

Simple analytic rules for model reduction and PID controller tuning*

SIGURD SKOGESTAD†

Keywords: *Process control; feedback control; IMC; PI-control; integrating process; time delay*

The aim of this paper is to present analytic rules for PID controller tuning that are simple and still result in good closed-loop behavior. The starting point has been the IMC-PID tuning rules that have achieved widespread industrial acceptance. The rule for the integral term has been modified to improve disturbance rejection for integrating processes. Furthermore, rather than deriving separate rules for each transfer function model, there is a just a single tuning rule for a first-order or second-order time delay model. Simple analytic rules for model reduction are presented to obtain a model in this form, including the 'half rule' for obtaining the effective time delay.

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 paper 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 a simple two-step procedure that satisfies these objectives is presented:

- Step 1.* Obtain a first- or second-order plus delay model. The effective delay in this model may be obtained using the proposed half-rule.
- 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.

There has been previous work along these lines, including the classical paper by Ziegler & 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 & 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 & Luyben, 1992; Astrom & Hagglund, 1995), and also give poor performance

*Reprinted from *Journal of Process Control*, Vol. 13, pp. 291–309, Copyright 2003, with permission from Elsevier. Additional remarks from the author are given at the end of the paper.

†Department of Chemical Engineering, Norwegian University of Science and Technology, N-7491 Trondheim Norway, tel.: +47-7359-4154, email: skoge@chemeng.ntnu.no

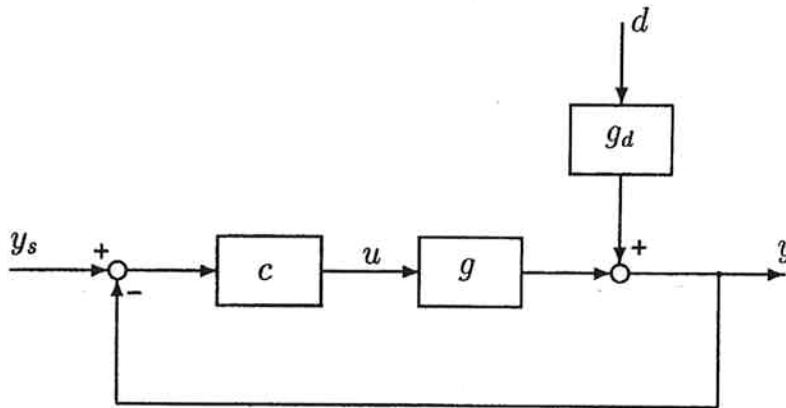


Figure 1. Block diagram of feedback control system. In this paper we consider an input ('load') disturbance ($g_d = g$).

for processes with a dominant delay. On the other hand, the analytically derived IMC-settings in Rivera *et al.* (1986) are known to result in a poor disturbance response for integrating processes (e.g. Chien & Fruehauf, 1990; Horn *et al.*, 1996), but are robust and generally give very good responses for setpoint changes. The single tuning rule presented in this paper works well for both integrating and pure time delay processes, and for both setpoints and load disturbances.

1.1. Notation

The notation is summarized in Figure 1, where u is the manipulated input (controller output), d the disturbance, y the controlled output, and y_s the setpoint (reference) for the controlled output. $g(s) = (\Delta y / \Delta u)$ denotes the process transfer function and $c(s)$ is the feedback part of the controller. The Δ used to indicate deviation variables is deleted in the following. The Laplace variable s is often omitted to simplify notation. The settings given in this paper are for the series (cascade, 'interacting') form PID controller:

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

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 equation (36).

1.2. Simulations

The following series form PID controller is used in all simulations and evaluations of performance:

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

with $\tau_F = \alpha \tau_D$ and $\alpha = 0.01$ (the robustness margins have been computed with $\alpha = 0$). Note that we, in order to avoid ‘derivative kick’ do not differentiate the setpoint in equation (2). The value $\alpha = 0.01$ was chosen in order to not bias the results, but in practice (and especially for noisy processes) a larger value of α in the range 0.1–0.2 is normally used. 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. In the time domain the PI-controller becomes

$$u(t) = u_0 + K_c \left((by_s(t) - y(t)) + \frac{1}{\tau_I} \int_0^t (y_s(\tau) - y(\tau)) d\tau \right) \quad (3)$$

where we have used $b = 1$ for the proportional setpoint weight.

2. Model approximation (Step 1)

The first step in the proposed design procedure is to obtain from the original model $g_0(s)$ an approximate first- or second-order time delay model $g(s)$ in the form

$$\begin{aligned} g(s) &= \frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)} e^{-\theta s} \\ &= \frac{k'}{(s + 1/\tau_1)(\tau_2 s + 1)} e^{-\theta s} \end{aligned} \quad (4)$$

Thus, we need to estimate the following model information (see Figure 2):

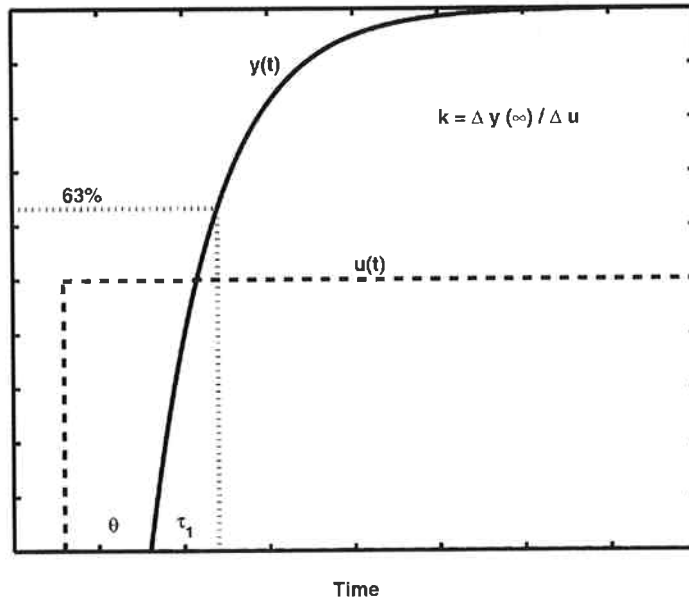


Figure 2. Step response of first-order plus time delay process, $g(s) = ke^{-\theta s}/(\tau_1 s + 1)$.

- 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)

If the response is lag-dominant, i.e. if $\tau_1 > 8\theta$ approximately, then the individual values of the time constant τ_1 and the gain k may be difficult to obtain, but at the same time are not very important for controller design. Lag-dominant processes may instead be approximated by an integrating process, using

$$\frac{k}{\tau_1 s + 1} \approx \frac{k}{\tau_1 s} = \frac{k'}{s} \quad (5)$$

which is exact when $\tau_1 \rightarrow \infty$ or $1/\tau_1 \rightarrow 0$. In this case we need to obtain the value for the

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

The problem of obtaining the effective delay θ (as well as the other model parameters) can be set up as a parameter estimation problem, for example, by making a least squares approximation of the open-loop step response. However, our goal is to use the resulting effective delay to obtain controller settings, so a better approach would be to find the approximation which for a given tuning method results in the best closed-loop response (here 'best' could, for example, be to minimize the integrated absolute error (IAE) with a specified value for the sensitivity peak, M_s). However, our main objective is not 'optimality' but 'simplicity', so we propose a much simpler approach as outlined next.

2.1. Approximation of effective delay using the half rule

We first consider the control-relevant approximation of the fast dynamic modes (high-frequency plant dynamics) by use of an effective delay. To derive these approximations, consider the following two first-order Taylor approximations of a time delay transfer function:

$$e^{-\theta s} \approx 1 - \theta s \quad \text{and} \quad e^{-\theta s} = \frac{1}{e^{\theta s}} \approx \frac{1}{1 + \theta s} \quad (6)$$

From equation (6) we see that an 'inverse response time constant' T_0^{inv} (negative numerator time constant) may be approximated as a time delay:

$$(-T_0^{\text{inv}} s + 1) \approx e^{-T_0^{\text{inv}} s} \quad (7)$$

This is reasonable since an inverse response has a deteriorating effect on control similar to that of a time delay (e.g. Skogestad & Postlethwaite, 1996). Similarly, from equation (6) a (small) lag time constant τ_0 may be approximated as a time delay:

$$\frac{1}{\tau_0 s + 1} \approx e^{-\tau_0 s} \quad (8)$$

Furthermore, since

$$\frac{-T_0^{\text{inv}} s + 1}{\tau_0 s + 1} e^{-\theta_0 s} \approx e^{-\theta_0 s} \cdot e^{-T_0^{\text{inv}} s} \cdot e^{-\tau_0 s} = e^{-(\theta_0 + T_0^{\text{inv}} + \tau_0)s} = e^{-\theta s}$$

it follows that the effective delay θ can be taken as the sum of the original delay θ_0 , and the contribution from the various approximated terms. In addition, for digital implementation with sampling period h , the contribution to the effective delay is approximately $h/2$ (which is the average time it takes for the controller to respond to a change).

In terms of control, the lag-approximation equation (8) is conservative, since the effect of a delay on control performance is worse than that of a lag of equal magnitude (e.g. Skogestad & Postlethwaite, 1996). In particular, this applies when approximating the largest of the neglected lags. Thus, to be less conservative it is recommended to use the simple *half rule*:

- **Half rule:** The largest neglected (denominator) time constant (lag) is distributed evenly to the effective delay and the smallest retained time constant.

In summary, let the original model be in the form

$$\frac{\prod_j (-T_{j0}^{inv}s + 1)}{\prod_i \tau_{i0}s + 1} e^{-\theta_0 s} \quad (9)$$

where the lags τ_{i0} are ordered according to their magnitude, and $T_{j0}^{inv} > 0$ denote the inverse response (negative numerator) time constants. Then, according to the half-rule, to obtain a first-order model $e^{-\theta s}/(\tau_1 s + 1)$, we use

$$\tau_1 = \tau_{10} + \frac{\tau_{20}}{2}, \quad \theta = \theta_0 + \frac{\tau_{20}}{2} + \sum_{i \geq 3} \tau_{i0} + \sum_j T_{j0}^{inv} + \frac{h}{2} \quad (10)$$

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

$$\tau_1 = \tau_{10}; \quad \tau_2 = \tau_{20} + \frac{\tau_{30}}{2}, \quad \theta = \theta_0 + \frac{\tau_{30}}{2} + \sum_{i \geq 4} \tau_{i0} + \sum_j T_{j0}^{inv} + \frac{h}{2} \quad (11)$$

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

The main basis for the empirical half-rule is to maintain the robustness of the proposed PI- and PID-tuning rules, as is justified by the examples later.

Example E1. 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 $\tau_1 = 1 + 0.2/2 = 1.1$.

2.2. Approximation of positive numerator time constants

We next consider how to get a model in the form of equation (9), if we have positive numerator time constants I_0 in the original model $g_0(s)$. It is proposed to cancel the numerator term $(T_0 s + 1)$ against a ‘neighbouring’ denominator term $(\tau_0 s + 1)$ (where both T_0 and τ_0 are positive and real) using the following approximations:

$$\frac{T_0 s + 1}{\tau_0 s + 1} \approx \begin{cases} T_0/\tau_0 & \text{for } T_0 \geq \tau_0 \geq \theta & \text{(Rule T1)} \\ T_0/\theta & \text{for } T_0 \geq \theta \geq \tau_0 & \text{(Rule T1a)} \\ 1 & \text{for } \theta \geq T_0 \geq \tau_0 & \text{(Rule T1b)} \\ T_0/\tau_0 & \text{for } \tau_0 \geq T_0 \geq 5\theta & \text{(Rule T2)} \\ \frac{(\tilde{\tau}_0/\tau_0)}{(\tilde{\tau}_0 - T_0)s + 1} & \text{for } \tilde{\tau}_0 \stackrel{\text{def}}{=} \min(\tau_0, 5\theta) \geq T_0 & \text{(Rule T3)} \end{cases} \quad (12)$$

Here θ is the (final) effective delay, which exact value depends on the subsequent approximation of the time constants (half rule), so one may need to guess θ and iterate. If there is more than one positive numerator time constant, then one should approximate one T_0 at a time, starting with the largest T_0 .

We normally select τ_0 as the closest *larger* denominator time constant ($\tau_0 > T_0$) and use Rules T2 or T3. The exception is if there exists no larger τ_0 , or if there is smaller denominator time constant 'close to' T_0 , in which case 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 τ_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 the Appendix.

Example E3. For the process (Example 4 in Astrom *et al.*, 1998)

$$g_0(s) = \frac{2(15s + 1)}{(20s + 1)(s + 1)(0.1s + 1)^2} \quad (13)$$

we first introduce from Rule T2 the approximation

$$\frac{15s + 1}{20s + 1} \approx \frac{15s}{20s} = 0.75$$

(Rule T2 applies since $T_0 = 15$ is larger than 5θ , where θ is computed below.) Using the half rule, the process may then be approximated as a first-order time delay model with

$$k = 2 \cdot 0.75 = 1.5; \quad \theta = \frac{0.1}{2} + 0.1 = 0.15; \quad \tau_1 = 1 + \frac{0.1}{2} = 1.05$$

or as a second-order time delay model with

$$k = 1.5; \quad \theta = \frac{0.1}{2} = 0.05; \quad \tau_1 = 1; \quad \tau_2 = 0.1 + \frac{0.1}{2} = 0.15$$

Derivation of PID tuning rules (Step 2)

Direct synthesis (IMC tuning) for setpoints

Next, we derive for the model in equation (4) PI-settings or PID-settings using the method of direct synthesis for setpoints (Smith & 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

$$\frac{y}{y_s} = \frac{g(s)c(s)}{g(s)c(s) + 1} \quad (14)$$

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 equation (14) we get

$$c(s) = \frac{1}{g(s)} \frac{1}{(y/y_s)_{\text{desired}} - 1} \quad (15)$$

We here consider the second-order time delay model $g(s)$ in equation (4), and specify that we, following the delay, desire a simple first-order response with time constant τ_c (Rivera *et al.*, 1986; Smith & Corripio, 1985):

$$\left(\frac{y}{y_s}\right)_{\text{desired}} = \frac{1}{\tau_c s + 1} e^{-\theta s} \quad (16)$$

We have kept the delay θ in the ‘desired’ response because it is unavoidable. Substituting equations (16) and (4) into equation (15) gives a ‘Smith Predictor’ controller (Smith, 1957):

$$c(s) = \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{k} \frac{1}{(\tau_c s + 1 - e^{-\theta s})} \quad (17)$$

τ_c is the desired closed-loop time constant, and is the sole tuning parameter for the controller. Our objective is to derive PID settings, and to this effect we introduce in equation (17) a first-order Taylor series approximation of the delay, $e^{-\theta s} \approx 1 - \theta s$. This gives

$$c(s) = \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{k} \frac{1}{(\tau_c + \theta)s} \quad (18)$$

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

$$K_c = \frac{1}{k} \frac{\tau_1}{\tau_c + \theta} = \frac{1}{k} \frac{1}{(\tau_c + \theta)}; \quad \tau_I = \tau_1; \quad \tau_D = \tau_2 \quad (19)$$

3.2. Modifying the integral time for improved disturbance rejection

The PID-settings in equation (19) were derived by considering the *setpoint* response, and the result was that we should effectively cancel the first order dynamics of the process by selecting the integral time $\tau_I = \tau_1$. This is a robust setting which results in very good responses to setpoints and to disturbances entering directly at the process *output*. However, it is well known that for lag dominant processes with $\tau_1 \gg \theta$ (e.g. an integrating processes), the choice $\tau_I = \tau_1$ results in a long settling time for *input* (‘load’) disturbances (Chien & Fruehauf, 1990). To improve the load disturbance response we need to reduce the integral time, but not by too much, because otherwise we get slow oscillations caused by having almost have two integrators in series (one from the controller and almost one from the slow lag dynamics in the process). This is illustrated in Figure 3, where we for the process

$$e^{-\theta s}/(\tau_1 s + 1) \quad \text{with} \quad \tau_1 = 30, \theta = 1$$

consider PI-control with $K_c = 15$ and four different values of the integral time:

- $\tau_I = \tau_1 = 30$ (‘IMC-rule’, see equation (19): excellent setpoint response, but slow settling for a load disturbance.

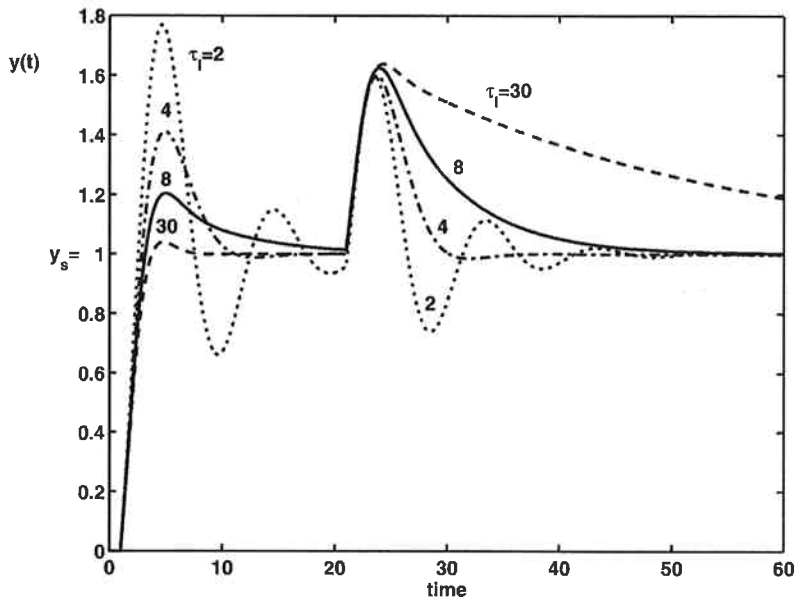


Figure 3. Effect of changing the integral time τ_I for PI-control of 'almost integrating' process $g(s) = e^{-s}/(30s + 1)$ with $K_c = 15$. Unit setpoint change at $t = 0$; load disturbance of magnitude 10 at $t = 20$.

- $\tau_I = 8\theta = 8$ (SIMC-rule, see below): faster settling for a load disturbance.
- $\tau_I = 4$: even faster settling, but the setpoint response (and robustness) is poorer.
- $\tau_I = 2$: poor response with 'slow' oscillations.

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 corresponds to $\tau_I = 8\theta$ in the above example. Let us analyze this in more detail. First, note that these 'slow' oscillations are *not* caused by the delay θ (and occur at a lower frequency than the 'usual fast' oscillations which occur at about frequency $1/\theta$). Because of this, we neglect the delay in the model when we analyze the slow oscillations. The process model then becomes

$$g(s) = k \frac{e^{-\theta s}}{\tau_1 s + 1} \approx k \frac{1}{\tau_1 s + 1} \approx \frac{k}{\tau_1 s} = \frac{k'}{s}$$

where the second approximation applies since the resulting frequency of oscillations ω_0 is such that $(\tau_1 \omega_0)^2$ is much larger than 1.¹ With a PI controller $c = K_c(1 + (1/\tau_I)s)$ the closed-loop characteristic polynomial $1 + gc$ then becomes

$$\frac{\tau_I}{k'K_c} s^2 + \tau_I s + 1$$

which is in standard second-order form, $\tau_0^2 s^2 + 2\tau_0 \zeta s + 1$, with

¹From equations (20) and (22) we get $\tau_0 = \tau_I/2$, so $\omega_0 \tau_1 = (1/\tau_0)\tau_1 = 2(\tau_1/\tau_I)$. Here $\tau_1 \geq \tau_I$ and it follows that $\omega_0 \tau_1 \geq 1$.

$$\tau_0 = \sqrt{\frac{\tau_1}{k'K_c}}; \quad \zeta = \frac{1}{2}\sqrt{k'K_c\tau_1} \quad (20)$$

Oscillations occur for $\zeta < 1$. Of course, some oscillations may be tolerated, but a robust choice is to have $\zeta = 1$ (see also Marlin (1995), p. 588), or equivalently

$$K_c\tau_1 = 4/k' \quad (21)$$

Inserting the recommended value for K_c from equation (19) then gives the following modified integral time for processes where the choice $\tau_1 = \tau_1$ is too large:

$$\tau_1 = 4(\tau_c + \theta) \quad (22)$$

3.3. SIMC-PID tuning rules

To summarize, the recommended SIMC PID settings² for the second-order time delay process in equation (4) are³

$$K_c = \frac{1}{k} \frac{\tau_1}{\tau_c + \theta} = \frac{1}{k'} \frac{1}{\tau_c + \theta} \quad (23)$$

$$\tau_I = \min \{ \tau_1, 4(\tau_c + \theta) \} \quad (24)$$

$$\tau_D = \tau_2 \quad (25)$$

Here the desired first-order closed-loop response time τ_c is the only tuning parameter. Note that the same rules are used both for PI- and PID-settings, but the actual settings will differ. To get a PI-controller we start from a first-order model (with $\tau_2 = 0$), and to get a PID-controller we start from a second-order model. PID-control (with derivative action) is primarily recommended for processes with dominant second order dynamics (with $\tau_2 > \theta$, approximately), and 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 settings for a few special cases, including the pure time delay process, integrating process, and double integrating process. For the double integrating process, we let $\tau_2 \rightarrow \infty$ and introduce $k'' = k'/\tau_2$ and find (after some algebra) that the PID-controller for the integrating process with lag approaches a PD-controller with

$$K_c = \frac{1}{k''} \cdot \frac{1}{4(\tau_c + \theta)^2}; \quad \tau_D = 4(\tau_c + \theta) \quad (26)$$

This controller gives good setpoint responses for the double integrating process, but results in steady-state offset for load disturbances occurring at the input. To remove this offset, we need to reintroduce integral action, and as before propose to use

$$\tau_I = 4(\tau_c + \theta) \quad (27)$$

It should be noted that derivative action is required to stabilize a double integrating process if we have integral action in the controller.

²Here SIMC means 'Simple control' or 'Skogestad IMC'.

³The derivative time in equation (25) is for the series form PID-controller in equation (1).

Table 1. SIMC PID-settings (equations (23)–(25)) for some special cases of equation (4) (with τ_c as a tuning parameter)

Process	$g(s)$	K_c	τ_1	τ_D^d
First-order	$k \frac{e^{-\theta s}}{(\tau_1 s + 1)}$	$\frac{1}{k} \frac{\tau_1}{\tau_c + \theta}$	$\min\{\tau_1, 4(\tau_c + \theta)\}$	—
Second-order, equation (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 ^a	$ke^{-\theta s}$	0	0 ^e	—
Integrating ^b	$k' \frac{e^{-\theta s}}{s}$	$\frac{1}{k'} \cdot \frac{1}{(\tau_c + \theta)}$	$4(\tau_c + \theta)$	—
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 ^c	$k'' \frac{e^{-\theta s}}{s^2}$	$\frac{1}{k''} \cdot \frac{1}{4(\tau_c + \theta)^2}$	$4(\tau_c + \theta)$	$4(\tau_c + \theta)$

^a The pure time delay process is a special case of a first-order process with $\tau_1 = 0$.

^b The integrating process is a special case of a first-order process with $\tau_1 \rightarrow \infty$.

^c For the double integrating process, integral action has been added according to equation (1).

^d The derivative time is for the series form PID controller in equation (27).

^e Pure integral controller $c(s) = \frac{K_I}{s}$ with $K_I \stackrel{\text{def}}{=} \frac{K_c}{\tau_1} = \frac{1}{k(\tau_c + \theta)}$.

3.4. Recommended choice for tuning parameter τ_c

The value of the desired closed-loop time constant τ_c can be chosen freely, but from equation (23) 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. Fast speed of response and good disturbance rejection (favored by a small value of τ_c)
2. Stability, robustness and small input variation (favored by a large value of τ_c).

A good trade-off is obtained by choosing τ_c equal to the time delay:

$$\text{SIMC-rule for fast response with good robustness: } \tau_c = \theta \quad (28)$$

This gives a reasonably fast response with moderate input usage and good robustness margins, and for the second-order time delay process in equation (4) results in the following SIMC-PID settings which may be easily memorized ($\tau_c = \theta$):

$$K_c = \frac{0.5}{k} \frac{\tau_1}{\theta} = \frac{0.5}{k'} \frac{1}{\theta} \quad (29)$$

$$\tau_1 = \min\{\tau_1, 8\theta\} \quad (30)$$

$$\tau_D = \tau_2 \quad (31)$$

The corresponding settings for the ideal PID-controller are given in equations (37) and (38).

Table 2. Robustness margins for first-order and integrating time delay process using the SIMC-settings in equations (29) and (30) ($\tau_c = \theta$)

Process $g(s)$	$\frac{k}{\tau_1 s + 1} e^{-\theta s}$	$\frac{k'}{s} e^{-\theta s}$
Controller gain, K_c	$\frac{0.5 \tau_1}{k \theta}$	$\frac{0.5}{k' \theta}$
Integral time, τ_I	τ_1	8θ
Gain margin (GM)	3.14	2.96
Phase margin (PM)	61.4°	46.9°
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
Allowed time delay error, $\Delta\theta/\theta$	2.14	1.59

The same margins apply to a second-order process (equation (4)) if we choose $\tau_D = \tau_2$, see equation (31).

4. Evaluation of the proposed tuning rules

In this section we evaluate the proposed SIMC PID tuning rules in equations (23)–(31) with the choice $\tau_c = \theta$. We first consider processes that already are in the second-order plus delay form in equation (4). In Section 4.2 we consider more complicated processes which must first be approximated as second-order plus delay processes (Step 1), before applying the tuning rules (Step 2).

4.1. First- or second-order time delay processes

4.1.1. *Robustness* The robustness margins with the SIMC PID-settings in equations (29)–(31), when applied to first- or second-order time delay processes, are always between the values given by the two columns in Table 2.

For processes with $\tau_1 \leq 8\theta$, for which we use $\tau_I = \tau_1$ (left column), the system always has a gain margin GM = 3.14 and phase margin PM = 61.4°, which is much better than the typical minimum requirements GM > 1.7 and PM > 30° (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 $\Delta\theta/\theta = \text{PM}[\text{rad}]/(\omega_c \cdot \theta)$, 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 $(1 + 2.14)\theta = 3.14\theta$).

As expected, the robustness margins are somewhat poorer for lag-dominant processes with $\tau_1 > 8\theta$, where we in order to improve the disturbance response use $\tau_I = 8\theta$. Specifically, for the extreme case of an integrating process (right column) the suggested settings give GM = 2.96, PM = 46.9°, $M_s = 1.70$ and $M_t = 1.30$, and the maximum allowed time delay error is $\Delta\theta = 1.59\theta$.

Of the robustness measures listed above, we will in the following concentrate on M_s , which is the peak value as a function of frequency of the sensitivity function $S = 1/(1 + gc)$. Notice that $M_s < 1.7$ guarantees GM > 2.43 and PM > 34.2° (Rivera *et al.*, 1986).

4.1.2. *Performance.* To evaluate the closed-loop performance, we consider a unit step setpoint change ($y_s = 1$) and a unit step input (load) disturbance ($g_d = g$ and $d = 1$), and for each of the two consider the input and output performance:

4.1.2.1. *Output performance.* To evaluate the output control performance we compute the integrated absolute error (IAE) of the control error $e = y - y_s$.

$$\text{IAE} = \int_0^{\infty} |e(t)| dt$$

which should be as small as possible.

4.1.2.2. *Input performance.* To evaluate the manipulated input usage we compute the total variation (TV) of the input $u(t)$, which is sum of all its moves up and down. TV is a bit difficult to define compactly for a continuous signal, but if we discretize the input signal as a sequence, $[u_1, u_2, \dots, u_i, \dots]$, then

$$\text{TV} = \sum_{i=1}^{\infty} |u_{i+1} - u_i|$$

which should be as small as possible. The total variation is a good measure of the 'smoothness' of a signal.

In Table 3 we summarize the results with the choice $\tau_c = \theta$ for the following five first-order time delay processes:

- Case 1. Pure time delay process
- Case 2. Integrating process
- Case 3. Integrating process with lag $\tau_2 = 4\theta$
- Case 4. Double integrating process
- Case 5. First-order process with $\tau_1 = 4\theta$

Note that the robustness margins fall within the limits given in Table 2, except for the double integrating process in case 4 where we, from equation (27), have added integral action and robustness is somewhat poorer.

4.1.2.3. *Setpoint change.* The simulated time responses for the five cases are shown in Figure 4. The setpoint responses are nice and smooth. For a unit setpoint change, the minimum achievable IAE-value for these time delay processes is $\text{IAE} = \theta$ (e.g. using a Smith Predictor controller equation (17) with $\tau_c = 0$). From Table 3 we see that with the proposed settings the actual IAE-setpoint-value varies between 2.17θ (for the first-order process) to 7.92θ (for the more difficult double integrating process).

To avoid 'derivative kick' on the input, we have chosen to follow industry practice and not differentiate the setpoint, see equation (2). This is the reason for the difference in the setpoint responses between cases 2 and 3, and also the reason for the somewhat sluggish setpoint response for the double integrating process in case 4. Note also that the setpoint response can always be modified by introducing a 'feedforward' filter on the setpoint or using $b \neq 1$ in equation (3).

4.1.2.4. *Load disturbance.* The load disturbance responses in Figure 4 are also nice and smooth, although a bit sluggish for the integrating and double integrating processes. In the last column in Table 3 we compare the achieved IAE-value with

Table 3. SIMC settings and performance summary for five different time delay processes ($\tau_c = \theta$)

Case	$g(s)$	K_c	τ_1	τ_D^c	M_s	Setpoint ^a			Load disturbance		
						IAE(y)	TV(u)	IAE(y)	TV(u)	IAE	IAEmin
1	$ke^{-\theta s}$	0	0^d	—	1.59	2.17 θ	$1.08 \frac{1}{k}$	$2.17k\theta$	1.08	1.59	
2	$k' \frac{e^{-\theta s}}{s}$	$\frac{0.5}{k'} \cdot \frac{1}{\theta}$	8 θ	—	1.70	3.92 θ	$1.22 \frac{1}{k'\theta}$	$16k'\theta^2$	1.55	3.27	
3	$k' \frac{e^{-\theta s}}{s(4\theta s+1)}$	$\frac{0.5}{k'} \cdot \frac{1}{\theta}$	8 θ	$\tau_2=4\theta$	1.70	5.28 θ	$1.23 \frac{1}{k'\theta}$	$16k'\theta^2$	1.59	5.41	
4	$k'' \frac{e^{-\theta s}}{s^2}$	$\frac{0.0625}{k''} \cdot \frac{1}{\theta^2}$	8 θ	8 θ	1.96	7.92 θ	$0.205 \frac{1}{k''\theta^2}$	$128k''\theta^3$	2.34	5.49	
5	$k \frac{e^{-\theta s}}{4\theta s+1}$	$\frac{0.5}{k} \frac{\tau_1}{\theta} = \frac{2}{k}$	$\tau_1=4\theta$	—	1.59	2.17 θ	$4.11 \frac{1}{k}$	$2k\theta$	1.08	2.41	

^aThe IAE and TV-values for PID control are without derivative action on the setpoint.

^bIAEmin is for the IAE-optimal PI/PID-controller of the same kind.

^cThe derivative time is for the series form PID controller in equation (1).

^dPure integral controller $c(s) = \frac{K_I}{s}$ with $K_I = \frac{K_c}{\tau_I} = \frac{0.5}{k\theta}$.

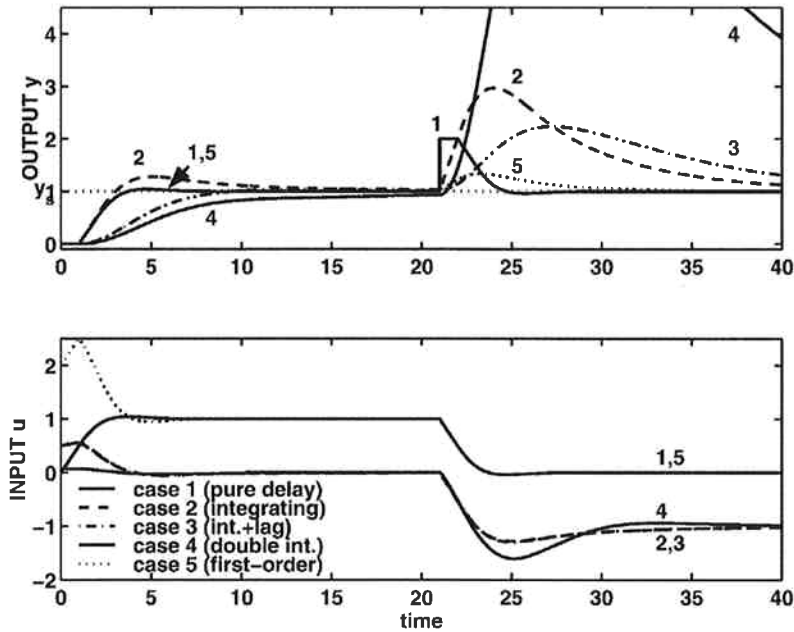


Figure 4. Responses using SIMC settings for the five time delay processes in Table 3 ($\tau_c = \theta$). Unit setpoint change at $t=0$; Unit load disturbance at $t=20$. Simulations are without derivative action on the setpoint. Parameter values: $\theta = 1$, $k = 1$, $k' = 1$, $k'' = 1$.

that for the IAE-optimal controller of the same kind (PI or series-PID). The ratio varies from 1.59 for the pure time delay process to 5.49 for the more difficult double integrating process.

However, lower IAE-values generally come at the expense of poorer robustness (larger value of M_s), more excessive input usage (larger value of TV), or a more complicated controller. For example, for the integrating process, the IAE-optimal PI-controller ($K_c = (0.91/k') \cdot (1/\theta)$, $\tau_1 = 4.1\theta$) reduces IAE(load) by a factor 3.27, but the input variation increases from $TV = 1.55$ to $TV = 3.79$, and the sensitivity peak increases from $M_s = 1.70$ to $M_s = 3.71$. The IAE-optimal PID-controller ($K_c = (0.80/k') \cdot (1/\theta)$, $\tau_1 = 1.26\theta$, $\tau_D = 0.76\theta$) reduces IAE(load) by a factor 8.2 (to $IAE = 1.95k'\theta^2$), but this controller has $M_s = 4.1$ and $TV(\text{load}) = 5.34$. The lowest achievable IAE-value for the integrating process is for an ideal Smith Predictor controller equation (17) with $\tau_c = 0$, which reduces IAE(load) by a factor 32 (to $IAE = 0.5k'\theta^2$). However, this controller is unrealizable with infinite input usage and requires a perfect model.

4.1.2.5. *Input usage.* As seen from the simulations in the lower part of Figure 4 the input usage with the proposed settings is very smooth in all cases. To have no steady-state offset for a load disturbance, the minimum achievable value is $TV(\text{load}) = 1$ (smooth input change with no overshoot), and we find that the achieved value ranges from 1.08 (first-order process), through 1.55 (integrating process) and up to 2.34 (double integrating process).

4.2. More complex processes: obtaining the effective delay

We here consider some cases where we must first (Step 1) approximate the model as a first- or second-order plus delay process, before (Step 2) applying the proposed tuning rules.

In Table 4 we summarize for 15 different processes (E1–E15), the model approximation (Step 1), the SIMC-settings with $\tau_c = \theta$ (Step 2) and the resulting M_s -value, setpoint and load disturbance performance (IAE and TV). For most of the processes, both PI- and PID-settings are given. For some processes (E1, E12, E13, E14, E15) only first-order approximations are derived, and only PI-settings are given. The model approximations for cases E2, E3, E6 and E13 are studied separately; see equations (41), (13), (42) and (43). Processes E1 and E3–E8 have been studied by Astrom and coworkers (Astrom *et al.*, 1998; Hagglund & Astrom, in press), and in all cases the SIMC PI-settings and IAE-load-values in Table 4 are very similar to those obtained by Astrom and coworkers for similar values of M_s . Process E11 has been studied by Schei, 1994.

The peak sensitivity (M_s) for the 25 cases ranges from 1.23 to 2, with an average value of 1.64. This confirms that the simple approximation rules (including the half rule for the effective delay) are able to maintain the original robustness where M_s ranges from 1.59 to 1.70 (see Table 2). The poorest robustness with $M_s = 2$ is obtained for the two inverse response processes in E14 and E15. For these two processes, we also find that the input usage is large, with TV for a load disturbance larger than 3, whereas it for all other cases is less than 2 (the minimum value is 1). The inverse responses processes E14 and E15 are rather unusual in that the process gain remains finite (at 1) at high frequencies, and we also have that they give instability with PID control.

The input variation (TV) for a setpoint change is large in some cases, especially for cases where the controller gain K_c is large. In such cases the setpoint response may be slowed down by, for example, prefiltering the setpoint change or using b smaller than 1 in equation (3). (Alternatively, if input usage is not a concern, then prefiltering or use of $b > 1$ may be used to speed up the setpoint response.)

The last column in Table 4 gives for a load disturbance the ratio between the achieved IAE and the minimum IAE with the same kind of controller (PI or series-PID) with no robustness limitations imposed. In many cases this ratio is surprisingly small (e.g. less than 1.4 for the PI-settings for cases E2, E7, E9, E11 and E15). However, in most cases the ratio is larger, and even infinity (cases E1 and E6-PID). The largest values are for processes with little or no inherent control limitations (e.g. no time delay), such that theoretically very large controller gains may be used. In practice, this performance can not be achieved due to unmodeled dynamics and limitations on the input usage.

For example, for the second-order process $g(s) = (1/(s+1)(0.2s+1))$ (case E1) one may in theory achieve perfect control (IAE = 0) by using a sufficiently high controller gain. This is also why no SIMC PID-settings are given in Table 4 for this process, because the choice $\tau_c = \theta = 0$ gives infinite controller gain. More precisely, going back to equations (23) and (24), the SIMC-PID settings for process E1 are

$$K_c = \frac{1}{k} \frac{\tau_1}{\tau_c} = \frac{1}{\tau_c}; \quad \tau_1 = 4\tau_c; \quad \tau_D = \tau_2 = 0.1 \quad (32)$$

These settings give for any value of τ_c excellent robustness margins. In particular, for $\tau_c \rightarrow 0$ we get GM = ∞ , PM = 76.3°, $M_s = 1$, and $M_l = 1.15$. However, in this case

Table 4. Approximation $g(s) = k \frac{e^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)}$, SIMC PI/PID-settings ($\tau_c = \theta$) and performance summary for 15 processes

Case	Process model, $g_0(s)$	Approximation, $g(s)$							SIMC settings					Performance			
		k	θ	τ_1	τ_2	K_c	τ_1	τ_1^b	M_s	IAE(y)	TV(u)	IAE(y)	TV(u)	IAE	IAEmin		
E1 (PI)	$\frac{1}{(s+1)(0.2s+1)}$	1	0.1	1.1	—	5.5	0.8	—	1.56	0.36	12.7	0.15	1.55	∞			
E2 (PI)	$\frac{(-0.3s+1)(0.08s+1)}{(2s+1)(1s+1)(0.4s+1)(0.2s+1)(0.05s+1)^3}$	1	1.47	2.5	—	0.85	2.5	—	1.66	3.56	1.90	2.97	1.26	1.39			
E2 (PID)		1	0.77	2	1.2	1.30	2	1.2	1.73	2.73	2.84	1.54	1.33	1.99			
E3 (PI)	$\frac{2(15s+1)}{(20s+1)(s+1)(0.1s+1)^2}$	1.5	0.15	1.05	—	2.33	1.05	—	1.55	0.46	4.97	0.45	1.30	3.82			
E3 (PID)		1.5	0.05	1	0.15	6.67	0.4	0.15	1.47	0.25	15.0	0.068	1.45	64			
E4 (PI)	$\frac{1}{(s+1)^4}$	1	2.5	1.5	—	0.3	1.5	—	1.46	5.59	1.15	5.40	1.10	1.93			
E4 (PID)		1	1.5	1.5	1	0.5	1.5	1	1.43	4.31	1.27	3.13	1.12	3.49			
E5 (PI)	$\frac{1}{(s+1)(0.2s+1)(0.04s+1)(0.0008s+1)}$	1	0.148	1.1	—	3.72	1.1	—	1.59	0.45	8.17	0.30	1.41	4.1			
E5 (PID)		1	0.028	1.0	0.22	17.9	0.224	0.22	1.58	0.27	43.3	0.056	1.49	27			
E6 (PI)	$\frac{(0.17s+1)^2}{s(s+1)^2(0.028s+1)}$	1	1.69	^d	—	0.296	13.5	—	1.48	6.50	0.67	45.7	1.55	10.1			
E6 (PID)		1	0.358	^d	1.33	1.40	2.86	1.33	1.23	1.95	3.19	2.04	1.55	∞			
E7 (PI)	$\frac{-2s+1}{(s+1)^3}$	1	3.5	1.5	—	0.214	1.5	—	1.66	7.28	1.06	8.34	1.28	1.23			
E7 (PID)		1	2.5	1.5	1	0.3	1.5	1	1.85	5.99	1.02	6.23	1.57	1.22			

Table 4. Continued

E8 (PI)	$\frac{1}{s(s+1)^2}$	1	1.5	^d	—	0.33	12	—	1.76	6.47	0.84	36.4	1.78	3.2
E8 (PID)		1	0.5	^d	1.5	1.5	4	1.5	1.79	2.02	4.21	2.67	1.99	40
E9 (PI)	$\frac{e^{-s}}{(s+1)^2}$	1	1.5	1.5	—	0.5	1.5	—	1.61	3.38	1.31	3.14	1.15	1.34
E9 (PID)		1	1	1	1	0.5	1	1	1.59	3.03	1.29	2	1.10	1.60
E10 (PI)	$\frac{e^{-s}}{(20s+1)(2s+1)}$	1	2	21	—	5.25	16	—	1.72	6.34	12.3	3.05	1.49	2.9
E10 (PID)		1	1	20	2	10	8	2	1.65	4.32	22.8	0.80	1.37	4.9
E11 (PI)	$\frac{(-s+1)}{(6s+1)(2s+1)^2}e^{-s}$	1	5	7	—	0.7	7	—	1.63	11.5	1.59	10.1	1.20	1.37
E11 (PID)		1	3	6	3	1	6	3	1.66	9.09	2.11	6.03	1.24	1.86
E12 (PI)	$\frac{(6s+1)(3s+1)e^{-0.3s}}{(10s+1)(8s+1)(s+1)}$	0.225	0.3	1	—	7.41	1	—	1.66	1.07	18.3	0.15	1.39	2.1
E13 (PI)	$\frac{2s+1}{(10s+1)(0.5s+1)}e^{-s}$	0.625	1.25	4.5	—	2.88	4.50	—	1.74	2.86	6.56	1.61	1.20	3.39
E14 (PI)	$\frac{-s+1}{s}$	1	1	^d	—	0.5	8	—	2	3.59	2.04	17.3	3.40	3.75
E15 (PI)	$\frac{-s+1}{s+1}$	1	1	1	—	0.5	1	—	2	2	1.02	2.85	3.00	1.23

^aThe IAE- and TV-values for PID control are without derivative action on the setpoint.

^bIAE_{min} is for the IAE-optimal PI- or PID-controller.

^cThe derivative time is for the series form PID controller in equation (1).

^dIntegrating process, $g(s) = k' \frac{e^{-\theta s}}{s(\tau_2 s + 1)}$.

the good margins are misleading since the gain crossover frequency, $\omega_c \approx 1/\tau_c$, approaches infinity as τ_c goes to zero. Thus, the time delay error $\Delta\theta = PM/\omega_c$ that yields instability approaches zero (more precisely, $1.29\tau_c$) as τ_c goes to zero.

The recommendation given earlier was that a second-order model (and thus use of PID control with SIMC settings) should only be used for dominant second-order process with $\tau_2 > \theta$, approximately. This recommendation is justified by comparing for cases E1–E11 the results with PI-control and PID-control. We note from Table 4 that there is a close correlation between the value of τ_2/θ and the improvement in IAE for load changes. For example, τ_2/θ is infinite for case E1, and indeed the (theoretical) improvement with PID control over PI control is infinite. In cases E5, E6, E8, E3, E10 and E2 the ratio τ_2/θ is larger than 1 (ranges from 7.9 to 1.6), and there is a significant improvement in IAE with PID control (by a factor 242–1.9). In cases E11, E9, E4 and E7 the ratio τ_2/θ is less than 1 (ranges from 1 to 0.4) and the improvement with PID control is rather small (by a factor 1.6 to 1.3). This improvement is too small in most cases to justify the additional complexity and noise sensitivity of using derivative action.

In summary, these 15 examples illustrate that the simple SIMC tuning rules used in combination with the simple half-rule for estimating the effective delay, result in good and robust settings.

5. Comparison with other tuning methods

Above we have evaluated the proposed SIMC tuning approach on its own merit. A detailed and fair comparison with other tuning methods is virtually impossible—because there are many tuning methods, many possible performance criteria and many possible models. Nevertheless, we here perform a comparison for three typical processes; the integrating process with delay (Case 2), the pure time delay process (Case 1), and the fourth-order process E5 with distributed time constants. The following four tuning methods are used for comparison:

5.1. Original IMC PID tuning rules

In Rivera *et al.* (1986) PI and PID settings for various processes are derived. For a first-order time delay process their ‘improved IMC PI-settings’ for fast response ($\varepsilon = 1.7\theta$) are:

$$\text{IMC PI: } K_c = \frac{0.588}{k} \frac{\left(\tau_1 + \frac{\theta}{2}\right)}{\theta}; \quad \tau_I = \tau_1 + \frac{\theta}{2} \quad (33)$$

and the PID-settings for fast response ($\varepsilon = 0.8\theta$) are

$$\text{IMC series-PID: } K_c = \frac{0.769}{k} \frac{\tau_1}{\theta}; \quad \tau_I = \tau_1; \quad \tau_D = \frac{\theta}{2} \quad (34)$$

Note that these rules give $\tau_I \geq \tau_1$, so the response to input load disturbances will be poor for lag dominant processes with $\tau_1 \gg \theta$.

5.2. Astrom/Schei PID tuning (maximize K_I)

Schei (1994) argued that in process control applications we usually want a robust design with the highest possible attenuation of low-frequency disturbances, and

proposed to maximize the low-frequency controller gain $K_1 \stackrel{\text{def}}{=} (K_c/\tau_1)$ subject to given robustness constraints on the sensitivity peaks M_s and M_t . Both for PI- and PID-control, maximizing K_1 is equivalent to minimizing the integrated error (IE) for load disturbances, which for robust designs with no overshoot is the same as minimizing the integral absolute error (IAE) (Astrom & Hagglund, 1995). Note that the use of derivative action (τ_D) does not affect the IE (and also not the IAE for robust designs), but it may improve robustness (lower M_s) and reduce the input variation (lower TV—at least with no noise). Astrom *et al.* (1998) showed how to formulate the minimization of K_1 as an efficient optimization problem for the case with PI control and a constraint on M_s . The value of the tuning parameter M_s is typically between 1.4 (robust tuning) and 2 (more aggressive tuning). We will here select it to be the same as for the corresponding SIMC design, that is, typically around 1.7.

5.3. Ziegler–Nichols (ZN) PID tuning rules

In Ziegler & Nichols (1942) it was proposed as the first step to generate sustained oscillations with a P-controller, and from this obtain the ‘ultimate’ gain K_u and corresponding ‘ultimate’ period P_u (alternatively, this information can be obtained using relay feedback (Astrom & Hagglund, 1995)). Based on simulations, the following ‘closed-loop’ settings were recommended:

$$\text{P-control: } K_c = 0.5K_u$$

$$\text{PI-control: } K_c = 0.45K_u; \quad \tau_1 = P_u/1.2$$

$$\text{PID-control (series): } K_c = 0.3K_u; \quad \tau_1 = P_u/4; \quad \tau_D = P_u/4$$

Remark. We have here assumed that the PID-settings given by Ziegler & Nichols ($K'_c = 0.6K_u$, $\tau'_1 = P_u/2$, $\tau'_D = P_u/8$) were originally derived for the ideal form PID controller (see Hellem (2001) for justification), and have translated these into the corresponding series settings using equation (36). This gives somewhat less aggressive settings and better IAE-values than if we assume that the ZN-settings were originally derived for the series form. Note that K_c/τ_1 and $K_c\tau_D$ are not affected, so the difference is only at intermediate frequencies.

5.4. Tyreus–Luyben modified ZN PI tuning rules

The ZN settings are too aggressive for most process control applications, where oscillations and overshoot are usually not desired. This led Tyreus & Luyben (1992) to recommend the following PI-rules for more conservative tuning:

$$K_c = 0.313K_u; \quad \tau_1 = 2.2P_u$$

5.5. Integrating process

The results for the integrating process, $g(s) = k'(e^{-\theta s}/s)$, are shown in Table 5 and Figure 5. The SIMC-PI controller with $\tau_c = \theta$ yields $M_s = 1.7$ and $\text{IAE}(\text{load}) = 16$. The Astrom/Schei PI-settings for $M_s = 1.7$ are very similar to the SIMC settings, but with somewhat better load rejection (IAE reduced from 16 to 13). The ZN PI-controller has a shorter integral time and larger gain than the SIMC-controller, which results in much better load rejection with IAE reduced from 16 to 5.6. However,

Table 5. Tunings and performance for integrating process, $g(s) = k'e^{-\theta s}/s$

Method	$K_c \cdot k'\theta$	τ_I/θ	τ_D/θ^a	M_s	Setpoint ^b		Load disturbance	
					IAE(y)	TV(u)	IAE(y)	TV(u)
SIMC ($\tau_c = \theta$)	0.5	8	—	1.70	3.92	1.22	16.0	1.55
IMC ($\varepsilon = 1.7\theta$)	0.59	∞	—	1.75	2.14	1.32	∞	1.24
Astrom/Schei ($M_s = 1.7$)	0.404	7.0	—	1.70	4.56	1.16	13.0	1.88
ZN-PI	0.71	3.33	—	2.83	3.92	2.83	5.61	2.87
Tyreus–Luyben	0.49	7.32	—	1.70	3.95	1.21	14.9	1.59
ZN-PID	0.471	1	1	2.29	2.88	2.45	3.32	3.00

^aThe derivative time is for the series form PID controller in equation (1).

^bThe IAE- and TV-values for PID control are without derivative action on the setpoint.

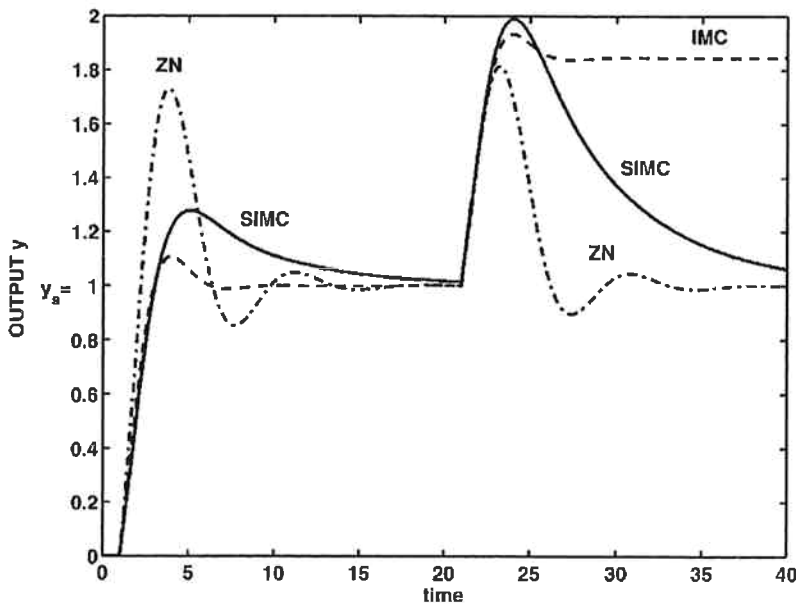


Figure 5. Responses for PI-control of integrating process, $g(s) = e^{-s}/s$, with settings from Table 5. Setpoint change at $t = 0$; load disturbance of magnitude 0.5 at $t = 20$.

the robustness is worse, with M_s increased from 1.70 to 2.83 and the gain margin reduced from 2.96 to 1.86. The IMC settings of Rivera *et al.* (1986) result in a pure P-controller with very good setpoint responses, but there is steady-state offset for load disturbances. The modified ZN PI-settings of Tyreus–Luyben are almost identical to the SIMC-settings. This is encouraging since it is exactly for this type of process that these settings were developed (Tyreus & Luyben, 1992).

5.6. Pure time delay process

The results for the pure time delay process, $g(s) = ke^{-\theta s}$, are given in Table 6 and Figure 6. Note that the setpoint and load disturbances responses are identical for this process, and also that the input and output signals are identical, except for the time delay.

Table 6. Tunings and performance for pure time delay process, $g(s) = ke^{-\theta s}$

Method	$K_c \cdot k$	$K_I \cdot k \cdot \theta^a$	τ_D / θ^c	M_s	Setpoint ^b		Load disturbance	
					IAE(y)	TV(u)	IAE(y)	TV(u)
SIMC ($\tau_c = \theta$)	0	0.5	—	1.59	2.17	1.08	2.17	1.08
IMC-PI ($\varepsilon = 1.7\theta$)	0.294	0.588	—	1.62	1.71	1.22	1.71	1.22
Astrom/Schei ($M_s = 1.6$)	0.200	0.629	—	1.60	1.59	1.08	1.59	1.08
Pessen	0.25	0.751	—	1.80	1.45	1.30	1.45	1.30
ZN-PI	0.45	0.27	—	1.85	3.70	1.53	3.70	1.53
Tyres-Luyben	0.313	0.071	—	1.46	14.1	1.22	14.1	1.22
IMC-PID ($\varepsilon = 0.8\theta$)	0	0.769	0.5	2.01	1.90	1.06	1.38	1.67
ZN-PID	0.3	0.6	0.5			Unstable		

^a $K_I = K_c / \tau_I$ is the integral controller gain.

^bThe derivative time is for the series form PID controller in equation (1).

^cThe IAE- and TV-values for PID control are without derivative action on the setpoint.

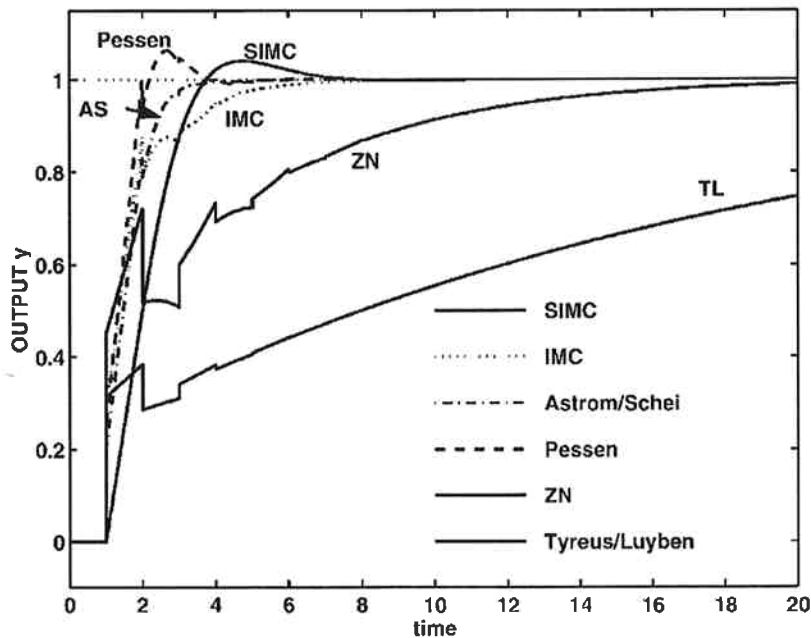


Figure 6. Setpoint responses for PI-control of pure time delay process, $g(s) = e^{-s}$, with settings from Table 6.

Recall that the SIMC-controller for this process is a pure integrating controller with $M_s = 1.59$ and IAE = 2.17. The minimum achievable IAE-value for any controller for this process is IAE = 1 (using a Smith Predictor equation (17) with $\tau_c = 0$). We find that the PI-settings using SIMC (IAE = 2.17), IMC (IAE = 1.71) and Astrom/Schei (IAE = 1.59) all yield very good performance. In particular, note that the excellent Astrom/Schei performance is achieved with good robustness ($M_s = 1.60$) and very smooth input usage (TV = 1.08). Pessen (1994) recommends PI-settings for the time delay process that give even better performance (IAE = 1.44), but with somewhat worse robustness ($M_s = 1.80$). The ZN PI-controller is significantly more

Table 7. Tuning and performance for process $g(s) = \frac{1}{(s+1)(0.2s+1)(0.04s+1)(0.008s+1)}$ (E5)

Method	K_c	τ_i	τ_D^a	M_s	Setpoint ^b		Load disturbance	
					IAE(y)	TV(u)	IAE(y)	TV(u)
SIMC-PI ($\tau_c = \theta$)	3.72	1.1	—	1.59	0.45	8.2	0.296	1.41
Astrom/Schei ($M_s = 1.6$)	2.74	0.67	—	1.60	0.58	6.2	0.246	1.52
ZN-PI	13.6	0.47	—	11.3	1.87	207	0.137	13.9
Tyreus-Luyben	9.46	1.24	—	2.72	0.50	35.8	0.131	2.91
SIMC-PID ($\tau_c = \theta$)	17.9	1.0	0.22	1.58	0.27	43.3	0.056	1.49
ZN-PID	9.1	0.14	0.14	2.39	0.24	39.2	0.025	3.09

^aThe derivative time is for the series form PID controller in Eq. (1)

^bThe IAE- and TV-values for PID control are without derivative action on the setpoint.

sluggish with IAE = 3.70, and the Tyreus-Luyben controller is extremely sluggish with IAE = 14.1. This is due to low value of the integral gain K_I .

Because the process gain remains constant at high frequency, any 'real' PID controller (with both proportional and derivative action), yields instability for this process, including the ZN PID-controller (Rivera *et al.*, 1986). (However, the IMC PID-controller is actually an ID-controller, and it yields a stable response with IAE = 1.38.)

The poor response with the ZN PI-controller and the instability with PID control, may partly explain the myth in the process industry that time delay processes cannot be adequately controlled using PID controllers. However, as seen from Table 6 and Figure 6, excellent performance can be achieved even with PI-control.

5.7. Fourth-order process (E5)

The results for the fourth-order process E5 (Astrom *et al.*, 1998) are shown in Table 7 and Figure 7. The SIMC PI-settings again give a smooth response (TV(load) = 1.41) with good robustness ($M_s = 1.59$) and acceptable disturbance rejection (IAE = 0.296). The Astrom/Schei PI-settings with $M_s = 1.6$ give very similar responses. IMC-settings are not given since no tuning rules are provided for models in this particular form (Rivera *et al.*, 1986). The Ziegler-Nichols PI-settings give better disturbance rejection (IAE = 0.137), but as seen in Figure 7 the system is close to instability. This is confirmed by the large sensitivity peak ($M_s = 11.3$) and excessive input variation (TV = 13.9) caused by the oscillations. The Tyreus-Luyben PI-settings give IAE = 0.131 and a much smoother response with TV = 2.91, but the robustness is still somewhat poor ($M_s = 2.72$). As expected, since this is a dominant second-order process, a significant improvement can be obtained with PID-control. As seen from Table 7 the performance of the SIMC PID-controller is not quite as good as the ZN PID-controller, but the robustness and input smoothness is much better.

Discussion

6.1. Detuning the controller

The above recommended SIMC settings with $\tau_c = \theta$, as well as almost all other PID tuning rules given in the literature, are derived to give a 'fast' closed-loop

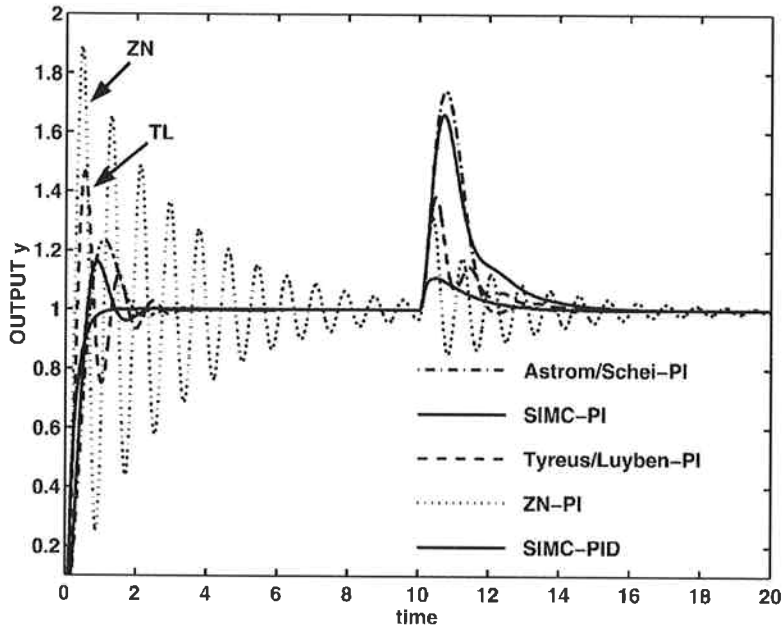


Figure 7. Responses for process $1/(s+1)(0.2s+1)(0.04s+1)(0.008s+1)$ (E5) with settings from Table 7. Setpoint change at $t=0$; load disturbance of magnitude 3 at $t=10$.

response subject to achieving reasonable robustness. However, in many practical cases we do need fast control, and to reduce the manipulated input usage, reduce measurement noise sensitivity and generally make operation smoother, we may want detune the controller. One main advantage of the SIMC tuning method is that detuning is easily done by selecting a larger value for τ_c . From the SIMC tuning rules, equations (23) and (24), a larger value of τ_c decreases the controller gain and, for lag-dominant processes with $\tau_1 > 4(\tau_c + \theta)$, increases the integral time. Fruehauf *et al.* (1994) state that in process control applications one typically chooses $\tau_c > 0.5$ min, except for flow control loops where one may have τ_c about 0.05 min.

6.2. Measurement noise

Measurement noise has not been considered in this paper, 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) with moderate input usage (small TV) 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 ZN-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 equation (2). With the proposed SIMC-settings one

can typically increase the filter time constant τ_F up to about 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 going back to equations (23) and (24) and selecting a larger value for τ_c .

6.3. Ideal form PID controller

The settings given in this paper (K_c , τ_1 , τ_D) are for the series (cascade, 'interacting') form PID controller in equation (1). To derive the corresponding settings for the ideal (parallel, 'non-interacting') form PID controller

$$\begin{aligned} c'(s) &= K_c' \left(1 + \frac{1}{\tau_1' s} + \tau_D' s \right) \\ &= \frac{K_c'}{\tau_1' s} (\tau_1' \tau_D' s^2 + \tau_1' s + 1) \end{aligned} \quad (35)$$

we use the following translation formulas

$$K_c' = K_c \left(1 + \frac{\tau_D}{\tau_1} \right); \quad \tau_1' = \tau_1 \left(1 + \frac{\tau_D}{\tau_1} \right); \quad \tau_D' = \frac{\tau_D}{1 + \frac{\tau_D}{\tau_1}} \quad (36)$$

The SIMC-PID series settings in equations (29)–(31) then correspond to the following SIMC ideal-PID settings ($\tau_c = \theta$):

$$\tau_1 \leq 8\theta: \quad K_c' = \frac{0.5}{k} \frac{(\tau_1 + \tau_2)}{\theta}; \quad \tau_1' = \tau_1 + \tau_2; \quad \tau_D' = \frac{\tau_2}{1 + \frac{\tau_2}{\tau_1}} \quad (37)$$

$$\tau_1 \geq 8\theta: \quad K_c' = \frac{0.5}{k} \frac{\tau_1}{\theta} \left(1 + \frac{\tau_2}{8\theta} \right); \quad \tau_1' = 8\theta + \tau_2; \quad \tau_D' = \frac{\tau_2}{1 + \frac{\tau_2}{8\theta}} \quad (38)$$

We see that the rules are much more complicated when we use the ideal form.

Example. Consider the second-order process $g(s) = e^{-s}/(s+1)^2$ (E9) with the $k=1$, $\theta=1$, $\tau_1=1$ and $\tau_2=1$. The series-form SIMC settings are $K_c=0.5$, $\tau_1=1$ and $\tau_D=1$. The corresponding settings for the ideal PID controller in equation (35) are $K_c'=1$, $\tau_1'=2$ and $\tau_D'=0.5$. The robustness margins with these settings are given by the first column in Table 2.

Remarks:

1. Use of the above formulas make the series and ideal controllers identical when considering the feedback controller, but they may differ when it comes to setpoint changes, because one usually does not differentiate the setpoint and the values for K_c differ.

2. The tuning parameters for the series and ideal forms are equal when the ratio between the derivative and integral time, τ_D/τ_I , approaches zero, that is, for a PI-controller ($\tau_D = 0$) or a PD-controller ($\tau_I = \infty$).
3. Note that it is *not* always possible to do the reverse and obtain series settings from the ideal settings. Specifically, this can only be done when $\tau_I' \geq 4\tau_D'$. This is because the ideal form is more general as it also allows for complex zeros in the controller. Two implications of this are:
 - (a) We should start directly with the ideal PID controller if we want to derive SIMC-settings for a second-order oscillatory process (with complex poles).
 - (b) Even for non-oscillatory processes, the ideal PID may give better performance due to its less restrictive form. For example, for the process $g(s) = 1/(\tau s + 1)^4$ (E4), the minimum achievable IAE for a load disturbance is IAE = 0.89 with a series-PID, and 40% lower (IAE = 0.52) with an ideal PID. The optimal settings for the ideal PID-controller ($K_c' = 4.96$, $\tau_I' = 1.25$, $\tau_D' = 1.84$) can not be represented by the series controller because $\tau_I' < 4\tau_D'$.

6.4. Retuning for integrating processes

Integrating processes are common in industry, but control performance is often poor because of incorrect settings. When encountering oscillations, the intuition of the operators is to reduce the controller gain. This is the exactly opposite of what one should do for an integrating process, since the product of the controller gain K_c and the integral time τ_I must be *larger* than the value in equation (22) in order to avoid slow oscillations. 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 $3 \times \tau_{i0}$, approximately). Then we likely have a close-to integrating process $g(s) = k'(e^{-\theta s}/s)$ for which the product of the controller gain and integral time ($K_{c0}\tau_{i0}$) is too low. From equation (20) we can estimate the damping coefficient ζ and time constant τ_0 associated with these oscillations of period P_0 , and a standard analysis of second-order systems (e.g. Seborg *et al.* (1989), p. 118) gives that the corresponding period is

$$P_0 = \frac{2\pi}{\sqrt{1-\zeta^2}} \tau_0 = \frac{2\pi}{\sqrt{1-\zeta^2}} \sqrt{\frac{\tau_{i0}}{k'K_{c0}}} \approx 2\pi \sqrt{\frac{\tau_{i0}}{k'K_{c0}}} \quad (39)$$

where we have assumed $\zeta^2 \ll 1$ (significant oscillations). Thus, from equation (39) the product of the original controller gain and integral time is approximately

$$K_{c0} \cdot \tau_{i0} = (2\pi)^2 \frac{1}{k'} \left(\frac{\tau_{i0}}{P_0} \right)^2$$

To avoid oscillations ($\zeta \geq 1$) with the new settings we must from equation (21) require $K_c \tau_I \geq 4/k'$, that is, we must require that

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

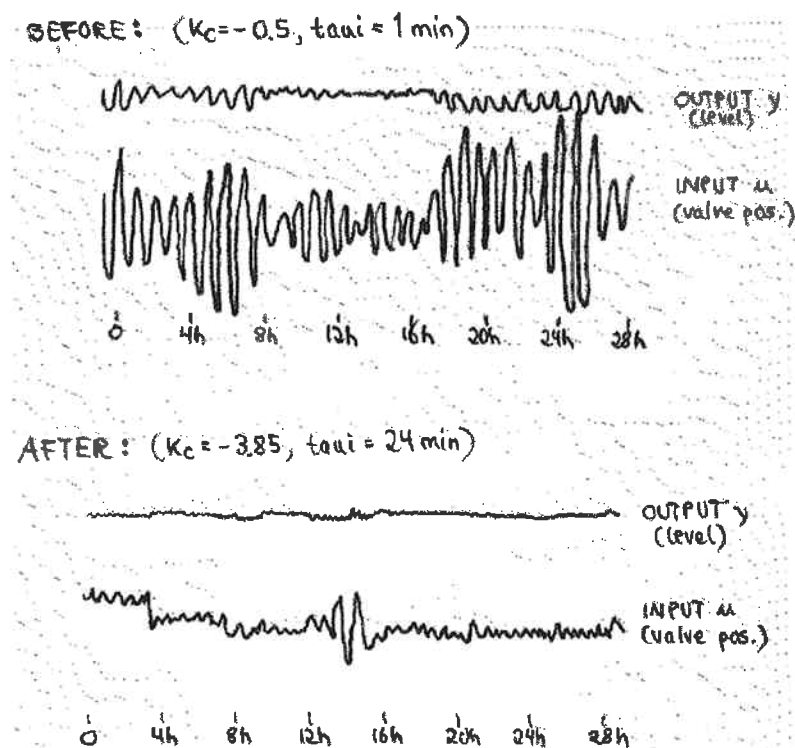


Figure 8. Industrial case study of retuning reboiler level control system.

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

- To avoid 'slow' oscillations of period P_0 the product of the controller gain and integral time should be increased by a factor $f \approx 0.1(P_0^2/\tau_{i0})^2$.

Example. This actual industrial case originated as a project to improve the purity control of a distillation column. It soon became clear that the main problem was large variations (disturbances) in its feed flow. The feed flow was again the bottoms flow from an upstream column, which was again set by its reboiler level controller. The control of the reboiler level itself was acceptable, but the bottoms flowrate showed large variations. This is shown in Figure 8, where y is the reboiler level and u is the bottoms flow valve position. The PI settings had been kept at their default setting ($K_c = -0.5$ and $\tau_{i1} = 1$ min) since start-up several years ago, and resulted in an oscillatory response as shown in the top part of Figure 8.

From a closer analysis of the 'before' response we find that the period of the slow oscillations is $P_0 = 0.85$ h = 51 min. Since $\tau_{i1} = 1$ min, we get from the above rule we should increase $K_c \cdot \tau_{i1}$ by a factor $f \approx 0.1 \cdot (51)^2 = 260$ to avoid the oscillations. The plant personnel were somewhat sceptical to authorize such large changes, but eventually accepted to increase K_c by a factor 7.7 and τ_{i1} by a factor 24, that is, $K_c \tau_{i1}$ was increased by $7.7 \cdot 24 = 185$. The much improved response is shown in the 'after' plot in Figure 8. There is still some minor oscillations, but these may be caused by disturbances outside the loop. In any case the control of the downstream distillation column was much improved.

6.5. Derivative action to counteract time delay?

Introduction of derivative action, e.g. $\tau_D = \theta/2$, is commonly proposed to improve the response when we have time delay (Rivera *et al.*, 1986; Smith & Corripio, 1985). To derive this value we may in equation (17) use the more exact 1st order Pade approximation, $e^{-\theta s} \approx (-\theta/2)s + 1)/((\theta/2)s + 1)$. With the choice $\tau_c = \theta$ this results in the same series-form PID-controller, equation (18), found above, but in addition we get a term $((\theta/2)s + 1)/(0.5(\theta/2)s + 1)$. This is as an additional derivative term with $\tau_D = \theta/2$, effective over only a small range, which increases the controller gain by a factor of two at high frequencies. However, with the robust SIMC settings used in this paper ($\tau_c = \theta$), the addition of derivative action (without changing K_c or τ_I) has in most cases *no effect on IAE* for load disturbances, since the integral gain $K_I = K_c/\tau_I$ is unchanged and there are no oscillations (Astrom & Hagglund, 1995). Although the robustness margins are somewhat improved (for example, for an integrating with delay process, $k's^{-\theta s}/s$, the value of M_s is reduced from 1.70 (PI) to 1.50 (PID) by adding derivative action with $\tau_D = \theta/2$), this probably does not justify the increased complexity of the controller and the increased sensitivity to measurement noise. This conclusion is further confirmed by Table 6 and Figure 6, where we found that a PI-controller (and even a pure I-controller) gave very good performance for a pure time delay process. In conclusion, it is *not* recommended to use derivative action to counteract time delay, at least not with the robust settings recommended in this paper.

6.6. Concluding remarks

- As illustrated by the many examples, the very simple analytic tuning procedure presented in this paper yields surprisingly good results. Additional examples and simulations are available in reports that are available over the Internet (Holm & Butler, 1998; Skogestad, 2001). The proposed analytic SIMC-settings are quite similar to the 'simplified IMC-PID tuning rules' of Fruehauf *et al.* (1994), which are based on extensive simulations and have been verified industrially. Importantly, the approach is analytic, which makes it very well suited for teaching and for gaining insight. Specifically, it gives invaluable insight into how the controller should be retuned in response to process changes, like changes in the time delay or gain.
- The approach has been developed for typical process control applications. Unstable processes have not been considered, with the exception of integrating processes. Oscillating processes (with complex poles or zeros) have also not been considered.
- 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 equations (23)–(25), 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 equations (10) and (11), 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 .
- The tuning method presented in this paper starts with a transfer function model of the process. If such a model is not known, then it is recommended to

use plant data, together with a regression package, to obtain a detailed transfer function model, which is then subsequently approximated as a model with effective delay using the proposed half-rule.

7. Conclusion

A two-step procedure is proposed for deriving PID-settings for typical process control applications.

1. The half rule is used to approximate the process as a first or second order model with effective delay θ , see equations (10) and (11).
2. For a first-order model (with parameters k , τ_1 and θ) the following SIMC PI-settings are suggested:

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

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

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

Note that although the same formulas are used to obtain K_c and τ_1 for both PI- and PID-control, the actual values will differ since the effective delay θ is smaller for a second-order model (PID). 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 (small TV) and robustness (small M_s) on the other side. The recommended choice of $\tau_c = \theta$ gives robust (M_s about 1.6 to 1.7) and somewhat conservative settings when compared with most other tuning rules.

Acknowledgements

Discussions with Professors David E. Clough, Dale Seborg and Karl J. Astrom are gratefully acknowledged.

Appendix: approximation of positive numerator time constants

In Figure 9 we consider four approximations of a real numerator term $(Ts + 1)$ where $T > 0$. In terms of the notation used in the rules presented earlier in the paper, these approximations correspond to

$$\text{Approximation 1: } \frac{(T_0s + 1)}{(\tau_0s + 1)} \approx T_0/\tau_0 \geq 1$$

$$\text{Approximation 2: } \frac{(T_0s + 1)}{(\tau_0s + 1)} \approx T_0/\tau_0 \leq 1$$

$$\text{Approximation 3: } \frac{(T_0s + 1)}{(\tau_0s + 1)} \approx \frac{1}{(\tau_0 - T_0)s + 1}$$

$$\text{Approximation 4: } \frac{(T_0s + 1)}{(\tau_{0a}s + 1)(\tau_{0b}s + 1)} \approx \frac{1}{\left(\frac{\tau_{0a}\tau_{0b}}{T_0}s + 1 \right)}$$

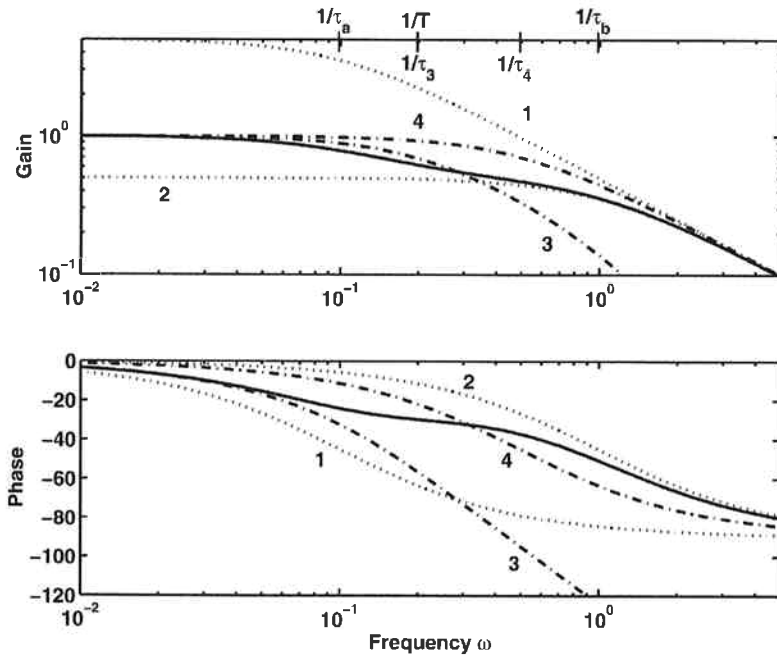


Figure 9. Comparison of $g_0(s) = \frac{(Ts+1)}{(\tau_a s+1)(\tau_b s+1)}$ with $\tau_a \geq T \geq \tau_b$ (solid line), with four

approximations (dashed and dotted lines): $g_1(s) = \frac{(T/\tau_b)}{(\tau_a s+1)}$, $g_2(s) = \frac{T/\tau_a}{(\tau_b s+1)}$,
 $g_3(s) = \frac{1}{(\tau_3 s+1)(\tau_b s+1)}$ with $\tau_3 = \tau_a - T$, and $g_4(s) = \frac{1}{(\tau_4 s+1)}$ with $\tau_4 = \frac{\tau_a \tau_b}{T}$.

For control purposes we have that

- Approximations that give a too high gain are 'safe' (as they will increase the resulting gain margin)
- Approximations that give too much negative phase are 'safe' (as they will increase the resulting phase margin)

and by considering Figure 9 and we have that

1. Approximation 1 (with $T_0 \geq \tau_0$) is always safe (both in gain and phase). It is good for frequencies $\omega > 1/\tau_0$.
2. Approximation 2 (with $T_0 \leq \tau_0$) is never safe (neither in gain or phase). It is good for $\omega > 5/T$.
3. Approximation 3 is good (and safe) for $\omega < 1/(\tau_0 - T_0)$. At high frequencies it is unsafe in gain.
4. Approximation 4 is good (and safe) for $\omega > 1/\tau_4 = T_0/(\tau_{0a}\tau_{0b})$. At low frequencies it is somewhat unsafe in phase.

'Good' here means that the resulting controller settings yield acceptable performance and robustness. Note that approximations 1 and 2 are asymptotically correct (and best) at high frequency, whereas approximation 3 is asymptotically correct (and best) at low frequency. Approximation 4 is asymptotically correct at both high and low frequencies.

Furthermore, for control purposes it is most critical to have a good approximation of the plant behavior at about the bandwidth frequency. For our model this is approximately at $\omega = 1/\theta$ where θ is the effective delay. From this we derive:

1. If T_0 is larger than all denominator time constant (τ_0) use Approximation 1 (this is the only approximation that applies in this case and it is always safe).
2. If $\tau_0 \geq T_0 \geq 5\theta$ use Approximation 2. (Approximation 2 is 'unsafe', but with $T_0 \geq 5\theta$ the resulting increase in M_s with the suggested SIMC-settings is less than about 0.3).
3. If the resulting $\tau_3 = \tau_0 - T_0$ is smaller than θ use Approximation 3.
4. If the resulting τ_4 is larger than θ use Approximation 4.

The first three approximations have been the basis for deriving the corresponding rules T1–T3 given in the paper. The rules have been verified by evaluating the resulting control performance when using the approximated model to derive SIMC PID settings. Some specific comments on the rules:

- Since the loss in accuracy when using Approximation 3 instead of Approximation 4 is minor, even for cases where Approximation 4 applies, it was decided to not include Approximation 4 in the final rules.
- Approximation 1,

$$\frac{(T_0s + 1)}{(\tau_0s + 1)} \approx k$$

where $k = (T_0/\tau_0) \geq 1$ is good for $\tau_0 \geq \theta$. It may be safely applied also when $\tau_0 < \theta$, but then gives conservative controller settings because the gain $k = T_0/\tau_0$ is too high at the important frequency $1/\theta$. This is the reason for the two modifications T1a and T1b to Approximation 1. For example, for the process

$$g_0(s) = \frac{2s + 1}{(0.2s + 1)^2} e^{-s}$$

Approximation 1 gives

$$\frac{k}{0.2s + 1} e^{-s}$$

with $k = T_0/\tau_0 = 10$. With $\tau_c = \theta = 1$ the SIMC-rules then yield $K_c = 0.01$ and $\tau_I = 0.2$ which gives a very sluggish response with $\text{IAE}(\text{load}) = 20$ and $M_s = 1.10$. With the modification $k = T_0/\theta = 2$ (Rule T1a), we get $K_c = 0.05$ which gives $\text{IAE}(\text{load}) = 4.99$ and $M_s = 1.84$ (which is close to the IAE-optimal PI-settings for this process).

- The introduction of $\tilde{\tau}_0$ instead of τ_0 in Rule T3, gives a smooth transition between Rules T2 and T3, and also improves the accuracy of Approximation 3 for the case when τ_0 is large.
- We normally select $\tau_0 = \tau_{0a}$ (large), except when τ_{0b} is 'close to T_0 '. Specifically, we select $\tau_0 = \tau_{0b}$ (small) if $T_0/\tau_{0b} < \tau_{0a}/T_0$ and $T_0/\tau_{0b} < 1.6$. The factor 1.6 is partly justified because $8\theta/5\theta = 1.6$, and we then in some important cases get a smooth transition when there are parameter changes in the model $g_0(s)$.

Example E2. For the process

$$g_0(s) = k \frac{(-0.3s + 1)(0.08s + 1)}{(2s + 1)(1s + 1)(0.4s + 1)(0.2s + 1)(0.05s + 1)^3} \quad (41)$$

we first introduce from Rule T3 the approximation

$$\frac{0.08s + 1}{0.2s + 1} \approx \frac{1}{0.12s + 1}$$

Using the half rule the process may then be approximated as a first-order delay process with

$$\theta = 1/2 + 0.4 + 0.12 + 3 \cdot 0.05 + 0.3 = 1.47; \quad \tau_1 = 2 + 1/2 = 2.5$$

or as a second-order delay process with

$$\theta = 0.4/2 + 0.12 + 3 \cdot 0.05 + 0.3 = 0.77; \quad \tau_1 = 2; \quad \tau_2 = 1 + 0.4/2 = 1.2$$

Remark. We here used $\tau_0 = \tau_{0a} = 0.2$ (the closest larger time constant) for the approximation of the zero at $T_0 = 0.08$. Actually, this is a borderline case with $T_0 / \tau_{0b} = 1.6$, and we could instead have used $\tau_0 = \tau_{0b} = 0.05$ (the closest smaller time constant). Approximation using Rule T1b would then give

$$\frac{0.08s + 1}{0.05s + 1} \approx 1$$

but the effect on the resulting models would be marginal: the resulting effective time delay θ would change from 1.47 to 1.50 (first-order process) and from 0.77 to 0.80 (second-order process), whereas the time constants (τ_1 and τ_2) and gain (k) would be unchanged.

Example E6. For the process (Example 6 in Astrom *et al.* (1998)),

$$g_0(s) = \frac{(0.17s + 1)}{s(s + 1)^2(0.028s + 1)} \tag{42}$$

we first introduce from Rule T3 the approximation

$$\frac{(0.17s + 1)^2}{(s + 1)} \approx \frac{1}{(1 - 0.17 - 0.17)s + 1} = \frac{1}{0.66s + 1}$$

Using the half rule we may then approximate equation (42) as an integrating process, $g(s) = k'e^{-\theta s}/s$, with

$$k' = 1; \quad \theta = 1 + 0.66 + 0.028 = 1.69$$

or as an integrating process with lag, $g(s) = k'e^{-\theta s}/s(\tau_2 s + 1)$, with

$$k' = 1; \quad \theta = 0.66/2 + 0.028 = 0.358; \quad \tau_2 = 1 + 0.66/2 = 1.33$$

Example E13. For the process

$$g_0(s) = \frac{2s + 1}{(10s + 1)(0.5s + 1)} e^{-s} \tag{43}$$

the effective delay is (as we will show) $\theta = 1.25$. We then get $\bar{\tau}_0 = \min(\tau_0, 5\theta) = \min(10, 6.25) = 6.25$, and from Rule T3 we have

$$\frac{2s + 1}{10s + 1} \approx \frac{(6.25/10)}{(6.25 - 2)s + 1} = \frac{0.625}{4.25s + 1}$$

Using the half rule we then get a first-order time delay approximation with

$$k = 0.625; \quad \theta = 1 + 0.5/2 = 1.25; \quad \tau_1 = 4.25 + 0.5/2 = 4.5$$

References

- ASTROM, K. J. & HAGGLUND, T. (1995). *PID Controllers: Theory, Design and Tuning*, 2nd Edition (Instrument Society of America, Research Triangle Park).
- ASTROM, K. J., PANAGOPOULOS, H. & HAGGLUND, T. (1998). Design of PI controllers based on non-convex optimization, *Automatica* 34(5): 585–601.
- CHIEN, I. L. & FRUEHAUF, P. S. (1990). Consider IMC tuning to improve controller performance, *Chemical Engineering Progress*, 33–41.
- FRUEHAUF, P. S., CHIEN, I. L. & LAURITSEN, M. D. (1994). Simplified IMC-PID tuning rules, *ISA Transactions*, 33, 43–59.
- HAGGLUND, T. & ASTROM, K. J. (in press). Revisiting the Ziegler–Nichols tuning rules for PI control, *Asian Journal of Control*.
- HELLEM, S. M. (2001). Evaluation of simple methods for tuning of PID-controllers. Technical report, 4th year project. Department of Chemical Engineering, Norwegian University of Science and Technology, Trondheim, <http://www.chemeng.ntnu.no/users/skoge/diplom/prosjekt01/hellem/>.
- HOLM, O. & BUTLER, A. (1998). Robustness and performance analysis of PI and PID controller tunings, Technical report, 4th year project. Department of Chemical Engineering, Norwegian University of Science and Technology, Trondheim, <http://www.chemeng.ntnu.no/users/skoge/diplom/prosjekt98/holm-butler/>.
- HORN, I. G., ARULANDU, J. R., GOMBAS, J., VANANTWERP, J. G. & BRAATZ, R. D. (1996). Improved filter design in internal model control, *Ind. Eng. Chem. Res.*, 35(10): 3437–3441.
- MARLIN, T. E. (1995). *Process Control* (New York: McGraw-Hill).
- PESSON, D. W. (1994). A new look at PID-controller tuning, *Trans. ASME (J. of Dyn. Systems, Meas. and Control)*, 116: 553–557.
- RIVERA, D. E., MORARI, M. & SKOGESTAD, S. (1986). Internal model control. 4. PID controller design, *Ind. Eng. Chem. Res.*, 25(1): 252–265.
- SCHEI, T. S. (1994). Automatic tuning of PID controllers based on transfer function estimation, *Automatica*, 30(12): 1983–1989.
- SEBORG, D. E., EDGAR, T. F. & MELLICHAMP, D. A. (1989). *Process Dynamics and Control* (New York: John Wiley & Sons).
- SKOGESTAD, S. (2001). Probably the best simple PID tuning rules in the world. AIChE Annual Meeting, Reno, Nevada, November, <http://www.chemeng.ntnu.no/users/skoge/publications/2001/tuningpaper-reno>
- SKOGESTAD, S. & POSTLETHWAITE, I. (1996). *Multivariable Feedback Control* (Chichester: John Wiley & Sons).
- SMITH, O. J. (1957). Closer control of loops with dead time, *Chem. Eng. Prog.*, 53: 217.
- SMITH, C. A., CORRIPIO, A. B. (1985). *Principles and Practice of Automatic Process Control* (New York: John Wiley & Sons).
- TYREUS, B. D. & LUYBEN, W. L. (1992). Tuning PI controllers for integrator/dead time processes, *Ind. Eng. Chem. Res.*, 2628–2631.
- ZIEGLER, J. G. & NICHOLS, N. B. (1942). Optimum settings for automatic controllers, *Trans. ASME*, 64: 759–768.

May 2004: ADDITIONAL COMMENTS FROM THE AUTHOR

1. Model reduction

The main contribution in terms of model reduction is the simple *half rule* for the effective delay. It started in 1997 as a simple empirical rule that I used for teaching in my process control class; see the following handout from 1998 (in Norwegian): <http://www.nt.ntnu.no/users/skoge/publications/2003/tuningPID/more/1998>. Before 1997, when teaching the students about the ‘effective delay’, I had added the inverse time constant and *all* of the smaller (neglected) denominator time constants. However, this gives a conservative (too large) estimate of the delay when it is applied to common PID-rules, and gives rather sluggish tunings. The reason is that a time

constant is not quite ‘as bad’ as a delay. I experimented a bit and found that adding only half of the largest neglected time constant (and adding the rest to the smallest retained time constant) gave better tunings. The validity of the half rule was later confirmed by testing it on a large number of processes; see the paper and three student projects from 1998, 1999 and 2001 (see <http://www.nt.ntnu.no/users/skoge/publications/2003/tuningPID/more>). In summary, the main justification of the half rule in the context of PID-tuning is that it when combined with the SIMC PID-tuning rules consistently results in good tunings with M_s about 1.7 (with $\tau_c = \theta$). The half rule is reasonable but clearly not optimal in terms of open-loop model fitting (e.g. in the time or frequency domains). On the other hand, open-loop model fitting is not the objective, but rather that the resulting PID-tunings should work well. We have tried other values than a half (in the range from 0 to 1), e.g. see some of the student projects, but this did not improve the results.

Let me give some justification for the half rule in terms of tuning. Recall the tuning rules $K_c k = (\tau_1/\tau_c + \theta)$; $\tau_1 = \min\{4(\tau_c + \theta), \tau_1\}$ with the typical choice $\tau_c = \theta$. Consider an original second-order model on the form

$$g_0(s) = k \frac{e^{-\theta_0 s}}{(1 + \tau_{10}s)(1 + \tau_{20}s)}$$

which we want to approximate with a first-order model,

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

The ‘first part’ of the half rule is the effective delay. According to the half rule

$$\theta = \theta_0 + \frac{\tau_{20}}{2}$$

Compared with the conservative alternative of adding all of τ_{20} to the delay (i.e. $\theta = \theta_0 + \tau_{20}$), the half rule gives a shorter delay (up to a factor 2), which as expected results in a larger controller gain K_c (with the choice $\tau_c = \theta$ the increase in K_c is up to a factor of 2 for a small original delay θ_0). The integral time τ_1 may also be reduced by up to a factor of 2. Thus, the ‘first part’ of the half rule makes both the high-frequency (K_c) and low-frequency (K_c/τ_1) controller gain larger, and the effect is largest when the original delay θ_0 is small compared to τ_{20} .

Let us next consider the ‘second part’ of the half rule, which is the increase in τ_1 (from τ_{10} to $\tau_{10} + \tau_{20}/2$). This increases the controller gain K_c by up to a factor 1.5 (the largest increase is when $\tau_{10} = \tau_{20}$). However, exactly for cases where we get an increase in K_c (which is when τ_{20} is large), we also get an increase in the integral time because $\tau_1 = \tau_1$. Thus, the value of K_c/τ_1 (which is the low-frequency controller gain) is unchanged. Thus, the ‘second part’ of the half rule makes the high-frequency controller gain K_c larger, and the effect is largest when τ_{20} is close to τ_{10} .

In summary, we see that ‘both parts’ the half rule result in less conservative settings, especially for cases where τ_{20} (the largest neglected time constant) is large.

2. Tuning rules

The main contribution of the paper in terms of tuning rules is the correction on the integral time for integrating processes. This significantly improves the load rejection capabilities when compared to the IMC tuning rules of Rivera *et al.* (1986).

When deriving this correction it was required that the system is just at the limit to oscillations ($\zeta = 1$ in equation (20)), which results in the factor 4 in the expression $\tau_1 = 4(\tau_c + \theta)$. More generally, ζ could be left as a degree of freedom (thanks to Finn Haugen for noting this) which would result in the following intergral time for 'integrating' processes: $\tau_1 = c(\tau_c + \theta)$ where $c = 4\zeta^2$. For example, $\zeta = 0.7$ (some low-frequency oscillations allowed) gives the value $\tau_1 = 2(\tau_c + \theta)$. This gives a faster approach to steady state than $\tau_1 = 4(\tau_c + \theta)$ (corresponding to $\zeta = 1$), but on the other hand the robustness is not as good.

3. Tuning parameter

The simple choice $\tau_c = \theta$ for the tuning parameter was also originally chosen for its simplicity, and it is further justified because the resulting value of M_s in the range 1.6 to 1.7 usually gives a good trade off between speed of response and robustness.

4. Derivative action

Derivative action is recommended only for dominant second order processes (with τ_2 larger than θ , approximately). In the simulations I selected the filter time constant $\tau_F = 0.01\tau_D$ (see Section 1.2), but this is *not* the recommended value. Rather, τ_F should be adjusted online, as mentioned in Section 6.2, in order to reduce the input usage and sensitivity to measurement noise. A typical value is $\tau_F = (0.1 - 0.2)\tau_D$ (see also Section 1.2), but as mentioned in Section 6.2, one may increase τ_F up to about $0.5\tau_D$ if measurement noise is a serious problem.

5. Smooth control

The tunings in this work are chosen to give 'fast but still robust control'. However, there are cases where this gives unnecessary fast control. This is further discussed in the following paper: 'Tuning for smooth control: Lower limit on the controller gain for acceptable disturbance rejection', which was presented at the IFAC DYCOPS conference (Hong Kong 2003/2004). See http://www.nt.ntnu.no/users/skoge/publications/2003/tuningPID/smooth_tunings/

6. Second-order oscillatory process

As mentioned in Remark 3 on page 304, oscillatory processes are not covered in the paper, mainly because this requires one to use the ideal (parallel) form of the PID controller in equation (35) as derived in the following. Consider a second-order with delay process on standard form

$$g(s) = k \frac{e^{-\theta s}}{\tau_0^2 s^2 + 2\tau_0 \zeta s + 1}$$

where τ_0 is the time constant ($\tau_0 = 1/\omega_0$ where ω_0 is the natural frequency of oscillations for the plant) and ζ is the damping factor. $|\zeta| < 1$ gives an underdamped system with oscillations. We here assume a stable process with ζ non-negative.

We use the *ideal-form (parallel)* PID controller

$$c(s) = K'_c \left(1 + \frac{1}{\tau'_I s} + \tau'_D s \right) = \frac{K'_c}{\tau'_I} (\tau'_I \tau'_D s^2 + \tau'_I s + 1)$$

Direct synthesis based on $g(s)$ yields controller

$$c(s) = \frac{1}{k} \cdot \frac{(\tau_0^2 s^2 + 2\tau_0 \zeta s + 1)}{(\tau_c + \theta)s}$$

and matching terms yields the following tunings for the ideal form (parallel) PID controller:

$$K'_c = \frac{1}{k} \frac{2\tau_0 \zeta}{\tau_c + \theta}$$

$$\tau'_I = 2\tau_0 \zeta$$

$$\tau'_D = 0.5\tau_0/\zeta$$

With the choice $\tau_c = \theta$ the controller gain becomes

$$K'_c = \frac{1}{k} \frac{\tau_0 \zeta}{\theta}$$

The stability margins with this choice are given by the first column of Table 2 (GM = 3.14, PM = 61.4 degrees, Ms = 1.59).

For $\zeta > 1$ (underdamped process) we get a second-order process,

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

and the above rules reduce to the ones for 'second-order' given process in Table 1. However, note that the settings in Table 1 are for the cascade form, so they need to be translated using equation (36) to get the 'ideal-form' settings.

7. Unstable process

Rules are yet to be derived for this case. Some preliminary attempts are found in the early Norwegian 1998-version.

8. Ziegler Nichols tunings

The PID tuning rules in the original 1942-paper of Ziegler and Nichols state that the integral time is four times the derivative time. I have assumed in this paper that this is for the ideal (parallel form), so that for the cascade form we have that the integral time is equal to the derivative time; see section 5.3. Actually, I originally assumed that the ZN-tunings were for the cascade form controller, and this can be found in the original version of the paper presented at 2001 AIChE Meeting in Reno. This was based on the argument of Shinskey and others, that the pneumatic PID-controllers at that time gave a cascade form. However, after studying it in more detail (see diploma thesis by Hellem from 2001) and discussing it with Dale Seborg and

Karl Johan Åstrøm, I became convinced that the ZN-settings are for the ideal (parallel) form. First, they work best for the ideal form. Second, I think Nichols did numerical computations on an electronic computer where the ideal form was implemented. The authors were possibly not aware of the difference between the two forms or did not think it was important. We tried to search back into old reports, but unfortunately it seems that we were too late to find out conclusively.