Probably the best **simple** PID tuning rules in the world

Sigurd Skogestad*
Department of Chemical Engineering
Norwegian University of Science and Technology
N–7491 Trondheim Norway

Presented at AIChE Annual meeting, Reno, NV, USA, 04-09 Nov. 2001
Paper no. 276h ©Author
This version: November 19, 2001†

Abstract

The aim of this paper is to present analytic tuning rules which are as simple as possible and still result in a good closed-loop behavior. The starting point has been the IMC PID tuning rules of Rivera, Morari and Skogestad (1986) which have achieved widespread industrial acceptance. The integral term has been modified to improve disturbance rejection for integrating processes. Furthermore, rather than deriving separate rules for each transfer function model, we start by approximating the process by a first-order plus delay processes (using the "half method"), and then use a single tuning rule. This is much simpler and appears to give controller tunings with comparable performance. All the tunings are derived analytically and are thus very suitable for teaching.

1 Introduction

Hundreds, if not thousands, of papers have been written on tuning of PID controllers, and one must question the need for another one. The first justification is that PID controller is by far the most widely used control algorithm in the process industry, and that improvements in tuning of PID controllers will have a significant practical impact. The second justification is that the simple rules and insights presented in this paper may contribute to a significantly improved understanding into how the controller should be tuned.

The PID controller has three principal control effects. The proportional (P) action gives a change in the input (manipulated variable) directly proportional to the control errorr. The integral (I) action gives a change in the input proportional to the integrated error, and its main purpose is to eliminate offset. The less commonly used derivative (D) action is used in some cases to speed up the response or to stabilize the system, and it gives a change in the input proportional to the derivative of the controlled variable. The overall controller output is the sum of the contributions from these three terms. The corresponding three adjustable PID parameters are most commonly selected to be

*E-mail: skoge@chembio.ntnu.no; Phone: +47-7359-4154

†This is the revised version where the ZN-PID tunings are assumed to be given for a parallel form (ideal) controller. In the version distributed at the meeting (see Version 06Nov01), it was (probably incorrectly) assumed that the ZN-PID tunings were for a cascade form controller.

- Controller gain K_c (increased value gives more proportional action and faster control)
- Integral time τ_I [s] (decreased value gives more integral action and faster control)
- Derivative time τ_D [s] (increased value gives more derivative action and faster control)

Although the PID controller has only three parameters, it is not easy, without a systematic procesure, to find good values (tunings) 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 objective of this paper is to provide simple model-based tuning rules that give insight into how the tuning depends on the process parameters based on very simple process information. These rules may then be used to assist in retuning the controller if, for example, the production rate is changed. Another related objective is that the rules should be so simple that they can be memorized.

There has been previous work along these lines; most noteworthy the early paper by Ziegler and Nichols (1942), the IMC PID-tuning paper by Rivera, Morari and Skogestad (1986), and the book by Smith and Corripio (1985). The Ziegler-Nichols tunings result in a very good disturbance response for integrating processes, but are otherwise known to result in rather aggressive tunings (e.g., Tyreus and Luyben (1992)), and also give poor performance for processes with a dominant delay. On the other hand, the IMC-tunings of Rivera *et al.* (1986) are known to result in poor disturbance response for integrating processes (e.g., Chien and Fruehauf (1990), Horn *et al.* (1996)), but generally give very good responses for setpoint changes.

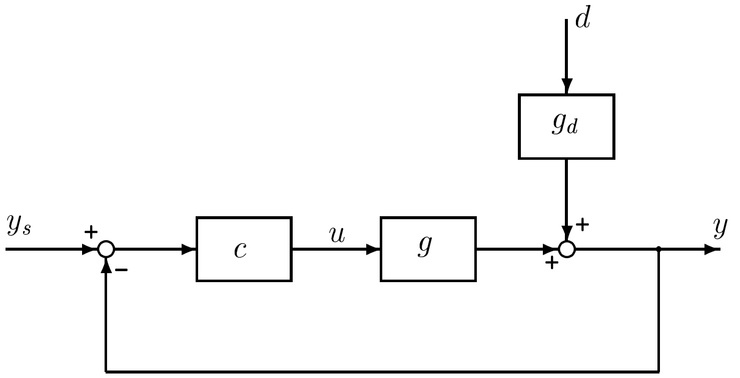

Figure 1: Block diagram of feedback control system.
In the simulations we consider input "load" disturbances ($g_d = g$). c is here a PID-controller

Notation. The notation is summarized in Figure 1. $g(s) = \frac{\Delta y}{\Delta u}$ denotes the process transfer function and $c(s)$ is the feedback part of the controller. The Δ to indicate deviation variables is deleted in the following. u is the manipulated input (controller output), d the disturbance, y the controlled output, y_s the setpoint (reference) for the controlled output, and $y - y_s$ the control error (offset).

2 Summary of method

2.1 Process information

The controller tunings are based on first approximating the process by a first- or second-order plus delay model with the following model information (see Figure 2):

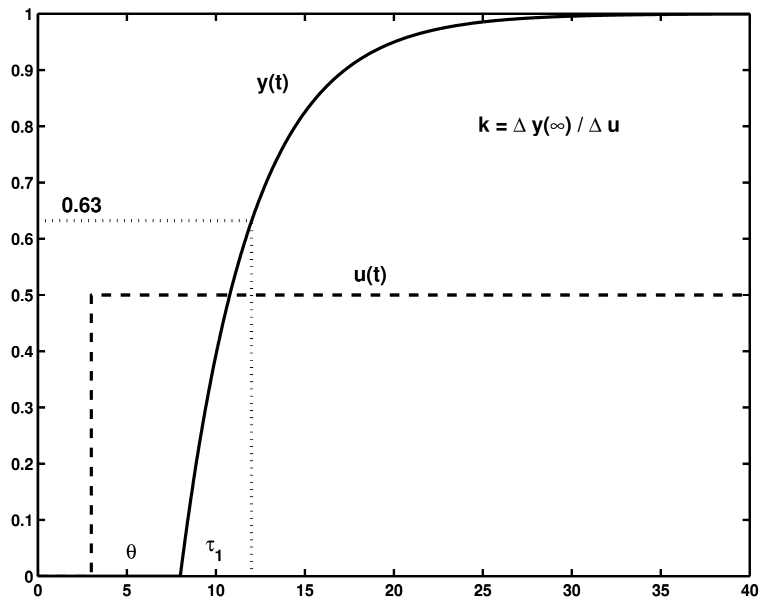

Figure 2: Step response of first-order with delay system, $g(s) = ke^{-\theta s}/(\tau_1 s + 1)$.

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

If the response is sluggish or integrating, i.e. typically if $\tau_1 > 8\theta$, then the exact value of the time constant τ_1 and of the gain k may be difficult to obtain and is also not important for controller design. For such processes one may, instead of using k and τ_1, set $1/\tau_1 = 0$ and obtain a good value for the

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

On transfer function form, the resulting model is then

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

Most of the examples in this paper (Section 5) assume that we start from a more detailed (high-order) model, and when deriving a model from experimental data it is recommended to first derive a more detailed model. To derive a first- or second-order model on the form (1) it is proposed to use a very simple method where the *effective delay* θ is taken as the sum of the

· "true" delay

+ inverse reponse time constant(s)

+ half of the largest neglected time constant not included in τ_1 (or τ_2 if we choose to use a second-order model) (the "half rule").

+ all smaller high-order time constants

The "other half" of the largest neglected time constant is added to τ_1 (or to τ_2 if we use a second-order model) – for more details see Section 5.

2.2 Recommended SIMC-PID tunings

The tunings given below are for the cascade (series) form PID controller:

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

The reason for using the cascade form is that the PID rules in this case are much simpler when we have derivative action. Following the internal model control approach (Rivera *et al.* 1986) where one specifies a first-order closed-loop response with time constant τ_c, the following SIMC tunings[1] are recommended for the process in (1) (see derivation in Section 3):

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

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

$$\tau_D = \tau_2 \tag{5}$$

where $-\theta < \tau_c < \infty$ is the tuning parameter. For robust tunings it is recommended to use $\tau_c \geq \theta$ (see below). The optimal value of τ_c is determined by a trade-off between (see also Figure 3):

1. Fast speed of response and good disturbance rejection (which are favored by a small value of τ_c), and
2. Stability, robustness and small input usage (which are favored by a large value of τ_c).

The original IMC tuning rules (Rivera *et al.* 1986) yield $\tau_I = \tau_1$, that is, the integral time is selected so as to exactly cancel the dynamics correpsonding to the dominant (first-order) time constant τ_1. However, this gives a very sluggish response to input (load) disturbances for "slow" (τ_1 large) or integrating processes (Chien and Fruehauf 1990). Therefore, for such processes it is suggested to use a smaller integral time, and the recommended value $\tau_I = \frac{4}{k' K_c}$ just avoids the slow oscillations that would otherwise result by using "too much" integral action for such a process (see derivation below).

Derivative action is primarily recommended for processes with dominant second order dynamics (with $\tau_2 > \theta$), and the derivative time is selected so as to cancel the second-largest process time constant. In addition, derivative action is often needed to stabilize unstable processes, but such processes are not covered here.

2.2.1 Tuning for fast response with good robustness

The main limitation on achieving a fast closed-loop response is the time delay. Selecting the desired response time equal to the time delay,

$$\text{SIMC}: \quad \tau_c = \theta \tag{6}$$

[1] The S in "SIMC" denotes Skogestad or Simple – pick your choice.

gives a reasonably fast response with moderate input usage and good robustness margins, and results in the following SIMC-PID tunings which may be easily memorized:

$$K_c = \frac{0.5}{k}\frac{\tau_1}{\theta} = \frac{0.5}{k'}\frac{1}{\theta} \tag{7}$$

$$\tau_I = \min\{\tau_1, 8\theta\} \tag{8}$$

$$\tau_D = \tau_2 \tag{9}$$

Two common robustness measures are the gain margin (GM) and phase margin (PM). Typical minimum requirements are GM>1.7 and PM$>30^o$ (Seborg *et al.* 1989), but for most control loops in the process industries larger margins are recommended. Two other robustness measures are the peak value M_s of the sensitivity function $S = 1/(1+gc)$, and the peak value M_t of the complementary sensitivity, $T = 1 - S = gc/(1+gc)$, for which small values are desirable. For example, $M_s < 2$ guarantees GM>2 and PM$>29.0^o$.

With the SIMC PID-tunings in (7)-(9) the gain margin is typically above 3, the phase margin is about 50^o to 60^o, M_s is 1.7 or less, and M_t is 1.3 or less. Specifically, with $\tau_I = \tau_1$ the system always has a gain margin of 3.14, a phase margin of 61.4^o, $M_s = 1.59$, $M_t = 1.00$, and a maximum allowed time delay error of 2.14θ (i.e., the tunings provide time delay error robustness in excess of 200%) (see Table 1). As expected, the robustness margins are somewhat poorer for "sluggish" processes, where we in order to improve the disturbance response use $\tau_I = 8\theta$. Specifically, for an integrating process the suggested tunings give a a gain margin of 2.96, a phase margin of 46.9^o, $M_s = 1.70$, $M_t = 1.30$, and a maximum allowed time delay error of 1.49θ.

Process $g(s)$ Controller gain, K_c Integral time, τ_I	$\frac{k}{\tau_1 s+1}e^{-\theta s}$ $\frac{0.5}{k}\frac{\tau_1}{\theta}$ τ_1	$\frac{k'}{s}e^{-\theta s}$ $\frac{0.5}{k'}\frac{1}{\theta}$ 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 1: Robustness margins for first-order and integrating delay process using SIMC-tunings in (7) and (8) ($\tau_c = \theta$). The same margins apply to second-order processes if we choose $\tau_D = \tau_2$.

Derivation: For the first-order delay process with $\tau_I = \tau_1$ the resulting loop transfer function is $L(s) = gc = \frac{0.5}{\theta s}e^{-\theta s}$. The frequency ω_{180} where the phase of L is -180^o is then $\angle L = -\omega_{180}\theta - \frac{\pi}{2} = -\pi \quad \Rightarrow \quad w_{180} = \frac{\pi}{2}\frac{1}{\theta}$. The gain of L as a function of frequency is $0.5/\theta\omega$ and by evaluating the gain at the frequency ω_{180} we find that GM $= 1/|L(j\omega_{180})| = \pi = 3.14$. Similarly, we can show that the frequency ω_c where $|L| = 1$ is $\omega_c = 0.5/\theta$ og we find that PM$= \pi - \angle L(j\omega_c) = \pi/2 - 0.5[rad] = 61.4^o$. The maximum allowed time delay error is then $\Delta\theta = \text{PM}/\omega_C = (\pi - 1)\theta = 2.14\theta$.

These good margins come at the expense of a somewhat more sluggish time response compared to that which can be achieved with more aggressive tunings. Note that for the case with $\tau_I = \tau_1$, increasing K_c by a factor of 2 (corresponding to choosing $\tau_c = 0$), reduces PM from 61^o to 33^o and reduces GM from 3.14 to 1.57, which are rather poor robustness margins. Thus, to maintain resonable robustness, the controller gain should be at most a factor of 2 larger than the value given in (7).

2.2.2 Tuning for slow response with acceptable disturbance rejection

In many practical situations we do not require very fast control, and to reduce the use of manipulated inputs and generally make operation smoother we may want to use lower controller gains, or equivalently increase. For example, this may be the case for processes with a small (effective) time delay. for example, a pure first-order process. More generally, there are cases where we want to use as little control as possible, that is, we want a slow or smooth response. However, there is usually some performance requirements in terms of the allowed output variation, and this gives a minimum controller gain needed to achieve satisfactory disturbance rejection. For example, for the case of input "load" disturbances we must approximately require that (see derivation of (33)):

$$K_c \geq \frac{d_u}{y_{max}} \tag{10}$$

Here $y_{\max}$ is the allowed output error $(y-y_s)$, and d_u is the magnitude of the input "load" disturbance. As expected, tight control with y_{max} small requires K_c large, as does a large disturbances with d_u large. After deciding on a reasonable value for K_c, one may from (3) back-calculate the corresponding value of τ_c, and then oobtain the integral time according to (4).

An application of (10) is for "averaging level control", where the objective is that the outflow (which is the manipulated input u in this case) shold vary as slowly as possible when there are variations in the inflow (which is the disturbance d_u), but subject to the requirement that the tank level (which is the controlled variable y) should not exceed it constraints.

Remark. If the "minimum" controller gain given by (10) is larger than the "maximum" the controller gain given in (7), then the process is not controllable – at least not with PID control with reasonably robust tunings. In words, the speed of response required for disturbance rejection (10) is faster than what can be achieved with the given time delay (7).

Example. *Consider a second-order with delay process with time constants $\tau_1 = 6$ and $\tau_2 = 1.2$, and time delay $\theta = 0.25$:*

$$g(s) = 4\frac{e^{-0.25s}}{(6s+1)(1.2s+1)} \tag{11}$$

The requirements is that the output deviation should be less than $y_{max} = 1$ in response to a load disturbance $d_u = 0.5$. It is also desirable that control is as smooth as possible.

Tuning for fast response. *With $\tau_c = \theta = 0.25$ the recommended tunings (7)-(9) for a cascade form PID controller are*

$$K_c = \frac{0.5}{k}\frac{\tau_1}{\theta} = 3; \quad \tau_I = 8\theta = 2; \quad \tau_D = \tau_2 = 1.2 \tag{12}$$

The load disturbance response in Figure 3 (solid line) is much better than the requirement, with a output deviation in response to the load disturbance of less than 0.1. However, the input has some overshoot and oscillations.

Tuning for slow response. *The above response is here unecessary fast so the controller gain may be reduced. However, to reject the disturbance we need a minimum gain, which from (10) is approximately $K_c = \frac{d_u}{y_{max}} = \frac{0.5}{1} = 0.5$ (corresponding to $\tau_c = 2.75$ in (3)), and the resulting PID tunings according to (4) and (5) are*

$$K_c = 0.5; \quad \tau_I = \tau_1 = 6; \quad \tau_D = \tau_2 = 1.2 \tag{13}$$

The load disturbance response in Figure 3 (dashed line) has a maximum output deviation $y - y_s$ of about $1.5 - 1 = 0.5$ which is well below 1, and the input is smooth with no overshoot or oscillations. Thus, this tuning is preferred in practice.

Remark: *We may reduce K_c further below 0.5 and still achieve an output deviation less than 1. The reason why (10) is not tight, is that (1) the expression is derived for sinusoidal disturbances whereas we consider a step disturbance, and (2) the derivative time is quite close to the integral time so that the controller gain as a function of frequency does not come down to its asymptotic value of K_c.*

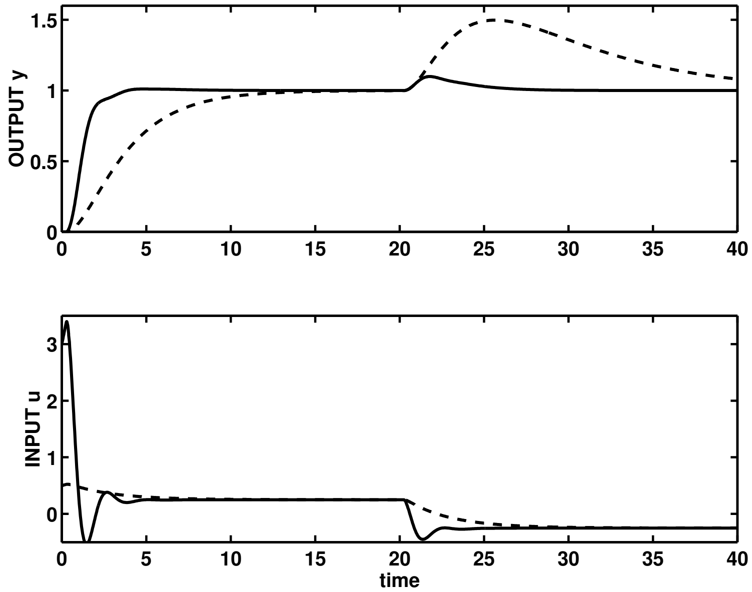

Figure 3: Responses for process (11) with "fast" PID-tunings (12) (solid line) and "slow" PID-tunings (13) (dashed line)

Setpoint change from $y_s = 0$ to $y_s = 1$ occurs at $t = 0$. Load disturbance of magnitude 0.5 occurs at $t = 20$.

2.3 Ideal PID controller

The above tunings (K_c, τ_I, τ_D) are for the cascade form PID controller in (2). To derive the corresponding tunings for the "ideal" (parallell form) PID controller

$$\text{Ideal PID}: \quad c'(s) = K_c'\left(1 + \frac{1}{\tau_I' s} + \tau_D' s\right) = \frac{K_c'}{\tau_I' s}\left(\tau_I'\tau_D' s^2 + \tau_I' s + 1\right) \tag{14}$$

we must use the formulas

$$K_c' = K_c\left(1 + \frac{\tau_D}{\tau_I}\right); \quad \tau_I' = \tau_I\left(1 + \frac{\tau_D}{\tau_I}\right); \quad \tau_D' = \frac{\tau_D}{1 + \frac{\tau_D}{\tau_I}} \tag{15}$$

Note that it is *not* always possible to do the reverse and obtain cascade tunings from the ideal tunings. This is because the ideal form is more general as it also allows for complex zeros in the controller. Thus, if we want to derive PID-tunings for a second-order oscillatory process which has complex poles, then we should start directly with the ideal PID controller.

The tuning parameters for the the cascade and ideal forms are identical 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 ($\tau_I = \infty$).

The SIMC-PID cascade tunings in (7)-(9) correspond to the following *SIMC ideal-PID tunings* ($\tau_c = \theta$):

$$\tau_1 \leq 8\theta: \quad K_c' = \frac{0.5}{k}\frac{\tau_1 + \theta_2}{\theta}; \quad \tau_I' = \tau_1 + \tau_2; \quad \tau_D' = \frac{\tau_2}{1 + \frac{\tau_2}{\tau_1}} \tag{16}$$

$$\tau_1 \geq 8\theta: \quad K_c' = \frac{0.5}{k}\frac{\tau_1}{\theta}\left(1+\frac{\tau_2}{8\theta}\right); \quad \tau_I' = 8\theta + \tau_2; \quad \tau_D' = \frac{\tau_2}{1+\frac{\tau_2}{8\theta}} \tag{17}$$

We see that the tuning rules for the ideal form are much more complicated.

Example. *Consider the second-order process in (11) with cascade-form PID tunings given in (12). The corresponding tunings for the ideal PID controller in (14) are*

$$K_c' = 4.8; \quad \tau_I' = 3.2; \quad \tau_D' = 0.75$$

The robustness margins with these tunings are given by the first column in Table 1.

3 Derivation of SIMC tuning rules

We will now derive the tuning rules in (3)-(5).

3.1 Tuning for setpoint response

Our starting point is a second-order with delay model,

$$g(s) = k\frac{e^{-\theta s}}{(\tau_1 s+1)(\tau_2 s+1)} \tag{18}$$

for which we want to derive analytical PID-settings. We use the *direct synthesis* approach (e.g., see Rivera *et al.* (1986)) where we specify a closed-loop setpoint response y/y_s. For the system in Figure 1

$$\frac{y}{y_s} = \frac{gc}{gc+1} \tag{19}$$

where c is the feedback controller, and we have assumed that the measurement of the output y is perfect. If we specify a desired setpoint response, then we can from (19) solve for the corresponding controller,

$$c(s) = \frac{1}{g(s)}\frac{1}{\frac{1}{(y/y_s)_{\text{desired}}}-1} \tag{20}$$

Following Rivera *et al.* (1986), we specify that we, after the delay, desire a simple first-order response

$$\left(\frac{y}{y_s}\right)_{\text{desired}} = \frac{1}{\tau_c s+1}e^{-\theta s} \tag{21}$$

We have kept the delay in the "desired" response because it is unavoidable. τ_c is the desired closed-loop time constant, and is the sole tuning parameter for the controller. Substituting 21) into (20) gives a "Smith Predictor" controller (Smith 1957):

$$c(s) = \frac{(\tau_1 s+1)(\tau_2 s+1)}{k}\frac{1}{(\tau_c s+1-e^{-\theta s})} \tag{22}$$

To get a PID-controller we introduce in (22) the follwing first-order Taylor approximation for the delay [2]

$$e^{-\theta s} \approx 1 - \theta s \tag{23}$$

[2]Rivera *et al.* (1986) used a slightly different approach where the Pade approximation was introduced in the process, before deriving the controller.

and derive

$$c(s) = \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{k} \frac{1}{(\tau_c + \theta)s} \tag{24}$$

which is a cascade form PID-controller (2) with

$$K_c = \frac{1}{k}\frac{\tau_1}{\tau_c + \theta}; \quad \tau_I = \tau_1; \quad \tau_D = \tau_2 \tag{25}$$

3.2 Modifying the integral term for improved disturbance rejection

The PID tunings in (25) were derived by considering the *setpoint* response. We found 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 when it comes to setpoint changes and also for disturbances entering directly at the process *output*. However, it is well known that for processes where τ_1 is "large" (e.g. an integrating processes), this choice results in a long settling time for *input "load" disturbances* (Chien and Fruehauf 1990). The reason is that the controller cancels the process dynamics, whereas for a disturbance occuring at the input we actually want to keep the dynamics in the loop. To improve the load disturbance response we therefore want to reduce the integral time. However, we must not reduce the integral time too much, because otherwise we will encounter slow oscillations caused by almost having two integrators in series (one from the slow dynamics in the process and one from the controller). This is illustrated in Figure 4, for a "slow" process with $\tau_1 = 30$ and $\theta = 1$, where we consider four values of the integral time:

- $\tau_I = \tau_1 = 30$ (original IMC-rule) gives excellent setpoint response, but slow settling for a load disturbance.
- $\tau_I = 8\theta = 8$ (SIMC-rule) gives faster settling for a load disturbance.
- $\tau_I = 4$ gives even faster settling, but the setpoint response (and robustness) is poorer.
- $\tau_I = 2$ gives poor response with 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 ($\tau_I = 8\theta = 8$ 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 "fast" oscillations at about frequency $1/\theta$ caused by the delay). 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 is such that $(\tau_1\omega)^2$ is much larger than 1 (see footnote 5). With a PI controller $c = K_c\left(1 + \frac{1}{\tau_I}s\right)$ the closed-loop characetristic equation $1 + gc$ then becomes

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

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

$$\tau_0 = \sqrt{\frac{\tau_I}{k'\ K_c}}; \quad \zeta = \frac{1}{2}\sqrt{k'\ K_c\ \tau_I} \tag{26}$$

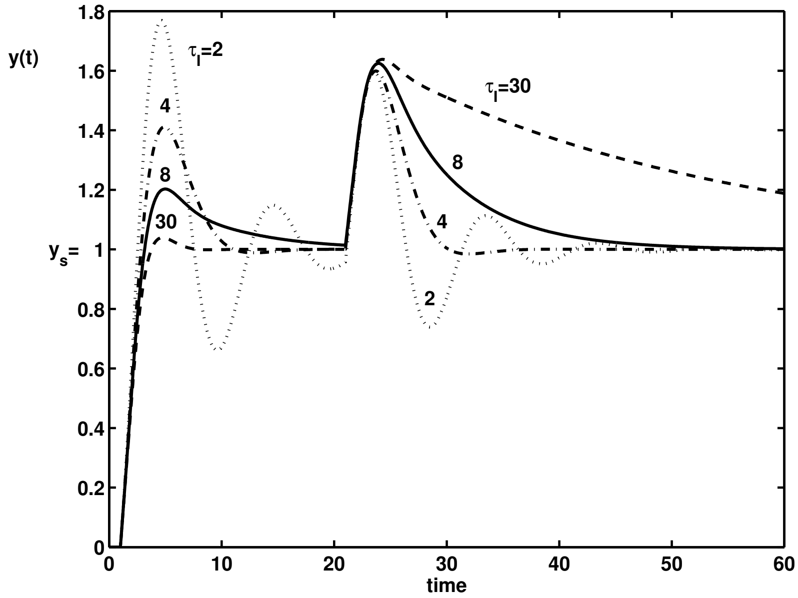

Figure 4: Effect of changing the integral time τ_I for PI-control of "slow" process $g(s) = e^{-s}/(30s+1)$ with $K_c = 15$.
Load disturbance of magnitude 10 occurs at $t = 20$.

To avoid slow oscillations we must have a damping coefficient $\zeta \geq 1$, or quivalently

$$K_c \tau_I > 4/k' \tag{27}$$

Of course, some oscillations may be tolerated, but nevertheless a good starting value is to have $\zeta = 1$ (see also Marlin (1995) page 588), and (27) gives

$$\tau_I = \frac{4}{k' K_c} \tag{28}$$

which is the value recommended in (4). The choice $K_c = \frac{0.5}{k'}\frac{1}{\theta}$ (7) (corresponding to $\tau_c = \theta$) gives $\tau_I = 8\theta$ as given in (8). For a first-order with delay process this gives a gain margin better than 2.96, a phase margin better than 46.9^o, and M_s less than 1.70; see Table 1.

3.3 Lower limit on controller gain

We here consider the performance requirements for disturbance rejection. The linear transfer fuction model in deviation variables is

$$y = g(s)u + g_d(s)d \tag{29}$$

where $g_d(s)$ is the disturbance transfer function model. With feedback control, $u = -c(s)y$, the effect of a disturbance d on the control output y is

$$y = S(s)g_d(s)d$$

where $S(s) = 1/(1 + g(s))c(s)$ is the sensitivity function. Let d denote the disturbance magnitude, and y_{max} the allowed output variation. We assume that this requirement applies on a frequency-by-frequency basis, i.e., for a sinusoidal disturbance with frequency ω [rad/min] and magnitude d, the resulting sinusoidal output should have a magnitude less than y_{max}. Since the sinusoidal response is mathematically obtained by setting $s = j\omega$, the requirement becomes

$$|y(j\omega)| = |S(j\omega)| \cdot |g_d(j\omega)| \cdot d \leq y_{max}$$

or

$$|1 + g(j\omega)c(j\omega)| \geq \frac{|g_d(j\omega)| \cdot d}{y_{max}}$$

At low frequencies (i.e., within the closed-loop bandwith) we have that $|gc| \gg 1$ and we derive the following lower limit on the frequency-dependent controller gain for acceptable disturbance rejection

$$|c(j\omega)| \geq \frac{|g_d(j\omega)| \cdot d}{|g(j\omega)| \cdot y_{max}} \tag{30}$$

At lower frequencies, where this expression applies, we effectively have "perfect control" and $y \approx 0$. From (29) the required input to reject the disturbance (i.e., achieve $y = 0$) is $u_d = (g_d/g)d$, and we derive the following alternative expression

$$|c(j\omega)| \geq \frac{u_d(j\omega)}{y_{max}} \tag{31}$$

where $u_d(j\omega)$ is the magnitude of the input change needed to reject the disturbances and y_{max} is the maximum allowed output deviation $(y - y_s)$. By constructing a controller which just satisfies the bound (30) or (31), we obtain the "slowest" acceptable controller (this is generally not a PID controller).

For the special case of a **load disturbance** (distubance d_u at the input) we have $g_d = g$ and the requirement (30) becomes

$$c(j\omega) \geq \frac{d_u}{y_{max}} \tag{32}$$

For a P-, PI- and PID-controller the controller gain $|c(j\omega)|$ has a minimum asymptotic value [3] of K_c, and we derive the following lower limit on the controller gain,

$$K_c \geq \frac{d_u}{y_{max}} \tag{33}$$

From (31) and (33) we then derive the following useful rule:

- The minimum controller gain is approximately equal to the expected input change divided by the allowed output variation.

We can rearrange (33) into $y_{max} = d_u/K_c$, which in words says that that the maximum output change y_{max} in response to a load disturbance is d_u/K_c. This is for a sinusoidal disturbance, but as illustrated in the simulations, a step disturbance often results in a similar value. For example, in Figure 4 we see that the maximum output deviation $y - y_s$ in response to a step disturbance is about 0.65 (independent of the integral time) which compares well the value $d_u/K_c = 10/15 = 0.67$.

[3] For a PID controller the break frequencies are at $1/\tau_I$ and $1/\tau_D$, and the controller gain as a function of frequency will only reach its asymptotic minimum value of K_c for cases where the integral time τ_I is significantly larger than the derivative time τ_D.

4 Some special cases and comparison with other tuning methods

We here present some special cases and compare the suggested SIMC tuning rules with the classical "closed-loop" tuning rules of Ziegler and Nichols (1942), as well as some other tuning methods. We find that the simple SIMC tunings generally perform very well.

Ziegle-Nichols (ZN) tuning rules. The first step in the Ziegler-Nichols procedure is to generate sustained oscillations with a P-controller, and from this obtain the "ultimate" gain K_u and corresponding "ultimate" period P_u. Based on simulations, Ziegler and Nichols (1942) recommended the following ZN tunings: [4]

$$\begin{aligned}
\text{P}-\text{control}: &\quad K_c = 0.5K_u \\
\text{PI}-\text{control}: &\quad K_c = 0.45K_u;\ \tau_I = P_u/1.2 \\
\text{PID}-\text{control (ideal)}: &\quad K_c' = 0.6K_u;\ \tau_I' = P_u/2;\ \tau_D' = P_u/8 \\
\text{PID}-\text{control (cascade)}: &\quad K_c = 0.3K_u;\ \tau_I = P_u/4;\ \tau_D = P_u/4
\end{aligned}$$

These tunings were based on experiments on a Taylor pneumatic controllers, and in this version of the paper we assume that this was close to today's ideal PID controller.

Tyreus-Luyben modified ZN tuning rules. The ZN tunings were derived to give decay ratio of 1/4. This is too aggressive for most process control systems, where oscillations and overshoot are usually not desired at all. This led Tyreus and Luyben (1992) to recommend the following PI-rules for more conservative loops:

$$K_c = 0.313K_u;\ \tau_I = 2.2P_u$$

Regressed tuning rules. Many papers on PID control include comparisons with the tuning rules of Cohen and Coon (1953) where the tunings are given by analytical functions of k, τ_1 and θ. These tunings were also derived for a decay ratio of 1/4 and are generally too aggressive, and performance is usually poor (this is probably why it is popular to compare with them since anyoone can beat them). Later, there has been many papers along these lines with improved tunings, e.g. see Ho *et al.* (1998).

Astrom/Schei PI tuning rules. Schei (1994) argued that in process control applications we usually want a robust design with the highest possible attenuation of low-frequency disturbances, and suggested to maximize the low-frequency controller gain

$$K_I = \frac{K_c}{\tau_I} \tag{34}$$

subject to given robustness constraints on the sensitivity peaks M_s and M_t. Astrom *et al.* (1998) showed how to formulate this as a convex 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) to 2 (more agressive tuning). To improve the setpoint performance Astrom *et al.* (1998) use a "two degrees of freedom controller" where they use only a fraction b of the propotional action on the setpoint, but we do not use this here (i.e., we set $b = 1$).

Original IMC PID tuning rules. Rivera *et al.* (1986) derived PID tunings for various processes. For a first-order with delay process their "improved IMC PI-tunings" for fast response ($\varepsilon = 1.7\theta$) are

$$\text{IMC PI}: \quad K_c = \frac{0.588}{k}\frac{\tau_1 + \frac{\theta}{2}}{\theta}; \quad \tau_I = \tau_1 + \frac{\theta}{2} \tag{35}$$

[4] In this version of the paper we assume that Ziegler and Nichols used an ideal (parallel form) PID controller. Note that this only effects the cases where derivative action is used.

and their PID-tunings for fast response ($\varepsilon = 0.8\theta$) are

$$\text{IMC cascade} - \text{PID}: \quad K_c = \frac{0.769}{k}\frac{\tau_1}{\theta}; \quad \tau_I = \tau_1; \quad \tau_D = \frac{\theta}{2} \tag{36}$$

Note that $\tau_I \geq \tau_1$ so the response to input load disturbances is poor for "slow" processes with τ_1 large.

4.1 Pure time delay process

The pure time delay process is

$$g(s) = ke^{-\theta s} \tag{37}$$

Note that a pure P-controller is unacceptable for this process, because even with maximum gain (at the limit to instability) the steady-state offset is 0.5 (50%). Thus integral action will be needed.

SIMC tunings. This is a special case of (1) with $\tau_1 = 0$ and $\tau_2 = 0$. The rules (3) and (4) give $K_c \to \infty$ and $\tau_I = \tau_1 = 0$. More precicely, the controller becomes

$$c(s) = \frac{1}{k\ (\tau_c + \theta)}\frac{1}{s}$$

This is a pure integral controller $c(s) = \frac{K_I}{s}$ where, with the suggested choice $\tau_c = \theta$, the integral gain is

$$K_I = \frac{0.5}{k\ \theta} \tag{38}$$

Remark. To approximate this controller by a PI-controller with parameters K_c and τ_I we may choose τ_I small (typically, less than 0.1θ) and use $K_c = \frac{0.5}{k} \cdot \frac{\tau_I}{\theta}$.

	$K_c \cdot k$	$K_I \cdot k \cdot \theta$	GM	PM	M_s	M_t
SIMC ($\tau_c = \theta$)	0	0.5	3.14	61.4^o	1.59	1.00
Astrom ($M_s = 1.4$)	0.16	0.47	3.77	71.9^o	1.40	1.00
Astrom ($M_s = 2$)	0.26	0.85	2.12	54.6^o	2.00	1.15
ZN	0.45	0.27	2.18	99.4^o	1.85	1.00

Table 2: PI tunings and robustness for pure time delay process, $g(s) = ke^{-\theta s}$. Note that $K_I \stackrel{\text{def}}{=} K_c/\tau_I$.

ZN tunings. For this process, a pure proportial control with gain $K_u = 1/k$ results in persistent oscillations with period $P_u = 2\theta$ (at the limit to instability). The resulting Ziegler-Nichols PI-tunings rules are $K_c = \frac{0.45}{k}$ and $\tau_I = 1.67\theta$ corresponding to GM = 2.18, PM = 99.5 ^{o} and $M_s = 1.85$. Thus, the robustness is acceptable, but the simulations in Figure 5 show that the reponse is sluggish. This may explained by the relatively low integral gain, $K_I = K_c/\tau_I = 0.27/(k\theta)$. The modified ZN PI-settings by Tyreus-Luyben yield a very sluggish response (not shown) due to an even longer integral time ($\tau_I = 2.2P_u = 4.4\theta$). The ZN PID-controller yields instability for this process.

We therefore conclude that the Ziegler-Nichols settings are generally poor for a pure time delay process. This may partly explain the myth in the process industry that PI-control should not be used for processes with large time delays.

Astrom tunings. With $M_s = 1.4$ (robust tunings), Astrom *et al.* (1998) derive a PI-controller $c(s) = K_c + \frac{K_I}{s}$ with $K_c = \frac{0.16}{k}$ and $K_I = \frac{0.4724}{k\theta}$. We note that the integral part is almost the same as for the SIMC integral controller in (38), and the response (not shown) is also quite similar. With

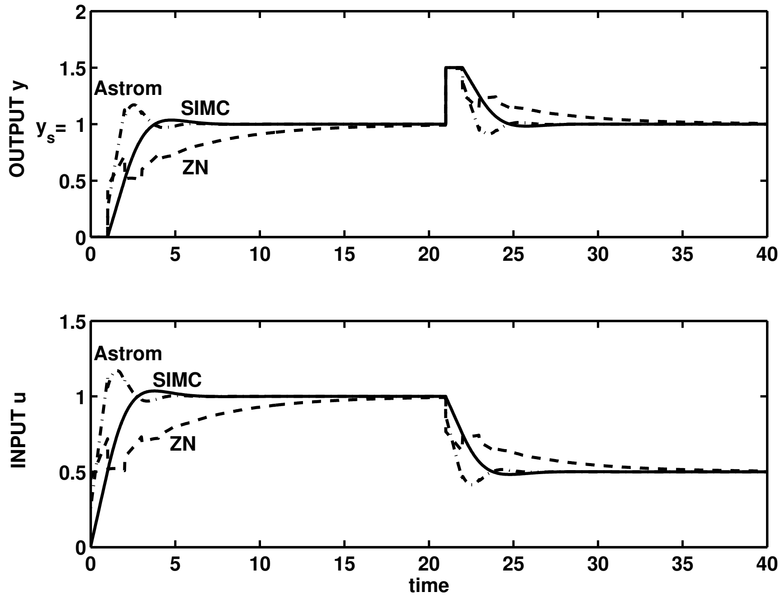

Figure 5: Responses for pure time delay process $g(s) = e^{-s}$ ($\theta = 1$) with tunings from Table 2. SIMC (solid line), ZN (dashed line), Astrom with $M_s = 1.4$ (dashed-dot).

$M_s = 2$ (more aggressive tunings) the response is somewhat faster (see Figure 5) but the reponse with the SIMC integral controller is smoother and more robust.

Remark. With the SIMC I-controller the response is somewhat improved by adding derivative action (e.g. $\tau_D = 0.25\theta$). For the ZN and Astrom PI-controllers the introduction of only very little derivative action (even $\tau_D = 0.01\theta$) gives instability. This illustrates that proportional action should not really be used for a pure delay process.

4.2 Integrating process

Consider an integrating process with delay θ,

$$g(s) = k' \frac{e^{-\theta s}}{s} \tag{39}$$

The corresponding PI tunings for some methods are summarized in Table 3.

	$K_c \cdot k' \cdot \theta$	τ_I/θ	τ_D/θ	GM	PM	M_s	M_t
SIMC ($\tau_c = \theta$)	0.5	8		2.96	46.9^o	1.70	1.30
IMC	0.59	∞		2.66	56.2^o	1.75	1.07
Astrom ($M_s = 1.4$)	0.28	7.0		5.24	47.5^o	1.39	1.43
Astrom ($M_s = 2$)	0.49	3.77		2.77	32.8^o	2.00	1.81
Tyreus-Luyben	0.49	7.32		3.00	45.9^o	1.70	1.33
ZN-PI	0.71	3.33		1.86	24.8^o	2.85	2.37
ZN-PID	0.47	1	1	1.79	32.6^o	2.29	1.92

Table 3: Tunings and robustness for integrating process, $g(s) = k'e^{-\theta s}/s$

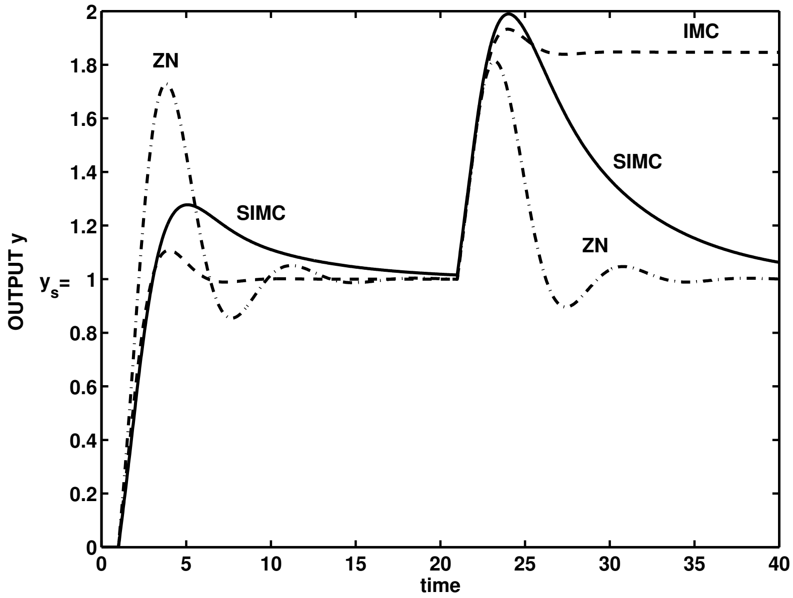

Figure 6: Responses for PI-control of integrating process, $g(s) = e^{-s}/s$, with tunings from Table 3

SIMC tunings. This is a special case of (1) with $\tau_1 \to \infty$. From (3) and (4) we get a PI-controller with

$$K_c = \frac{1}{k'} \cdot \frac{1}{\tau_c + \theta};; \quad \tau_I = 4(\tau_c + \theta) \tag{40}$$

ZN tunings. For this process a proportial controller with gain $K_u = \frac{\pi}{2}\frac{1}{k'}\frac{1}{\theta}$ results in persistent oscillations with period $P_u = 4\theta$. The resulting Ziegler-Nichols PI- and PID-tunings are given in Table 3. The robustness is somewhat poor, but as seen in Figure 6 the ZN PI-tunings give considerably faster settling than the SIMC PI-tunings ($\tau_c = \theta$) for load disturbances. The ZN-PID load response (not shown) is even better. In summary, the Ziegler-Nichols tunings are very well suited for an integrating process, although the robustness is somewhat poor.

Other tuning methods. The IMC tunings of Rivera *et al.* (1986) result in a pure P-controller since $\tau_I = \tau_1 + \frac{\theta}{2} \to \infty$. This P-controller is very good for setpoint changes, but load disturbances integrate and result in steady-state offset (see Figure 6). The Astrom PI-tunings with $M_s = 2$ give responses (not shown) somewhat in between SIMC and ZN. The Tyreus-Luyben modified (conservative) ZN-PI tunings are almost indentical to the SIMC tunings for this particular example.

4.3 Integrating process with delay and lag

Consider an integrating delay process with an additional lag with time constant τ_2,

$$g(s) = k' \frac{e^{-\theta s}}{s(\tau_2 s + 1)} \tag{41}$$

SIMC tunings. The tunings, responses and robustness are as for the process (39), except that we must add derivative action to counteract the lag,

$$\text{PID} - \text{cascade}: \quad K_c = \frac{1}{k'} \cdot \frac{1}{\tau_c + \theta}; \quad \tau_I = 4(\tau_c + \theta); \quad \tau_D = \tau_2 \tag{42}$$

If the time constant τ_2 for the lag is small, then one may approximate the process as $k' \; e^{-(\theta+\tau_2)s}/s$ and derive a PI-controller by using the rules for the integrating proces with delay in (39), but with θ replaced by $\theta + \tau_2$.

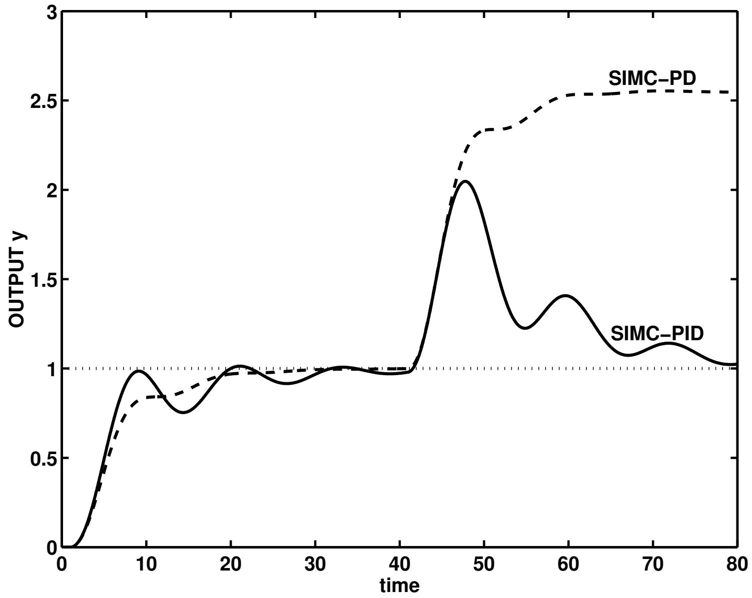

Figure 7: Responses for double integrating process, $g(s) = e^{-s}/s^2$.
SIMC-PD: $K_c = 0.0625$, $\tau_I = \infty$, $\tau_D = 8$.
SIMC-PID: $K_c = 0.0625$, $\tau_I = 8$, $\tau_D = 8$.
Disturbance of magnitude 0.1 occurs at $t = 40$.

If the time constant τ_2 for the lag is large, such that we in effect have a double integrating process, then the load response is poor, and the controller needs (42) to be modified. This is discussed next.

4.4 Double integrating process

$$g(s) = k'' \frac{e^{-\theta s}}{s^2} \tag{43}$$

SIMC tunings. By letting $\tau_2 \to \infty$ and introducing $k'' = k'/\tau_2$ it can be shown that the PID-controller (42) approaches a PD-controller with

$$K_c = \frac{1}{k''} \cdot \frac{1}{4(\tau_c + \theta)^2}; \quad \tau_D = 4(\tau_c + \theta) \tag{44}$$

This controller gives good setpoint responses for the process (43), but results in steady-state offset for load disturbances occuring at the input, see Figure 7. To remove this offset, we need to reintroduce integral action, and as before propose to use $\tau_I = 4(\tau_c + \theta)$. With the choice $\tau_c = \theta$ the resulting SIMC-PID parameters are

$$\mathrm{PID-cascade:} \quad K_c = \frac{0.0625}{k''} \cdot \frac{1}{\theta^2}; \quad \tau_I = 8\theta; \quad \tau_D = 8\theta \tag{45}$$

These tunings give acceptable robustness with GM$= 2.76$, PM$=33.1^o$ (a bit small), $M_s = 1.96$ and $M_t = 1.83$. It should be noted that derivative action is required in order to stabilize this process if we use integral action in the controller.

ZN-tunings can not be derived for this process because we get sustained oscillations with P-control even with $K_c = 0$.

5 More complex cases: Obtaining the effective delay using the half rule

The SIMC-PID tunings are for a first-order or second-order plus delay process model. This may seem restrictive, but more complex models can be handled by approximating the remaining high-order dynamics by an effective delay.

The problem of obtaining the effective delay can be set up as a parameter estimation problem, for example, by making an least squares approximation of the open-loop step response. However, our goal is to use the resulting effective delay to obtain controller tunings, 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, by the in terms of the minimum integrated absolute error (IAE)).

However, our objective is not "optmality" but "simplicity", so we choose to use a much simpler approach where we include all the "neglected" small time constants in the effective delay θ, except for the largest which we distribute evenly to the delay and the remaining time constant using the "half method". This extremely simple rule has been applied to numerous examples, and leads to very good final PID tunings.

The starting point is a model on the following standard form

$$g_0(s) = k\ e^{-\theta_o s}\ \frac{\prod_{j=1}^{m}(T_{j0}s+1)}{\prod_{i=1}^{n}(\tau_{i0}s+1)} = \frac{k'\ e^{-\theta_o s}}{s+1/\tau_{10}}\ \frac{\prod_{j=1}^{m}(T_{j0}s+1)}{\prod_{i=2}^{n}(\tau_{i0}s+1)} \tag{46}$$

where T_{j0} are the time constants for overshoot (or inverse response for the case when T_{j0} is negative), τ_{i0} are the lag time constants, and $k' = k/\tau_{10}$. We want to approximate (46) by a first- or second-order plus delay model,

$$g(s) = k\frac{e^{-\theta s}}{(\tau_1 s+1)(\tau_2 s+1)} = k'\frac{e^{-\theta s}}{(s+1/\tau_1 s)(\tau_2 s+1)} \tag{47}$$

Here θ is the "effective" delay, and we select $\tau_2 = 0$ if a first-order approximation is desired.

Rules for obtaining the effective delay

1. **Approximation of lags** τ_{i0}. The largest of the neglected time constants τ_{i0} is evenly distributed to the remaining time constant and to the delay ("the half rule"), whereas all the smaller time constants are added to the delay.

   That is, to obtain a first-order model ($\tau_2 = 0$) we choose

   $$\tau_1 = \tau_{10} + \frac{\tau_{20}}{2}; \quad \theta = \theta_0 + \frac{\tau_{20}}{2} + \sum_{i\geq 3}\tau_{i0}$$

   and to obtain a second-order delay model we choose

   $$\tau_1 = \tau_{10}; \quad \tau_2 = \tau_{20} + \frac{\tau_{30}}{2}; \quad \theta = \theta_0 + \frac{\tau_{30}}{2} + \sum_{i\geq 4}\tau_{i0}$$

2. **Approximation of negative T_{j0} as effective delay.** We simple add the inverse response time constant $|T_{j0}|$ to the effective delay θ_1 obtained so far,

   $$\theta = \theta_1 + |T_{j0}| = \theta_1 - T_{j0}$$

3. **Approximation of small positive T_{j0} as reduced effective delay.** For $T_{j0} < \theta_1/2$ (approximately) we simply subtract T_{j0} from the delay θ_1 obtained so far,

$$\theta = \theta_1 - T_{j0}$$

4. **Cancellation of large positive T_{j0} by reducing time constant.** For $T_{j0} > \theta_1/2$ (approximately) we cannot subtract T_{j0} from the delay, so we instead cancel it be subtracting it from a larger time constant, e.g. $\frac{T_{j0}s+1}{\tau_{10}s+1} \approx \frac{1}{(\tau_{10}-T_{j0})s+1}$.

The rules follow from the approximations $e^{-\theta s} \approx 1-\theta s$ and $e^{-\theta s} = 1/e^{\theta s} \approx 1/(1+\theta s)$. According to these approximations we should have added the whole of the largest neglected time constant to the effective delay, but this is rather conservative, and the "half rule" is used to get "faster" tunings.

The rules are best understood by considering some examples. Although the half rule has no theoretical basis, the simulations below show that the subsequent application of the SIMC tuning rules (7)-(9) result in good responses with good robustness in all cases.

Example 1. *The second-order process*

$$g_0(s) = 20\frac{1}{(10s+1)(s+1)} \tag{48}$$

is approximated as a first-order delay process ($\tau_2 = 0$) with

$$k = 20; \quad \tau_1 = 10 + 0.5 = 10.5; \quad \theta = 0.5$$

The corresponding SIMC-PI controller tunings with $\tau_c = \theta$ are $K_c = \frac{1}{k}\frac{\tau_1}{\tau_c+\theta} = 0.525$, and $\tau_I = 4(\theta+\tau_c) = 8\theta = 4$. The robustness is good with infinite gain margin, PM= 40.3^o, $M_s = 1.69$, and $M_t = 1.47$.

The model (48) is second-order with $\tau_1 = 10$, $\tau_2 = 1$ and $\theta = 0$, so no further approximations are needed to derive a PID-controller. The SIMC-PID tunings are $K_c = \frac{1}{k}\frac{\tau_1}{\tau_c}, \tau_I = \tau_1 = 10$ and $\tau_D = \tau_2 = 1$. The resulting loop transfer function is $L = gc = \frac{kK_c}{\tau_1 s} = \frac{1}{\tau_c s}$, and, irrespective of the value of τ_c, we get excellent robustness with infinite gain margin, PM=90^o, $M_s = 1$, and $M_t = 1$. In fact, since there is no delay, we may in theory *use $\tau_c = 0$ (infinite controller gain K_c) and achieve perfect responses with excellent robustness. However, in practice K_c will be limited due to uncertainty, unmodelled dynamics and limited input usage.*

Example 2. *The process*

$$g_0(s) = k\frac{(-0.3s+1)(0.08s+1)}{(2s+1)(1s+1)(0.4s+1)(0.2s+1)(0.05s+1)^3} \tag{49}$$

is approximated as a first-order delay process with

$$\tau_1 = 2 + 1/2 = 2.5; \quad \theta = 1/2 + 0.4 + 0.2 + 3\cdot 0.05 + 0.3 - 0.08 = 1.47$$

or as a second-order delay process with

$$\tau_1 = 2; \quad \tau_2 = 1 + 0.4/2 = 1.2; \quad \theta = 0.4/2 + 0.2 + 3\cdot 0.05 + 0.3 - 0.08 = 0.77$$

The corresponding tuning parameters for this process are given in Table 4. The responses with the SIMC tunings are very good as shown in Figure 8. Note that the disturbance responses are almost as good as with the less robust ZN-PID controller.

	$K_c \cdot k$	τ_I	τ_D	GM	PM	M_s	M_t
SIMC-PI	0.85	2.5	0	3.37	57.8^o	1.66	1.04
SIMC-PID	1.30	2	1.2	2.89	57.9^o	1.73	1.04
ZN-PID	1.28	1.32	1.32	2.38	42.1^o	2.08	1.45

Table 4: Example 2. Tunings and robustness for process (49)

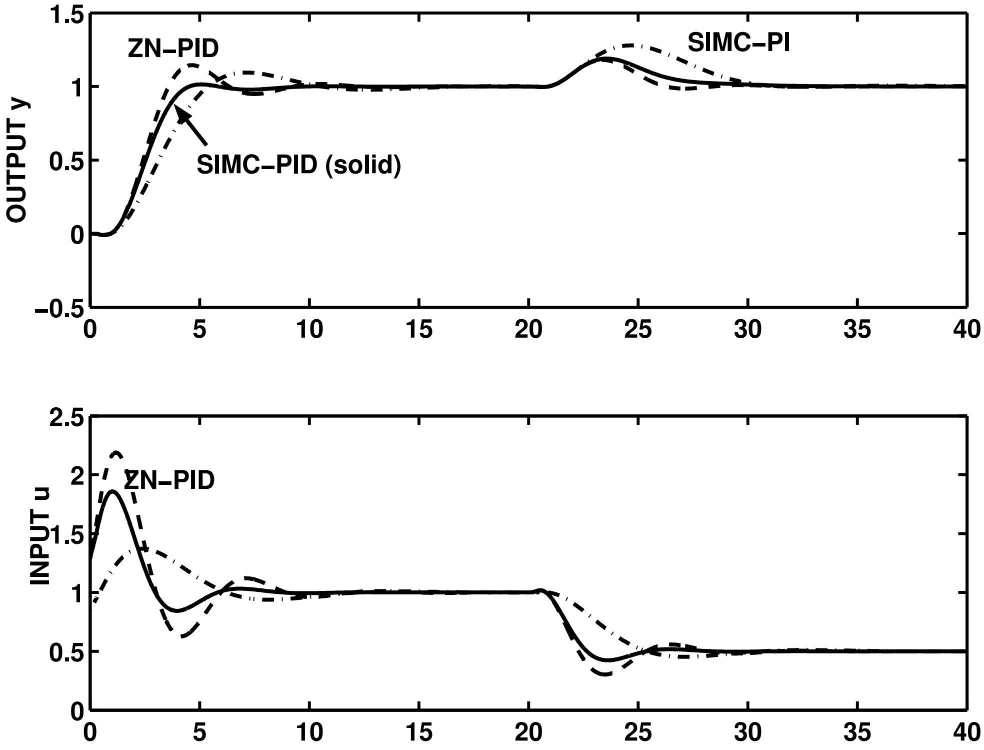

Figure 8: Example 2. Responses for process (49) with tunings from Table 4

Example 3. *The process*

$$g_0(s) = k\frac{(6s+1)(3s+1)e^{-0.3s}}{(10s+1)(8s+1)(s+1)} \tag{50}$$

is approximated as a first-order delay model with

$$\tau_1 = 10 + 8 - 6 - 3 + \frac{1}{2} = 9.5; \quad \theta = 0.3 + \frac{1}{2} = 0.8$$

Because of the presence of the large time constants in the numerator, the approximation of τ_1 *does not quite follow the above rules. The corresponding SIMC-PI controller tunings* $K_c = 5.94/k$ *and* $\tau_I = 6.4$ *result in nice responses (not shown). The robustness is very good with GM=* 4.23*, PM=*93.5^o*,* $M_s = 1.37$*, and* $M_t = 1.02$*. A second-order model (and thus the use of a PID controller) is not recommended here because the initial response is overall first order (with a pole excess of one).*

Example 4. *The process*

$$g_0(s) = k\frac{1}{(\tau_0 s+1)^4} \tag{51}$$

is approximated as a first-order delay process with

$$\tau_1 = 1.5\tau_0; \quad \theta = 2.5\tau_0$$

or as a second-order delay process with (here we interchange τ_1 and τ_2 since we want $\tau_1 > \tau_2$)

$$\tau_1 = 1.5\tau_0; \quad \tau_2 = \tau_0; \quad \theta = 1.5\tau_0$$

The corresponding SIMC PI- and PID-controller tunings are given in Table 5. In this case $\tau_2 < \theta$ and the use of derivative action is not really recommended. Actually, the response (not shown) is best with the PI-controller.

	$K_c \cdot k$	τ_I/τ_0	τ_D/τ_0	GM	PM	M_s	M_t
SIMC-PI	0.3	1.5	0	4.95	62.4^o	1.46	1.00
SIMC-PID	0.5	1.5	1	6.73	62.5^o	1.43	1.00

Table 5: Example 4. Tunings and robustness for process (51)

Example 5. *The process (Astrom* et al. *1998)*

$$g_0(s) = \frac{1}{(s+1)(0.2s+1)(0.04s+1)(0.008s+1)} \tag{52}$$

is approximated as a first-order delay process with

$$\tau_1 = 1.1; \quad \theta = 0.148$$

or as a second-order delay process with

$$\tau_1 = 1.0; \quad \tau_2 = 0.22; \quad \theta = 0.028$$

The corresponding tunings are given in Table 6. As seen in Figure 9 the Ziegler-Nichols PI-tunings almost give instability for this process, whereas the SIMC PI-tunings give nice closed-loop responses. The SIMC PID-controller gives a significant improvement in reponse time, which is expected since the process is dominant second order with $\tau_2 = 0.22$ much larger than $\theta = 0.028$.

	K_c	τ_I	τ_D	GM	PM	M_s	M_t
SIMC-PI	3.72	1.1	0	6.69	51.1^o	1.59	1.16
SIMC-PID	17.9	1.0	0.22	7.54	50.9^o	1.58	1.16
Astrom-PI($M_s = 2$)	4.13	0.59	0	4.90	35.1^o	2.00	1.66
ZN-PI	13.6	0.47	0	1.30	5.5^o	11.3	10.9
ZN-PID	9.07	0.14	0.14	13.0	24.8^o	2.37	2.71

Table 6: Example 5. Tunings and robustness for process (52)

Example 6. *The process (Astrom* et al. *1998)*

$$g_0(s) = \frac{(0.17s+1)^2}{s(s+1)^2(0.028s+1)} \tag{53}$$

is approximated as an integrating process, $e^{-\theta s}/s$, with

$$\theta = 2 \cdot 1 + 0.028 - 2 \cdot 0.17 = 1.69$$

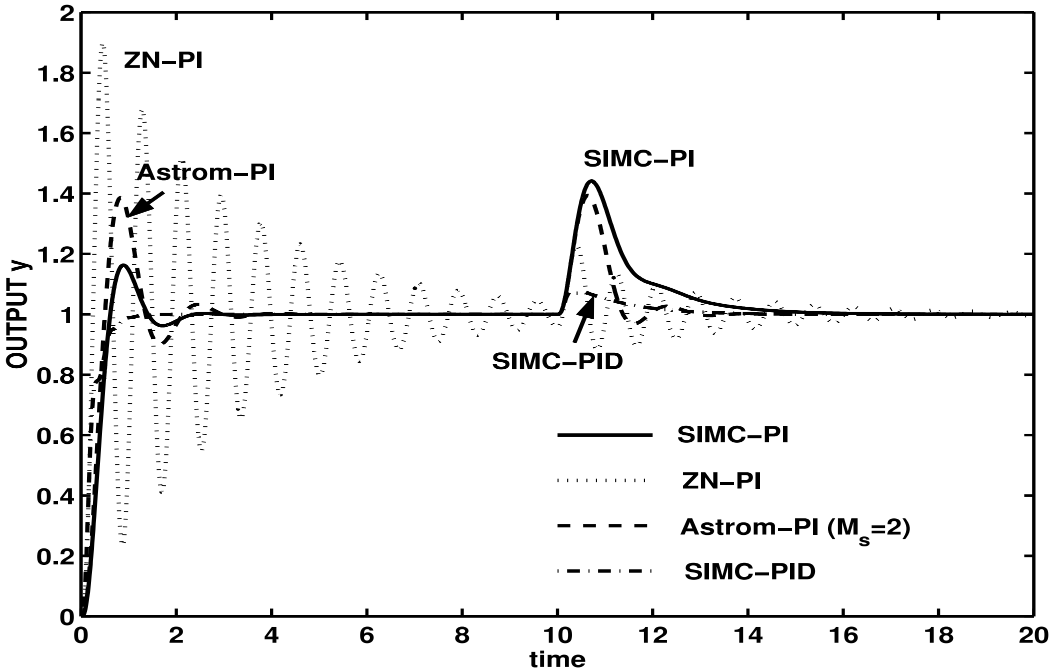

Figure 9: Example 5. Responses for process (52) with tunings from Table 6.
Load disturbance of magnitude 2 occurs at $t = 10$.

	K_c	τ_I	τ_D	GM	PM	M_s	M_t
SIMC-PI	0.296	13.52	0	16.6	48.8^o	1.48	1.29
SIMC-PID	1.397	2.894	1.33	∞	52.4^o	1.23	1.30
Astrom ($M_s = 2$)	0.47	7.01	0	8.2	33.1^o	2.00	1.77

Table 7: Example 6. Tunings and robustness for process (53)

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

$$\tau_2 = 1 + 1/2 - 0.17 = 1.33; \quad \theta = 1/2 + 0.028 - 0.17 = 0.358$$

The corresponding SIMC PI- and PID-controller tunings are given in Table 7. The corresponding closed-loop responses (Figure 10) are again very good, especially for the PID-controller.

In summary, these 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 tunings. The method for approximating a first-order with delay model ("half rule") and the PID tuning rules are not "optimal" in any mathematical sense, but they are simple and give surprisingly good robust performance. Furthermore, the reason for using a PID controller is simplicity, and if high performance control is desired, then one would not use PID control in the first place.

A large number of additional comparisons have been performed, and there has been few cases (if any) where the proposed SIMC tuning rules perform poorly.

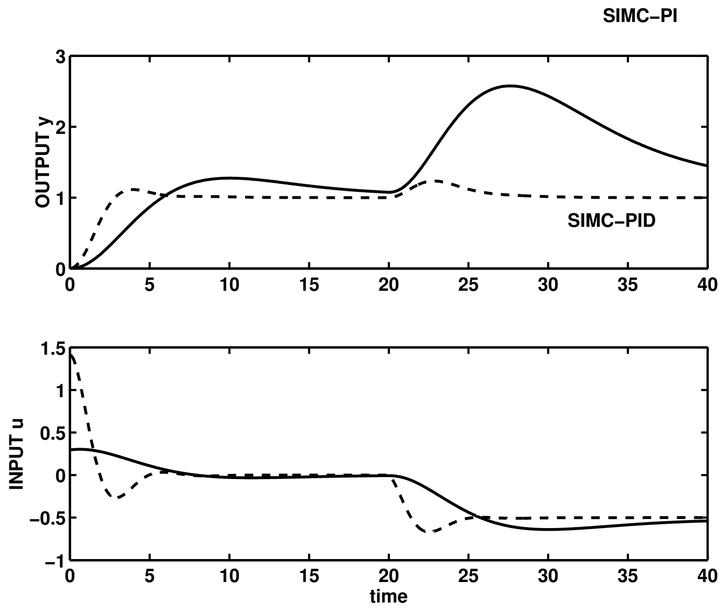

Figure 10: Example 6. Responses for process (53) with tunings from Table 7
Load disturbance of magnitude 2 occurs at $t = 10$.

6 Discussion

6.1 Guidelines for retuning

The tuning rules presented in this paper, see (3)-(5), give invalueable insights, for example, into how we must change the tuning parameters in response to changes in the process model:

1. An increase in the process gain k is counteracted by reducing the controller gain K_c such that $K_c k$ remains constant. (The integral time is kept constant, and the closed-loop response will remain unchanged unless there is also a change in the disturbance transfer function).

2. An increase in the process time constant τ_1 is counteracted by increasing K_c such that K_c/τ_1 remains constant. For a "fast" process where we use $\tau_I = \tau_1$, we also need to incrase the integral time (the closed-loop response will then remain unchanged). For a "slow" process where we use $\tau_I = 4(\tau_c + \theta)$, we keep τ_I unchanged (but the closed-loop response will change somewhat in this case).

3. In many cases there is a direct correlation between the gain and the time constant such that the initial slope $k' = k/\tau_1$ remains constant. In this case we should keep K_c constant. For "fast" processes where we use $\tau_I = \tau_1$ we should increase the integral time. For "slow" processes where we use $\tau_I = 4(\tau_c + \theta)$ we should keep the integral time constant.

4. Note that for a "slow" process, the tunings only depend on the initial response as expressed by $k' = k/\tau_1$ and θ, whereas for a "fast" process the steady-state gain k is also of importance.

5. An increase in the delay θ is counteracted by a corresponding decrease in K_c in order to maintain the same robustness. For "fast" processes with $\tau_I = \tau_1$ the integral time is kept unchanged, whereas for "slow" processes with $\tau_I = 4(\tau_c + \theta)$ the integral time is increased.

6. For a second order process the derivative time increases when the second order time constant τ_2 is increased.

When retuning the controller based on experimental responses the following guidelines for PI control may prove helpful. The basis for these guidelines is the disturbance response.

1. If the maximum output deviation is too large then the controller gain should be increased - recall (33).

2. If the settling time is too large then the integral time should be reduced.

3. If the oscillations are too large and these have a period shorter than the integral time τ_I, then the gain should be reduced or the integral time increased - recall Figure 11.

4. If the oscillations are too large and these have a period more than about three times the integral time τ_I, then the product of the controller gain and integral time should be increased, recall (27).

6.2 Period of slow and fast oscillations

We get slow oscillations if the product of the controller gain K_c and the integral time τ_I is reduced compared to the value given in (27). What is the period P of these "slow" oscillations? From a standard analysis of second-order systems, we have that (e.g. Seborg *et al.* (1989) page 118)

$$P = \frac{2\pi}{\sqrt{1-\zeta^2}}\,\tau_0 > 2\pi\tau_0 = 2\pi\sqrt{\frac{\tau_I}{k'K_c}} \tag{54}$$

where the inequality applies since the presence of oscillations requires $\zeta \leq 1$. With the suggested tuning $\tau_I = 4/k'K_c$ (28) this gives

$$P > \pi \cdot \tau_I \tag{55}$$

Thus, the "slow" oscillations which result by *reducing* the controller gain have a period larger than 3 times the integral time[5]. On the other hand, the "usual" fast oscillations that appear by *increasing* the controller gain have a period of about 6 times the delay. This is illustrated in Figure 11 for a "slow" process with $\tau_1 = 30$, $\theta = 1$ and integral time $\tau_I = 4$:

- $K_c = 15$ gives no apparent oscillations.

- Increasing the controller gain ($K_c = 30$) gives "fast" oscillations with a period of about 6 (about 6 times the delay).

- Decreasing the controller gain ($K_c = 3$) gives "slow" oscillations with a period of about 30 (larger than 3 times the integral time).

In summary, the slow oscillations, with a period larger than 3 times the integral time, are caused by the presence of integral action and too *low* controller gain. On the other hand, the more common fast oscillations, with a period of about 6 times the delay, are caused by the presence of time delay and too *high* controller gain. The suggested tuning rules in this paper avoid both these two kinds of oscillations.

[5] The corresponding normalized frequency of these slow oscillations is $\tau_1\omega = \tau_1 \cdot 2\pi/P \approx 2\tau_1/\tau_I$ which is larger than 2 since we have $\tau_I \leq \tau_1$.

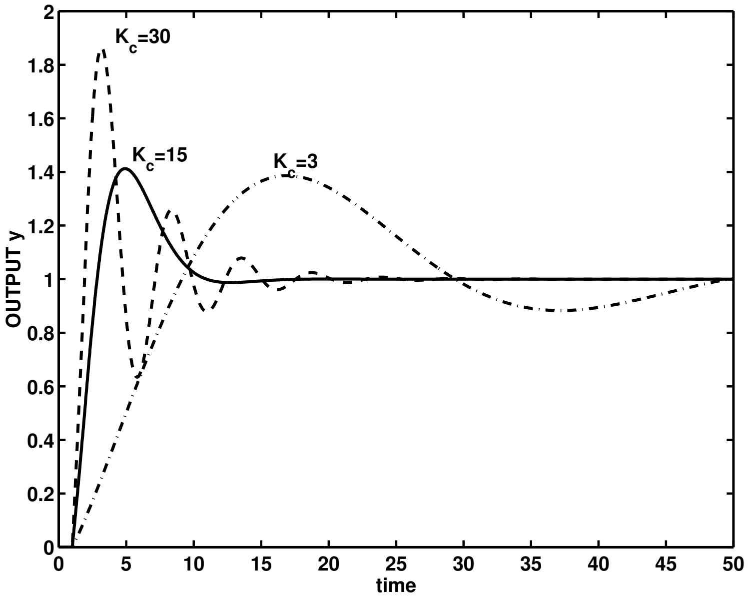

Figure 11: Effect of changing the gain K_c for PI-control of "slow" process $g(s) = e^{-s}/(30s+1)$ with $\tau_I = 4$.
Setpoint responses.

6.3 Retuning of oscillating integrating process

Many control loops for integrating processes, including many liquid level control systems, have oscillations because the controller gain is too low, or alternatively, the integral time is too short. Here we show how to retune the controller in such cases.

Consider a PI controller with (initial) tunings K_{c0} and τ_{I0} which results in "slow" oscillations with period P_0 (larger than $3\tau_{I0}$). Then we most likely have an integrating process

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

for which the product of the controller gain and integral time ($K_{c0}\tau_{I0}$) is too low. Assuming $\zeta^2 << 1$ (significant oscillations), (54) gives the following approximate expression for P_0

$$P_0 \approx 2\pi\tau_0 = 2\pi\sqrt{\frac{\tau_{I0}\ \tau_1}{k'\ K_{c0}}} \tag{56}$$

Thus, from (56) the product of the controller gain and integral time is approximately

$$K_{c0}\tau_{I0} = (2\pi)^2 \frac{\tau_1}{k'} \left(\frac{\tau_{I0}}{P_0}\right)^2$$

To avoid oscillations ($\zeta \geq 1$) we must from (28) require

$$K_c \tau_I \geq 4 \frac{\tau_1}{k'}$$

that is, we must require that

$$\frac{K_c\tau_I}{K_{c0}\tau_{I0}} \geq \frac{1}{\pi^2} \cdot \left(\frac{P_0}{\tau_{i0}}\right)^2 \tag{57}$$

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 at least a factor $f \approx 0.1(P_0/\tau_{I0})^2$.

The application of this simple rule should guarantee you immediate success and respect among plant operators.

Example. *A real industrial case study of a reboiler level control loop is shown in Figure 12. Here y is the reboiler level and u is the bottoms flow valve position. The PI tunings had been kept at their default setting ($K_c = -0.5$ and $\tau_I = 1$ min) since start-up several years ago, and resulted in an oscillatory response as shown in the top part of the Figure. The control of the level (y) itself was acceptable, but the bottoms flowrate (input u) showed large variations, and because it is the feed to the downstream column this caused poor temperature control in the downstream column.*

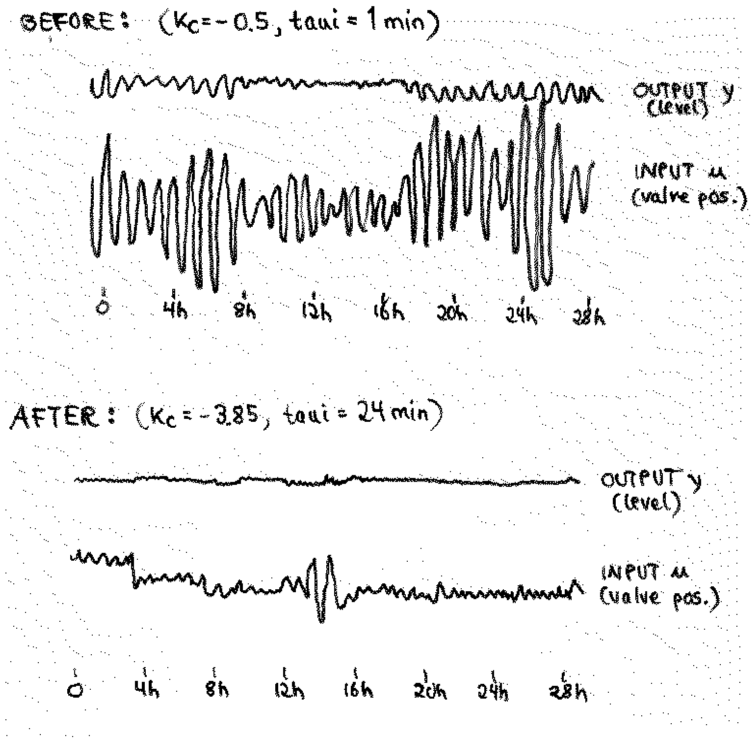

Figure 12: Industrial case study of retuning reboiler level control system

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 $\tau_I = 1$ min, we get from the above rule we should increase $K_c \cdot \tau_I$ by a factor $f = 0.1 \cdot (51)^2 = 260$ 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_I by a factor 24, that is, $K_c\tau_I$ was increased by $7.7 \cdot 24 = 185$. The much improved response is shown in the "after" plot in Figure 12. 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, and the plant personnel were very impressed by what the fresh engineer had learned in her control course in Trondheim.

6.4 Should we use derivative action to counteract the delay?

Derivative action, e.g. $\tau_D = \theta/2$, is commonly used to improve the response when we have a delay. To derive this value, we may instead of (23) use the more exact 1st order Pade approximation,

$$e^{-\theta s} = \frac{e^{-\theta/2s}}{e^{-\theta/2s}} \approx \frac{-\frac{\theta}{2}s + 1}{\frac{\theta}{2}s + 1}$$

With the choice $\tau_c = \theta$ this results in the same PID-controller (24) found above, but in addition we get a term

$$\frac{\frac{\theta}{2}s + 1}{0.5\frac{\theta}{2}s + 1} \tag{58}$$

This is as an additional derivative term, effective over only a very small range, which increases the controller gain by a factor 2 at high frequencies. However, simulations in Figure 13 show that performance is only moderately improved by adding this term, and thus does not justify the increased complexity of the controller,

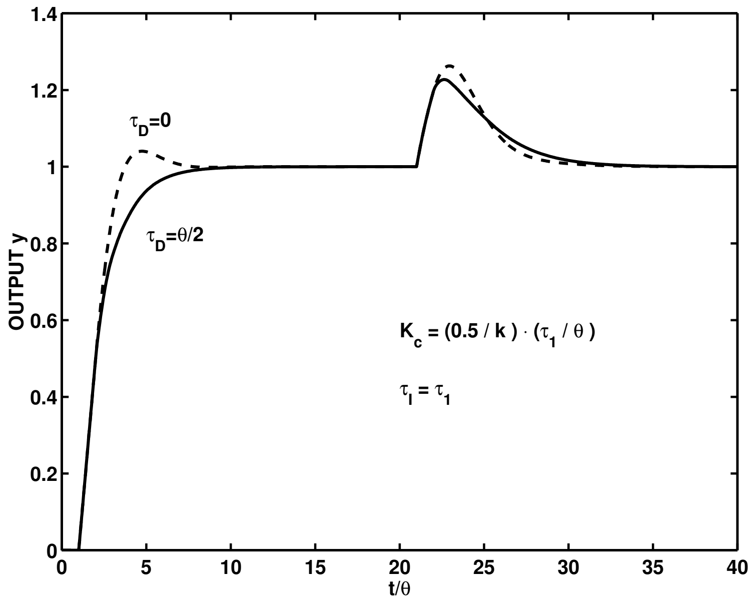

Figure 13: Introduction of derivative action (solid line) has only a minor effect for first-order with delay process $g(s) = ke^{-\theta s}/(\tau_1 + 1)$.
Controller gain corresponds to $\tau_c = \theta$ in (25). Responses shown for plant with $\tau_1 = 2$ and $\theta = 1$. Setpoint change at $t = 0$. Load disturbance of magnitude 0.5 occurs at $t = 20$.

7 Conclusion

A two-step procedure is proposed for deriving PID tunings for typical chemical processes.

1. The half rule is used to approximate the process as a first or second order process with effective delay.

2. Based on this process model with parameters k, τ_1 and θ the following SIMC tunings are suggested ($k' = k/\tau_1$):

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

If the process is second order (with $\tau_2 > \theta$, approximately) we may add derivative action with

$$\text{Cascade PID}: \quad \tau_D = \tau_2$$

The closed-loop time constant τ_c is the only tuning parameter, and a reasonably fast response with good robustness is obtained with $\tau_c = \theta$.

The above tuning rules are derived analytically and are thus very suitable for teaching.

The value $\tau_c = \theta$ gives robust (conservative) tunings when compared with most other tuning rules. To get a faster response one may decrease the value of τ_c, and possibly further reduce the integral time. However, one may also want to increase τ_c to get a slower and smoother response with less input usage. This results in a smaller controller gain K_c, but we must require

$$K_c \geq d_u/y_{max}$$

(approximately) in order to keep the output deviation less than y_{max} in response to a load disturbance of magnitude d_u.

We found that the Ziegler-Nichols tuning rules result in a very good load disturbance response for an integrating processes. However, the robustness is generally poor, and for many processes the tunings give instability.

Acknowledgement

Discussions with Professor David Clough from the University of Colorado at Boulder are gratefully acknowledged. Further examples are given in Holm and Butler (1998).

Simulations

In all simulations we have used a cascade PID controller with derivatice action effective over one decade ($\alpha = 0.1$), and to avoid "derivative kick" we do not differentiate the setpoint:

$$u(s) = K_c \frac{\tau_I s + 1}{\tau_I s} \left(y_s(s) - \frac{\tau_D s + 1}{\alpha \tau_D s + 1} y(s) \right) \tag{59}$$

However, note that stability margins etc. are computed with $\alpha = 0$. In most cases we use a PI controller, that is $\tau_D = 0$, and the controller becomes

$$u(s) = K_c \frac{\tau_I s + 1}{\tau_I s} \left(y_s(s) - y(s) \right) \tag{60}$$

or in the time domain

$$\Delta u(t) = K_c \left((y_s(t) - y(t)) + \frac{1}{\tau_I} \int_0^t (y_s(\tau) - y(\tau)) d\tau \right) \tag{61}$$

In the simulations a unit setpoint change $y_s = 1$ is introduced at time $t = 0$, and an input "load" disturbance of magnitude $d_u = 0.5$ occurs at $t = 20$ (unless otherwise stated).

References

Astrom, K.J., H. Panagopoulos and T. Hagglund (1998). Design of PI controllers based on non-convex optimization. *Automatica* **34**(5), 585–601.

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

Cohen, G.H. and G.A. Coon (1953). Theoretical consideration of retarded control. *Trans. ASME* **75**, 827–834.

Ho, W.K., K.W. Lim and W. Xu (1998). Optimal gain and phase margin tuning for PID controllers. *Automatica* **34**(8), 1009–1014. See Automatica.

Holm, O. and A. Butler (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.chembio.ntnu.no/users/skoge/diplom/reports/pid98-holm-butler/.

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.

Marlin, T.E. (1995). *Process Control.* McGraw-Hill.

Rivera, D.E., M. Morari and S. Skogestad (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., T.F. Edgar and D.A. Mellichamp (1989). *Process Dynamics and Control.* John Wiley & Sons.

Shinskey, F.G. (1998). Personal communication at Nordic Process Control Workshop in Stockholm.

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.

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