\tau_I = \min\{\tau_1, 4(\tau_c+\theta)\} \tag{25}$$

The desired first-order *closed-loop* time constant τ_c is the only tuning parameter.

For a **second-order model**

$$g_2(s) = \frac{k}{(\tau_1 s+1)(\tau_2 s+1)} e^{-\theta s} \tag{26}$$

the SIMC method results in a PID controller with settings (cascade form)

$$K_c = \frac{1}{k}\frac{\tau_1}{\tau_c+\theta} = \frac{1}{k'}\frac{1}{\tau_c+\theta} \tag{27}$$

$$\tau_I = \min\{\tau_1, 4(\tau_c+\theta)\} \tag{28}$$

$$\tau_D = \tau_2 \tag{29}$$

Again, the desired first-order *closed-loop* time constant τ_c is the only tuning parameter. These PID settings are for the cascade (series) form in (1). The corresponding settings for the ideal (parallel form) PID controller are easily obtained using (30).

PID-control (with derivative action) is primarily recommended for processes with dominant second order-dynamics, defined as having $\tau_2 > \theta$, approximately. We note that the derivative time is then selected so as to cancel the second-largest process time constant.

In Table 1 we summarize the resulting tunings for a few special cases, including the pure time delay process, integrating process, and double integrating process. The double integrating process corresponds to a second-order process with $\tau_2 = \infty$ and direct application of the rules actually yield a PD controller, so in Table 1 integral action has been added to eliminate the offset for input disturbances.

The choice of the tuning parameter τ_c is discussed in more detail below. If the objective is to have "tight control" (good output performance) subject to having good robustness, then the recommendation is to choose τ_c equal to the effective time delay, $\tau_c = \theta$. The same recommendation for τ_c applies to both PI- and PID-control, but the actual values will differ, because the effective delay θ in a first-order model (PI control) will be larger than that in a second-order model (PID control) of a given process.

Example E2 (further continued). *We want to derive PI- and PID-settings for the process*

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

using the SIMC tuning rules with the "default" recommendation $\tau_c = \theta$. *From the closed-loop setpoint response, we obtained in a previous example a first-order model with parameters* $k = 0.994, \theta = 1.67, \tau_1 = 3.00$ *(10). The resulting SIMC PI-settings with* $\tau_c = \theta = 1.67$ *are*

$$K_c = 0.904, \tau_I = 3$$

Process	$g(s)$	K_c	τ_I	$\tau_D^{(5)}$
First-order, eq.(3)	$k\frac{e^{-\theta s}}{(\tau_1 s+1)}$	$\frac{1}{k}\frac{\tau_1}{\tau_c+\theta}$	$\min\{\tau_1, 4(\tau_c+\theta)\}$	-
Second-order, eq.(4)	$k\frac{e^{-\theta s}}{(\tau_1 s+1)(\tau_2 s+1)}$	$\frac{1}{k}\frac{\tau_1}{\tau_c+\theta}$	$\min\{\tau_1, 4(\tau_c+\theta)\}$	τ_2
Pure time delay$^{(1)}$	$ke^{-\theta s}$	0	$0^{\ (*)}$	-
Integrating$^{(2)}$	$k'\frac{e^{-\theta s}}{s}$	$\frac{1}{k'}\cdot\frac{1}{(\tau_c+\theta)}$	$4(\tau_c+\theta)$	-
Integrating with lag	$k'\frac{e^{-\theta s}}{s(\tau_2 s+1)}$	$\frac{1}{k'}\cdot\frac{1}{(\tau_c+\theta)}$	$4(\tau_c+\theta)$	τ_2
Double integrating$^{(3)}$	$k''\frac{e^{-\theta s}}{s^2}$	$\frac{1}{k''}\cdot\frac{1}{4(\tau_c+\theta)^2}$	$4(\tau_c+\theta)$	$4(\tau_c+\theta)$
IPZ process$^{(4)}$	$k'\frac{e^{-\theta s}}{s}\frac{Ts+1}{\tau_2 s+1}$	$\frac{1}{k'T}\cdot\frac{\tau_2}{\tau_c+\theta}$	$\min\{\tau_2, 4(\tau_c+\theta)\}$	-

Table 1 SIMC PID-settings (27)-(29) for some special cases of (4) (with τ_c as a tuning parameter).
(1) The pure time delay process is a special case of a first-order process with $\tau_1 = 0$.
(2) The integrating process is a special case of a first-order process with $\tau_1 \to \infty$.
(3) For the double integrating process, integral action has been added according to eq.(22).
(4) For the integrating-pole-zero (IPZ) process we assume $T > \tau_2$. Then $(Ts+1)/s \approx T$ (rule T2) and the PI-settings follow.
(5) The derivative time is for the cascade form PID controller in eq.(1).
(*) Pure integral controller $c(s) = \frac{K_I}{s}$ with $K_I = \frac{K_c}{\tau_I} = \frac{1}{k(\tau_c+\theta)}$.

From the full-order model $g_0(s)$ and the half rule, we obtained in a previous example a first-order model with parameters $k = 1, \theta = 1.47, \tau_1 = 2.5$. The resulting SIMC PI-settings with $\tau_c = \theta = 1.47$ are

$$K_c = 0.850, \tau_I = 2.5$$

From the full-order model $g_0(s)$ and the half rule, we obtained a second-order model with parameters $k = 1, \theta = 0.77, \tau_1 = 2, \tau_2 = 1.2$. The resulting SIMC PID-settings with $\tau_c = \theta = 0.77$ are

$$\text{Cascade PID}: \quad K_c = 1.299, \tau_I = 2, \tau_D = 1.2$$

The corresponding settings with the more common ideal (parallel form) PID controller are obtained by computing $f = 1 + \tau_D/\tau_I = 1.60$ and we have

$$\text{Ideal PID}: \quad K_c' = K_c f = 1.69, \tau_I' = \tau_I f = 3.2, \tau_D' = \tau_D/f = 0.75 \tag{30}$$

The closed-loop responses for the three controllers to a setpoint change at $t = 0$ and an input (load) disturbance at $t = 10$ is shown in Figure 6. The responses for the two PI controllers are very similar, as expected. The PID controller shows better output performance (upper plot), especially for the disturbance, but it may not be sufficient to outweigh the increased input usage (lower plot) and increased sensitivity to noise (not shown in plot).

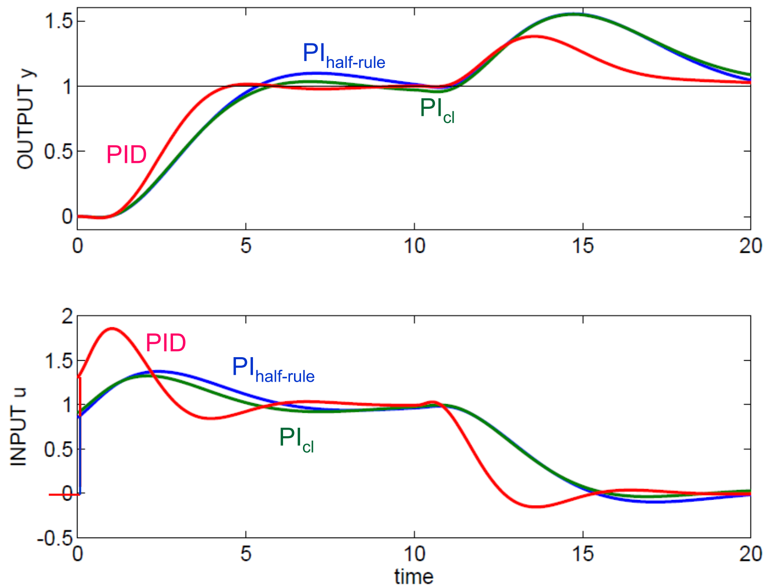

Fig. 6 Closed-loop responses for process E2 using SIMC PI- and PID-tunings with $\tau_c = \theta$. Setpoint change at $t = 0$ and input (load) disturbance at $t = 10$. For the PID controller, D-action is only on the feedback signal, i.e., not on the setpoint y_s.

4 Choice of tuning parameter τ_c

The value of the desired closed-loop time constant τ_c can be chosen freely, but from (27) we must have $-\theta < \tau_c < \infty$ to get a positive and nonzero controller gain. The optimal value of τ_c is determined by a trade-off between:

1. **Output performance (tight control):** Fast speed of response and good disturbance rejection (favored by a small value of τ_c). This "tightness" can be quantified by the magnitude of the setpoint error, $|y(t) - y_s(t)|$, which should be as small as possible. Here, one may consider different "norms" of the error, for example, the maximum deviation (∞-norm), the integrated square deviation (2-norm) and the integrated absolute error (IAE) (1-norm),

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

2. **Robustness (smooth control):** Good robustness, small input changes and small noise sensitivity (favored by a large value of τ_c). The "smoothness" is here quantified by the peak value $M_s \geq 1$ of the frequency-dependent sensitivity function, $S = 1/(1+gc)$. In terms of robustness, $1/M_s$ is the closest distance of the loop transfer function gc to the critical (-1)-point in the Nyquist diagram, so M_s

should be as small as possible. Notice that $M_s < 1.7$ guarantees gain margin (GM)> 2.43 and phase margin (PM)$> 34.2^o$ (Rivera *et al.* 1986).

In general, we have a multiobjective optimization problem, so there is no value of τ_c which is "optimal". We will consider in more detail the two limiting cases of "tight" and "smooth" control, and also consider in some detail the required input usage.

4.1 Tight control

With tight control, the primary objective is to keep the output close to its setpoint, but there should be some minimum requirement in terms of robustness and smoothness. A good trade-off is obtained by choosing τ_c equal to the time delay:

Tuning parameter τ_c. SIMC-recommendation for "tight control", or more precisely "tighest possible subject to maintaining smooth control":

$$\tau_c = \theta \tag{31}$$

The choice $\tau_c = \theta$ gives a reasonably fast response with moderate input usage and a good robustness with M_s about 1.6 to 1.7. More specifically, the robustness margins with the SIMC PID-settings in (27)-(29) and $\tau_c = \theta$, when applied to first- or second-order time delay processes, are always between the values given by the two columns in Table 2. The values in the left column in Table 2 apply to a case with a relatively small lag time constant (so $\tau_I = \tau_1$), and the somewhat less robust values in the right column apply to an integrating process (so $\tau_I = 4(\tau_c + \theta) = 8\theta$). For the integrating process, we reduce the integral time relative to the original value of $\tau_I = \tau_1$ to get better output performance for load disturbances, and not surprisingly we have to "pay" for this in terms of less robustness.

Process $g(s)$	$\frac{k}{\tau_1 s+1}e^{-\theta s}$	$\frac{k'}{s}e^{-\theta s}$
Controller gain, K_c ($\tau_c = \theta$)	$\frac{0.5}{k}\frac{\tau_1}{\theta}$	$\frac{0.5}{k'}\frac{1}{\theta}$
Integral time, τ_I	τ_1	8θ
Gain margin (GM)	3.14	2.96
Phase margin (PM)	61.4^o	46.9^o
Allowed time delay error, $\Delta\theta/\theta$	2.14	1.59
Sensitivity peak, M_s	1.59	1.70
Complementary sensitivity peak, M_t	1.00	1.30
Phase crossover frequency, $\omega_{180}\cdot\theta$	1.57	1.49
Gain crossover frequency, $\omega_c\cdot\theta$	0.50	0.51

Table 2 "Tight" settings: Robustness margins for first-order and integrating time delay process for SIMC-rules (24)-(25) with $\tau_c = \theta$. The same margins apply to a second-order process (4) if we choose $\tau_D = \tau_2$ in (29).

To be more specific, for processes with a relatively small time constant where we use $\tau_I = \tau_1$ (left column), the system always has a gain margin GM=3.14 and phase margin PM=61.4^o, which is much better than the typical minimum requirements GM$>$ 1.7 and PM$>$ 30^o (Seborg *et al.* 1989). The sensitivity and complementary sensitivity peaks are $M_s = 1.59$ and $M_t = 1.00$ (here small values are desired with a typical upper bound of 2). The maximum allowed time delay error is $\Delta\theta/\theta = \mathrm{PM}\ [\mathrm{rad}]/(w_c \cdot \theta)$, which in this case gives $\Delta\theta/\theta = 2.14$ (i.e., the system goes unstable if the time delay is increased from θ to $(1+2.14)\theta = 3.14\theta$).

For an integrating processes (right column) and $\tau_I = 8\theta$, the suggested "tight" settings give GM=2.96, PM=46.9^o, $M_s = 1.70$ and $M_t = 1.30$, and the maximum allowed time delay error is $\Delta\theta = 1.59\theta$.

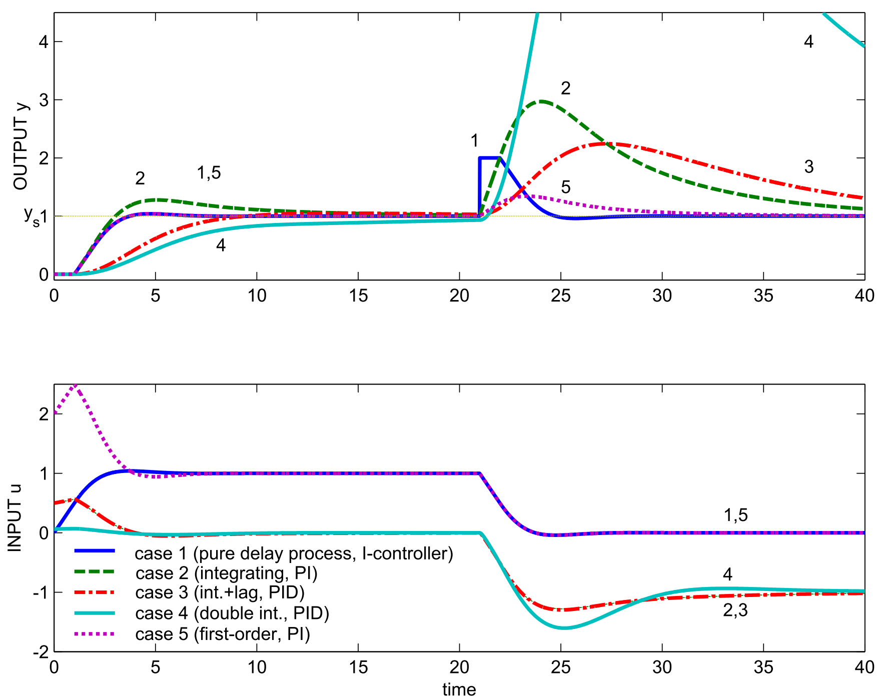

Fig. 7 Responses using SIMC settings for the five time delay processes ($\tau_c = \theta$).
Unit setpoint change at $t = 0$; Unit load disturbance at $t = 20$.
Simulations are without derivative action on the setpoint.
Parameter values: $\theta = 1, k = 1, k' = 1, k'' = 1$.

The simulated time responses to setpoint changes and disturbances with SIMC-settings are shown for five cases in Figure 7 (Skogestad 2003). Even though these are for the "tight" settings ($\tau_c = \theta$), the responses are all smooth. This means that it is certainly possible to get even tighter responses by choosing a smaller value, for example $\tau_c = 0.5\theta$, but for most process control applications this is not recommended because of less robustness, larger input usage and more sensitivity to noise. It may seem from Figure 7 that the SIMC PID-controller does not work well for the double integrating process (curve 4), but this is a difficult process to control and the response to a unit input disturbance will be large for any robust controller.

4.2 Smooth control

Even though the recommended "tight" settings ($\tau_c = \theta$) gives responses that are reasonably smooth, they may still be unnecessary aggressive compared to the required performance objectives, especially if the effective delay θ is small. For example, for the limiting case with $\theta = 0$ (no delay), we get with $\tau_c = \theta$ an infinite controller gain, which is clearly not realistic. Thus, in practice one often uses a "smoother" tuning, that is, $\tau_c > \theta$.

However, τ_c should not be too large, because otherwise the output y will go out of bound when there are disturbances d. The question is: *How slow (smooth) can we tune the controller and still get acceptable control?* This issue is addressed in the paper by Skogestad (2006) on "tuning for smooth PID control with acceptable disturbance rejection", where the following lower bound on the controller gain is derived (for both PI- and PID-control).

Controller gain. SIMC-recommendation for "smooth control", or more precisely "smoothest possible subject to acceptable disturbance rejection":

$$|K_c| > |K_{c,min}| = \frac{|\Delta u_0|}{|\Delta y_{max}|} \tag{32}$$

where
Δy_{max} = maximum allowed deviation in the output y
Δu_0 = required input change to reject the disturbance(s) d.

Substituting $K_{c,min}$ into (24) or (27) one can obtain the corresponding value $\tau_{c,max}$, and we end up with a region of recommended values for the tuning parameter τ_c:

$$\tau_{c,min} \text{ (“tight”)} < \tau_c < \tau_{c,max} \text{ (“smooth”)} \tag{33}$$

where

$$\tau_{c,min} = \theta, \quad \tau_{c,max} = \frac{1}{K_{c,min}} \cdot \frac{\tau_1}{k} - \theta \tag{34}$$

The final choice of τ_c is an engineering decision. A small value for τ_c ("tight control" of y) is typically desired for control of active constraints, because tight control reduces the required backoff (safety margin to the constraint). On the other hand, tight control will require larger input changes which may disturb the rest of the process. For example, for liquid level there is usually no reason to control the level tightly, so a large value of τ_c ("smooth control") is desired.

Details on the derivation of (32) and $\tau_{c,max}$ are given in (Skogestad 2006), but let us here give a simplified version. Consider disturbance rejection and assume we use a P-only controller with gain K_c. The input change (in deviation from the nominal value) is then $\Delta u = -K_c \Delta y$ or

$$|\Delta u| = |K_c| \cdot |\Delta y|$$

Assume that the required input change to reject a disturbance is Δu_0. For example, if we have a disturbance Δd_I at the input, then $\Delta u_0 = -\Delta d_I$. The smallest controller gain that can generate the required input change Δu_0 is obtained when we have the largest output change ($|\Delta y| = |\Delta y_{max}|$), and we get

$$|\Delta u_0| = |K_{c,min}| \cdot |\Delta y_{max}|$$

and (32) follows.

4.3 Input usage

The magnitude of the dynamic input change can be an important issue when tuning the controller, that is, when selecting the value for τ_c. The transfer function from the disturbance d to the input u is given by (see Figure 1):

$$u(s) = -\frac{g_d c}{1 + gc} d(s)$$

With integral action in the controller (e.g., PI or PID control), the steady-state input change to a step disturbance d is independent of the controller and is given by $u(t = \infty) = -\frac{k_d}{k} d$ where k_d is the steady-state disturbance gain and k is the steady-state process gain. We assume that we can reject the expected disturbances at steady-state, that is, we assume $|u(t = \infty)| = |\frac{k_d}{k} d| \leq |u_{max}|$ where $|u_{max}|$ and $|d|$ is the magnitude of the disturbance change, is the maximum allowed input change, because otherwise the process is not "controllable" (with any controller). However, the dynamic input change $u(t)$ will depend on the controller tuning, and we will consider the initial change (at $t = 0^+$) just after a step disturbance d.

We consider two important disturbances, namely an input "load" disturbance d_u (corresponding to $g_d = g$), and an output disturbance d_y (corresponding to $g_d = 1$). Note that an output disturbance has an immediate effect on the output y. A physical example is a process where we add another stream (output disturbance) just before the measurement y. Mathematically, an output disturbance is equivalent to a setpoint change (with $y_s = -d_y$)

For an *input ("load") disturbances* d_u, input usage is not an important issue for SIMC-tuning, even dynamically. This is because the SIMC controller gives a closed-loop transfer function $\frac{y}{y_s} = \frac{gc}{1+gc}$ with little or no overshoot, see (16) and (18), and since $\frac{u}{d_u} = -\frac{gc}{1+gc}$, we get for d_u a corresponding input response with little overshoot. This is illustrated by the input changes for a load disturbance ($t = 20$) in Figure 7.

On the other hand, for an *output disturbances* d_y ($g_d = 1$), or equivalently for a *setpoint change* $y_s = -d_y$, input usage may be an important issue for tuning. The steady-state input change to a step setpoint change y_s is $u(t=\infty) = \frac{1}{k} y_s$. However, with PI-control the input will initially jump to the value $u(t = 0^+) = K_c y_s$, as illustrated for the setpoint change in Figure 7 (e.g., see the first-order process, case 5). This initial change is larger than the steady-state change if $K_c k > 1$, which is usually the case, except for delay-dominant processes. If we assume that the allowed input change is $u_{\max}$, then to avoid input saturation we must select τ_c such that (SIMC PI control):

$$|u(t=0^+)| = |K_c y_s| = |\frac{\tau_1}{\tau_c + \theta}\frac{1}{k} y_s| \leq |u_{\max}| \tag{35}$$

Note that u and y_s are deviation variables. Consider, for example, a first-order process with $\tau_1 = 8$ and $\theta = 1$. With the choice $\tau_c = \theta$, the initial input change is $\tau_1/(\tau_c + \theta) = 4$ times the steady-state input change y_s/k. If such a large dynamic input change is not feasible then one would need to use “smoother” control with a larger value for τ_c in order to satisfy (35).[2]

With PID control, the derivative action will cause even larger input changes for output disturbances and this may be one reason for reducing or even avoiding derivative action. It is also the reason why we to avoid “derivative kick”, recommend that the setpoint is not differentiated, see (2).

5 Optimality of SIMC PI rules

How good are the SIMC PI rules, that is, how much room is there for improvements? To study this, we compare the SIMC PI performance, with τ_c as a parameter, to the “Pareto-optimal” PI-controller. Pareto-optimality applies to multiobjective problems, and means that no further improvement can be made in objective 1 (output performance in our case) without sacrificing objective 2 (robustness and input usage in our case).

We choose to quantify robustness and input usage in terms of the sensitivity peak M_s. We also considered other “robustness” measures, for example, the relative delay margin as suggested by Foley *et al.* (2005), but we choose to use M_s. One reason is that we found that the M_s-value correlates well with the input usage as given by its total variation (TV), which agrees with the findings of Foley *et al.* (2005). Such a

[2] It may seem from (35) that “slow” processes, which have a large time constant τ_1, will always require “slow” control (large τ_c) in order to avoid excessive input changes. However, this is usually not the case because such processes often have a corresponding large gain k, such that the value $k' = k/\tau_1$ may be sufficiently large to satisfy (35) even with $\tau_c = \theta$.

correlation is reasonable since a large M_s-value corresponds to an oscillatory system with large input variations.

We choose to quantify performance in terms of the integrated absolute error in response to a setpoint change (IAE_{ys}) and to an input "load" disturbance (IAE_d). The setpoint performance is often referred to as the "servo" behavior and the disturbance (in this case the input "load" disturbance) performance is often referred to as "regulator" behavior. It may be argued that a two-degree of freedom controller ("feedforward action") may be used to improve the response for setpoints, but note that a setpoint change is equivalent to an output disturbance (with $g_d = 1$ in Figure 1) which can only be counteracted by feedback. 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 defining the following performance cost

$$J(c) = 0.5\left[\frac{\mathrm{IAE}_{ys}(c)}{\mathrm{IAE}_{ys}^o} + \frac{\mathrm{IAE}_d(c)}{\mathrm{IAE}_d^o}\right] \tag{36}$$

where the reference values, IAE_{ys}^o and IAE_d^o, are for IAE-optimal PI-controllers (with $M_s = 1.59$) for a setpoint change and input disturbance, respectively. We could have used the truly optimal IAE-value as the reference when computing J (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 $M_s = 1.59$. Table 3 gives the tunings and reference values obtained using IAE-optimal PI-controllers (with $M_s = 1.59$) for four different processes, and Table 4 gives the tunings, costs J and M_s-values for the SIMC PI-controller (with $\tau_c = \theta$). Importantly, the weighted cost J is independent of the process gain k and the disturbance magnitude, and also of the unit used for time. 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 $J(c)$.

	Setpoint			Input disturbance			Optimal combined (minimize J)					
Process	K_c	τ_I	IAE_{ys}^o	K_c	τ_I	IAE_d^o	K_c	τ_I	IAE_{ys}	IAE_d	J	M_s
e^{-s}	0.20	0.32	1.607	0.20	0.32	1.607	0.20	0.32	1.607	1.607	1	1.59
$\frac{e^{-s}}{s+1}$	0.54	1.10	2.083	0.50	1.0	2.036	0.54	1.10	2.083	2.041	1.00	1.59
$\frac{e^{-s}}{8s+1}$	4.0	8	2.169	3.34	3.7	1.135	3.46	4.0	3.111	1.158	1.23	1.59
$\frac{e^{-s}}{s}$	0.50	∞	2.169	0.40	5.8	15.09	0.41	6.3	4.314	15.4	1.51	1.59

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

Table 3 Optimal PI-controllers ($M_s = 1.59$) and corresponding IAE-values for four processes.

Figure 8 shows the trade-off between performance (J) and robustness (M_s) for the SIMC PI-controller (blue solid curve) and the Pareto-optimal controller (dashed black curve) for four different processes: pure time delay ($\tau_1/\theta = 0$), small time constant ($\tau_1/\theta = 1$), intermediate time constant ($\tau_1/\theta = 8$), and integrating process ($\tau_1/\theta = \infty$). The curve for the SIMC controller was generated by varying the tuning

Process	SIMC PI ($\tau_c = \theta$)						Improved SIMC PI ($\tau_c = \theta$)					
	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 $K_I = K_c/\tau_I = 0.5$.

Table 4 SIMC PI-controllers ($\tau_c = \theta$) and corresponding J- and M_s-values for four processes.

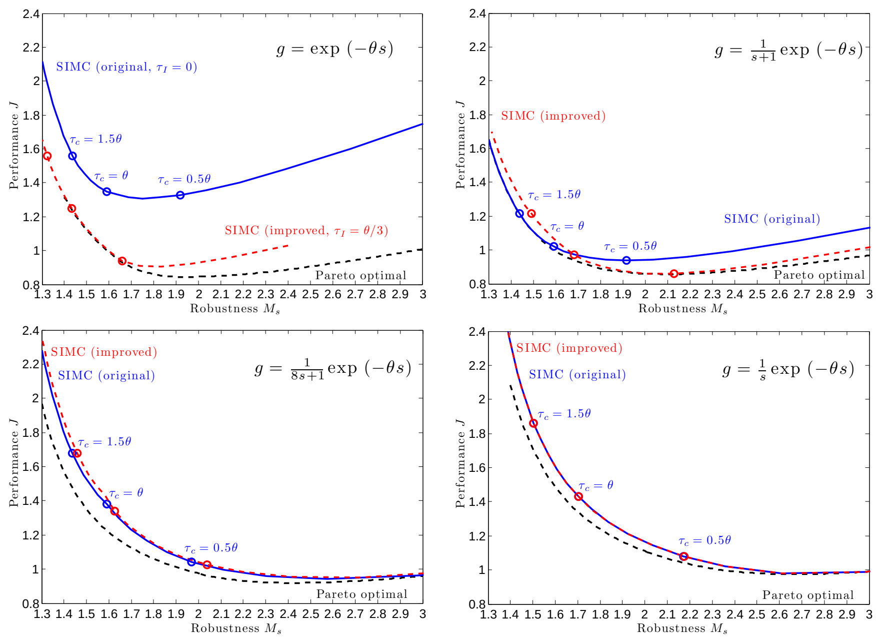

Fig. 8 Check of optimality of SIMC PI tuning rules for four processes.

parameter τ_c from a large to a small value. The controllers corresponding to the choices $\tau_c = 1.5\theta$ (smoother), $\tau_c = \theta$ (recommended) and $\tau_c = 0.5\theta$ (aggressive) are shown by circles. The Pareto-optimal curve was generated by finding for each value of M_s, the optimal PI-controller c, with the smallest IAE-value $J(c)$. Except for the pure time delay process, the differences between the J-values for SIMC (blue solid curve) and optimal (dashed black curve) are small (within 10%), which shows that the SIMC PI-rules are close to optimal.

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 (in the left region in the figures in Figure 8). We never want to be in the region with a zero or positive slope (to the right in the figures), because here we can improve both performance (J) and robustness (M_s) at the same time with another choice for the tuning parameter (using a larger value for τ_c). Another important observation from Figure 8 is then that the

SIMC-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 give 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 rules, at least when performance and robustness are as defined above (J and M_s).

The IAE-cost J in (36) is based on equal weighting of servo (output disturbance) and regulator (input disturbance) performance. The existence of a trade-off between servo and regulator performance, can be quantified by considering how much larger the (Pareto) optimal cost J_{opt} (dashed black line) is than 1 at the reference robustness, $M_s = 1.59$, see also Table 3. For a pure time delay-process, we have that $J_{opt} = 1$ for $M_s = 1.59$ and there is no trade-off. The reason is that the setpoint and output disturbance responses are the same. On the other hand, for the other extreme of an integrating process, we have a clear trade off since the optimal PI-controller has $J_{opt} = 1.51$ (the SIMC PI-controller with $M_s = 1.59$ is close to this with J about 1.6). The existence of the servo/regulator trade-off for an integrating process, implies that one for a given robustness (M_s-value) can find PI-settings with significantly better regulator (load disturbance) performance or better servo (setpoint) performance, but not both at the same time. To be able to shift the trade-off, one may introduce an extra parameter in the PID rules (Alcantara *et al.* 2010), in addition to τ_c. For the SIMC method, this extra servo/regulator trade-off parameter could be c in the following expression for the integral time,

$$\tau_I = \min(\tau_1, c(\tau_c + \theta)) \tag{37}$$

where $c = 4$ gives the original SIMC-rule. A larger value if c improves the setpoint performance, and a smaller value, e.g. $c = 2$, improves the input disturbance performance (Haugen 2010). However, introducing an extra parameter adds complexity and the potential benefit does not seem sufficiently large. Nevertheless, one may consider choosing another (lower) fixed value for c. There are two reasons why we recommend keeping the SIMC-value of $c = 4$. First, it is close to the Pareto-optimal PI controller (as seen from Figure 8), so we cannot get a significant improvement with our performance objective J. Second, with a smaller value for c, say $c = 2.5$, the recommended choice $\tau_c = \theta$ becomes less robust (with a higher M_s), so one would need to recommend a different value for τ_c for an integrating process, say $\tau_c = 1.5\theta$, which would add complexity. In summary, we find that the value $c = 4$ in the original SIMC rule provides a well-balanced servo/regulator trade-off.

6 Improved SIMC tuning rules

For a pure time delay process, we see from Figure 8 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 9 where we see that the SIMC PI-controller (denoted SIMC-original in the figure) gives a nice and smooth response. However, the response 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 $K_I = 0.62$). In fact, the optimal PI-controller for a pure time delay process (dashed black line in Figure 8), has an almost fixed integral time of approximately $\theta/3$ for all values of M_s between 1.4 and 1.7.

Based on this fact, we propose a simple change to the SIMC-rules, namely to replace τ_1 by $\tau_1 + \theta/3$ in the rules (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 rules.

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 9, 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 proposed value $\theta/3$ gives a faster settling and is also closer to the original SIMC-rule (which is zero for a time delay process). The conclusion is that we recommend to replace τ_1 by $\tau_1 + \theta/3$ in the SIMC rules to get the improved SIMC rules:

Improved SIMC PI-rule for first-order with delay process.

$$K_c = \frac{1}{k} \frac{\tau_1 + \frac{\theta}{3}}{\tau_c + \theta} \tag{38}$$

$$\tau_I = \min\{\tau_1 + \frac{\theta}{3}, 4(\tau_c + \theta)\} \tag{39}$$

The improvement of this rule for a pure time delay processes is clear from the red curves in Figures 9 and 8 (upper left); 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. For the process with a small time constant ($\tau_1 = \theta$), the improved SIMC rule (red curve in upper right plot in Figure 8) is slightly better than the "original" SIMC rule (blue 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 ($\tau_1 = 8\theta$ and $\tau_1 = \infty$) there are, as expected, almost no difference between the original and improved SIMC rules.

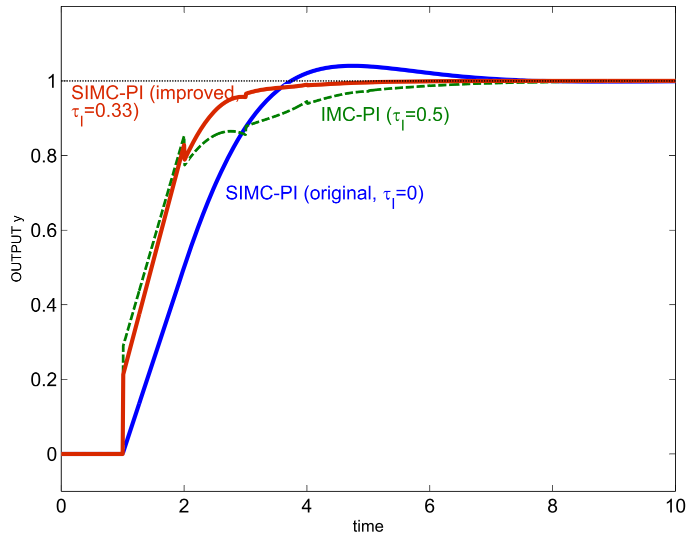

Fig. 9 Closed-loop setpoint responses for pure time delay process ($\theta = 1, k = 1, \tau_1 = 0$) with PI-control. All three controllers have the same robustness ($M_s = 1.59$).
For a pure time delay process, the setpoint and disturbance responses are identical, and the input and output are identical.
IMC PI: $K_c = 0.29$ and $\tau_I = 0.5$ ($K_I = K_c/\tau_I = 0.58$).
SIMC PI original ($\tau_c = \theta$): $K_c = 0$ and $\tau_I = 0$ ($K_I = 0.5$).
SIMC PI improved ($\tau_c = 0.61\theta$): $K_c = 0.207$ and $\tau_I = 0.333$ ($K_I = 0.62$).

7 Discussion

7.1 Measurement noise

Measurement noise has not been considered in this chapter, but it is an important consideration in many cases, especially if the proportional gain K_c is large, or, for cases with derivative action, if the derivative gain $K_c\tau_D$ is large. However, since the magnitude of the measurement noise varies a lot in applications, it is difficult to give general rules about when measurement noise may be a problem. In general, robust designs (with small M_s) are insensitive to measurement noise. Therefore, the SIMC rules with the recommended choice $\tau_c = \theta$, are less sensitive to measurement noise than most other published settings method, including the Ziegler-Nichols-settings. If actual implementation shows that the sensitivity to measurement noise is too large, then the following modifications may be attempted:

1. Filter the measurement signal, for example, by sending it through a first-order filter $1/(\tau_F s + 1)$; see also (2). With the proposed SIMC-settings one can typically increase the filter time constant τ_F up to almost 0.5θ, without a large affect on performance and robustness.

2. If derivative action is used, one may try to remove it, and obtain a first-order model before deriving the SIMC PI-settings.
3. If derivative action has been removed and filtering the measurement signal is not sufficient, then the controller needs to be detuned by selecting a larger value for τ_c.

7.2 *Retuning for integrating processes*

Integrating processes,

$$g(s) = k' \frac{e^{-\theta s}}{s}$$

are common in industry, but control performance is often poor because of incorrect controller settings. When encountering oscillations, the intuition of the operators is to reduce the controller gain. If the oscillations are relatively slow, then this is the exactly opposite of what one should do for an integrating process. The product of the controller gain K_c and the integral time τ_I must be *larger* than $4/k'$ to avoid slow oscillations (Skogestad 2003). One solution is to simply use proportional control (with $\tau_I = \infty$), but this is often not desirable. Here we show how to easily retune the controller to just avoid the oscillations without actually having to derive a model. This approach has been applied with success to industrial examples.

Consider a PI controller with (initial) settings K_{c0} and τ_{I0} which results in "slow" oscillations with period P_0 (larger than $3 \cdot \tau_{I0}$, approximately). Then we likely have a close-to integrating process for which the product of the controller gain and integral time ($K_{c0}\tau_{I0}$) is too low. To avoid oscillations with the new settings K_c and τ_I we must require (Skogestad 2003):

$$\frac{K_c \tau_I}{K_{c0}\tau_{I0}} \geq \frac{1}{\pi^2} \cdot \left(\frac{P_0}{\tau_{i0}}\right)^2 \tag{40}$$

Here $1/\pi^2 \approx 0.10$, so we have the **rule**:

- To avoid "slow" oscillations the product of the controller gain and integral time should be increased by a factor $f \approx 0.1(P_0/\tau_{I0})^2$.

7.3 *Controllability*

The effective delay θ is easily obtained using the proposed half rule. Since the effective delay is the main limiting factor in terms of control performance, its value gives invaluable insight about the inherent controllability of the process.

From the settings in (27)-(29), a PI-controller results from a first-order model, and a PID-controller from a second-order model. With the effective delay computed using the half rule in (12)-(13), it then follows that PI-control performance is limited by (half of) the magnitude of the second-largest time constant τ_2, whereas PID-control performance is limited by (half of) the magnitude of the third-largest time constant, τ_3.

8 Conclusions and Future Perspectives

This chapter has summarized the SIMC two-step procedure for deriving PID settings for typical process control applications.

Step 1. The real process is approximated by a first-order with delay model (for PI control) or a second-order model (for PID control). To obtain the model, the simplest approach is probably to use an open-loop step experiment (Figure 3), but if this is difficult for some reasons, then one may alternatively use a closed-loop setpoint response with P-controller (Figure 4). If the starting point is a detailed model, then the half rule may be used to obtain the effective delay θ, see (12)-(13).

Step 2. For a first-order model (with parameters k, τ_1 and θ) the following SIMC PI-settings are suggested (original SIMC rule):

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

where the closed-loop response time τ_c is the tuning parameter. For a dominant second-order process (for which $\tau_2 > \theta$, approximately), one needs to add derivative action with

$$\text{Series} - \text{form PID}: \quad \tau_D = \tau_2$$

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-rules in (38)-(39). A more careful analysis needs to be done to check if a similar improvement can be used with a PID controller.

Note that although the same formulas are used to obtain K_c and τ_I for both PI- and PID-control, the actual values will differ since the effective delay θ is smaller for a second-order model. 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, and if it is desirable to get faster control one may consider reducing τ_c to about $\theta/2$ (see Figure 8). More commonly, one may want to have "smoother" control with $\tau_c > \theta$ and a smaller controller gain K_c. However, the controller gain must be larger than the value given in (32) to achieve a minimum level of disturbance rejection.

Comparing the performance of the SIMC-rules with the optimal for a given robustness (M_s value) shows that the SIMC-rules are close to the Pareto-optimal settings (Figure 8). This means that the room for improving the SIMC PI-rules is limited, at least for the first-order plus delay processes considered in this chapter, and with a good trade-off between rejecting input and output (setpoint) disturbances.

However, it should be noticed that the SIMC rules apply to processes that can be reasonably well approximated by first or second order plus delay models. This applies to most process control applications, including some unstable plants, but it obviously does not apply in general, for example, for some of the unstable or oscillating processes found in mechanical systems. For such processes, it would be interesting to study the validity and extension of the SIMC rules or similar analytic model-based PID tuning rules. It is also interesting to establish for which processes the PID controller is a suitable controller and for which processes it is not.

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.

Chien, I.L. and P.S. Fruehauf (1990). Consider IMC tuning to improve controller performance. *Chemical Engineering Progress* pp. 33–41.

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.

Horn, I.G., J.R. Arulandu, J. Gombas, J.G. VanAntwerp and R.D. Braatz (1996). Improved filter design in Internal Model Control. *Ind. Eng. Chem. Res.* **35**(10), 3437–3441.

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, C.A. and A.B. Corripio (1985). *Principles and Practice of Automatic Process Control*. John Wiley & Sons.

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

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 expressson for k in (7) 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 (8)-(9) 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 $\theta = 0.43t_p$, whereas (37) uses $\theta = 0.305t_p$. 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 (9). Here, Shamsuzzoha and Skogestad (2010) recommend to use $\theta = 0.44t_p$ for $\tau_1 < 8\theta$ and $\theta = 0.305t_p$ 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)

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

as given in (8). Note here that $(0.309 + 0.209e^{-0.61(\tau_1/\theta)})$ is 0.518 for $r = \tau_1/\theta = 0$, and 0.309 for $r = \infty$.