Simple rules for PID tuning

Sigurd Skogestad
NTNU, Trondheim, Norway

Summary

- Main message: Can usually do much better by taking a systematic approach
- Key: Look at <u>initial part</u> of step response

  Initial slope: k’ = k/ $_1$
- SIMC tuning rules (“Skogestad IMC”)$^{(*)}$

  One tuning rule! Easily memorized

$$K_c = \frac{1}{k'} \cdot \frac{1}{(\theta + \tau_c)}$$

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

$_c \geq$ 0: desired closed-loop response time (tuning parameter)

For robustness select: $_c \geq$

Reference: S. Skogestad, “Simple analytic rules for model reduction and PID controller design”, *J.Proc.Control*, Vol. 13, 291-309, 2003

(*) “Probably the best simple PID tuning rules in the world”

Need a model for tuning

- Model: Dynamic effect of change in input u (MV) on output y (CV)
- First-order + delay model for PI-control

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

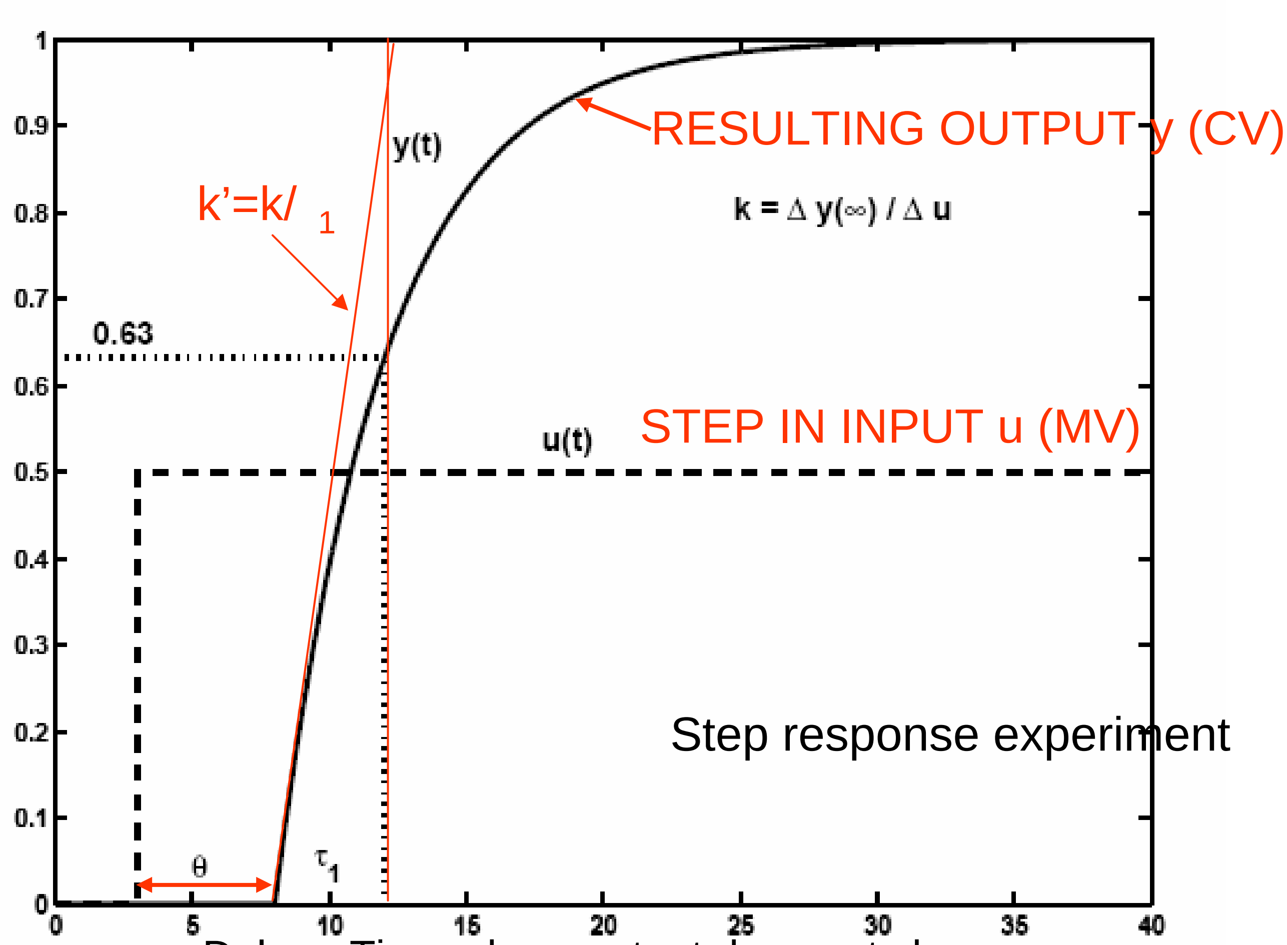
- Second-order model for PID-control

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

Step response experiment

- Make step change in one u (MV) at a time
- Record the output (s) y (CV)

First-order plus delay process



Delay - Time where output does not change

$_1$: Time constant - Additional time to reach 63% of final change

k : steady-state gain = y(∞)/ u

k’ : slope after response “takes off” = k/ $_1$

Model reduction of more complicated model

- Start with complicated stable model on the form

$$G_0(s) = k_0 \frac{(T_{10}s+1)(T_{20}s+1)\cdots}{(\tau_{10}s+1)(\tau_{20}s+1)\cdots} \; e^{-\theta_0 s}$$

- Want to get a simplified model on the form

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

- Most important parameter is usually the “effective” delay

OBTAINING THE EFFECTIVE DELAY θ

Basis (Taylor approximation):

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

Effective delay =

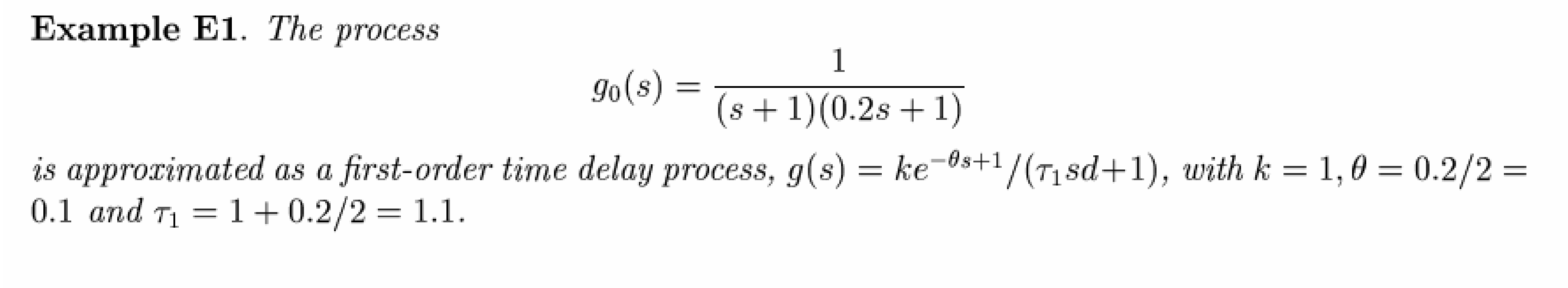
"true" delay

+ inverse reponse time constant(s)

+ half of the largest neglected time constant (the "half rule")
(this is to avoid being too conservative)

+ 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 use second-order model).



Example E1. *The process*

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

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

Example

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

half rule

is approximated as a first-order delay process with

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

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

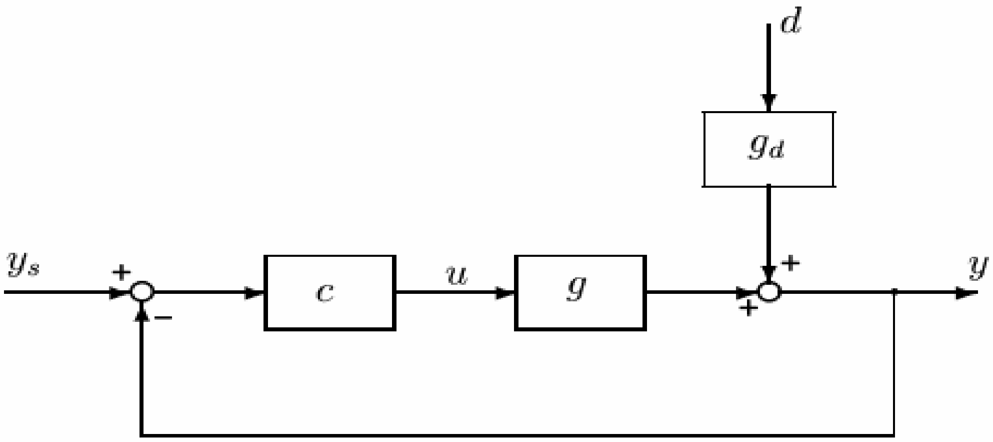
or as a second-order delay process with

$\tau_1 = 2$

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

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

Deriv ation of rules:
Direct synthesis (IMC)



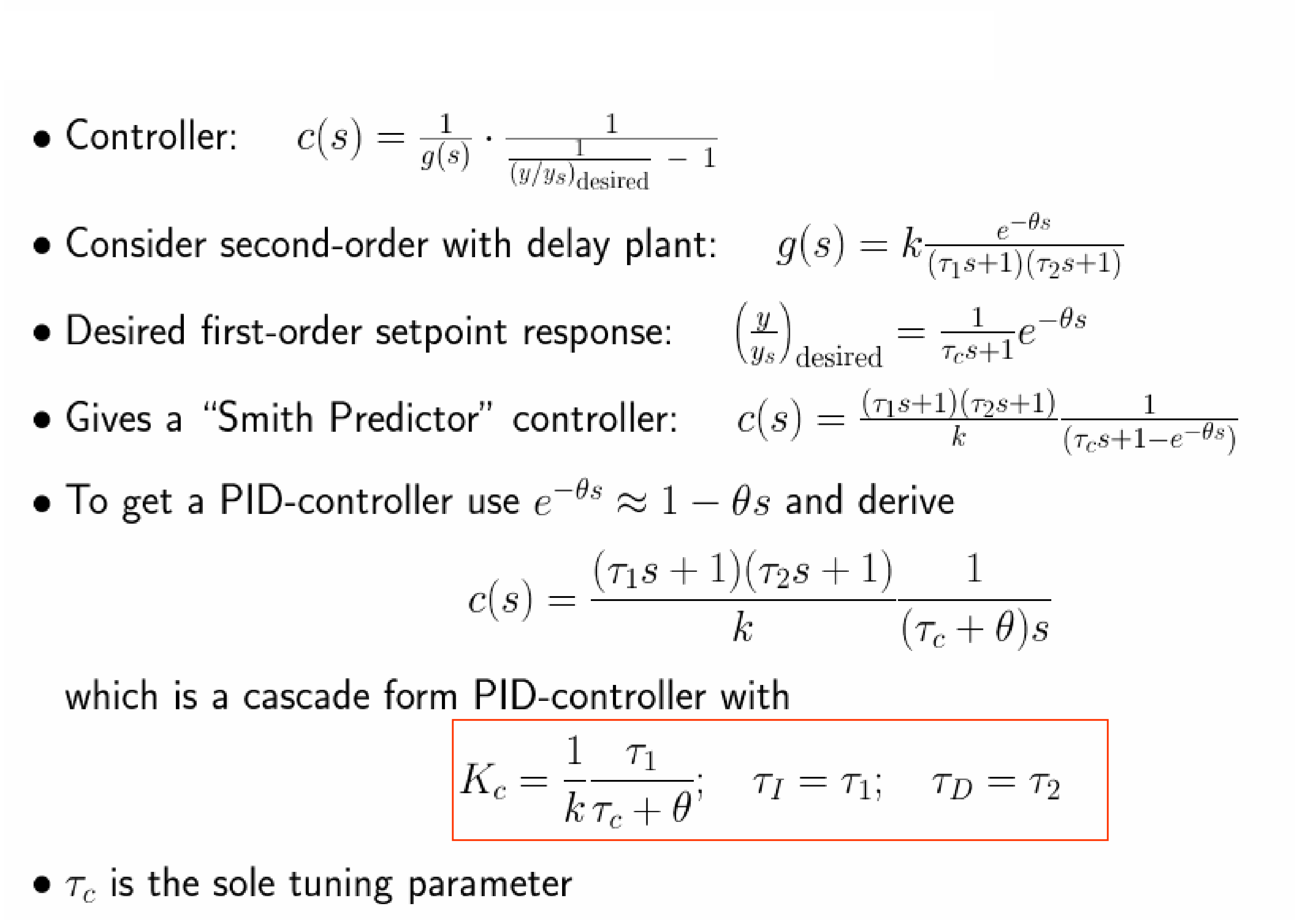
Closed-loop response to setpoint change

$$y = T \; y_s; \;\; T(s) = \frac{gc}{1+gc}$$

Idea: Specify desired response (y/y_s)=T and from this get the controller. Algebra:

$$c = \frac{1}{g} \cdot \frac{1}{\frac{1}{T}-1}$$

IMC Tuning = Direct Synthesis



- Controller: $c(s) = \frac{1}{g(s)} \cdot \frac{1}{\frac{1}{(y/y_s)_{\text{desired}}} - 1}$
- Consider second-order with delay plant: $g(s) = k\frac{e^{-\theta s}}{(\tau_1 s+1)(\tau_2 s+1)}$
- Desired first-order setpoint response: $\left(\frac{y}{y_s}\right)_{\text{desired}} = \frac{1}{\tau_c s+1}e^{-\theta s}$
- Gives a “Smith Predictor” controller: $c(s) = \frac{(\tau_1 s+1)(\tau_2 s+1)}{k}\frac{1}{(\tau_c s+1-e^{-\theta s})}$
- To get a PID-controller use $e^{-\theta s} \approx 1 - \theta s$ and derive

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

which is a cascade form PID-controller with

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

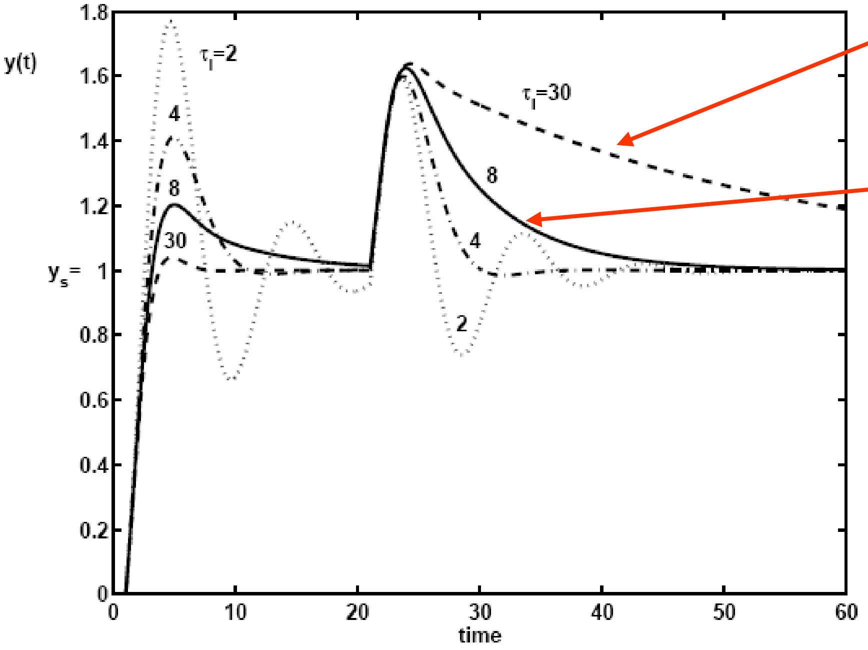
- τ_c is the sole tuning parameter

Integral time

- Found:

  Integral time = dominant time constant ($_{I}$ = $_{1}$)
- Works well for setpoint changes
- Needs to be modify (reduce) $_{I}$ for “integrating disturbances”

Example: Integral time for "slow"/integrating process



IMC rule:

$_I$ = $_1$ =30

- Reduce $_I$ to improve performance
- To just avoid slow oscillations:

$_I$ = 4 ($_c$+) = 8

(see derivation next page)

Figure 2: 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 $t = 20$.

Too large integral time: Poor disturbance rejection

Too small integral time: Slow oscillations

Derivation integral time: Avoiding slow oscillations for integrating process

.

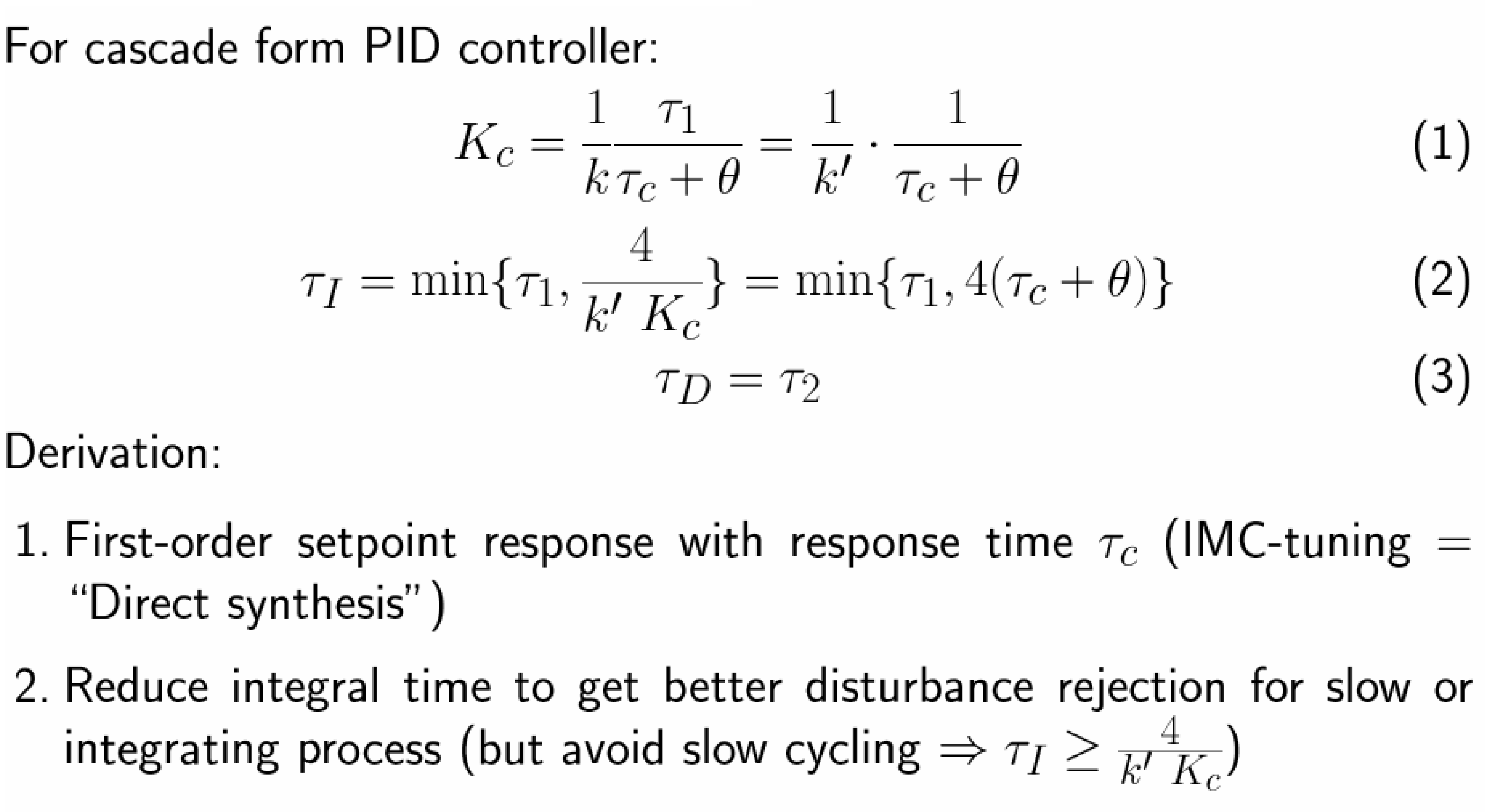
- Integrating process: $_1$ large
- Assume $_1$ large and neglect delay
  - G(s) = k e$^{-\quad s}$ /($_1$ s + 1) ≈ k/($_1$;s) = k'/s
- PI-control: C(s) = K$_c$ (1 + 1/ $_I$ s)
- Poles (and oscillations) are given by roots of closed-loop polynomial
  - 1+GC = 1 + k'/s · K$_c$(1+1/ $_I$ s) = 0
  - or $_I$ s^2 + k' K$_c$ $_I$ s + k' K$_c$ = 0
  - Can be written on standard form ($_0^2$ s^2 + 2 $_0$ s + 1) with

$$\tau_0 = \sqrt{\tau_I/(k'K_c)}; \zeta = \frac{1}{2}\sqrt{K_c \cdot k' \cdot \tau_I}$$

- To avoid oscillations must require | |≥ 1:
  - K$_c$ · k' · $_I$ ≥ 4 or $_I$ ≥ 4 / (K$_c$ k')
  - With choice K$_c$ = (1/k') (1/($_c$+)) this gives $_I$ ≥ 4 ($_c$+)

- Conclusion integrating process: Want $_I$ small to improve performance, but must be larger than 4 ($_c$+) to avoid slow oscillations

Summary: SIMC-PID Tuning Rules



For cascade form PID controller:

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

$$\tau_I = \min\{\tau_1, \frac{4}{k'\ K_c}\} = \min\{\tau_1, 4(\tau_c + \theta)\} \tag{2}$$

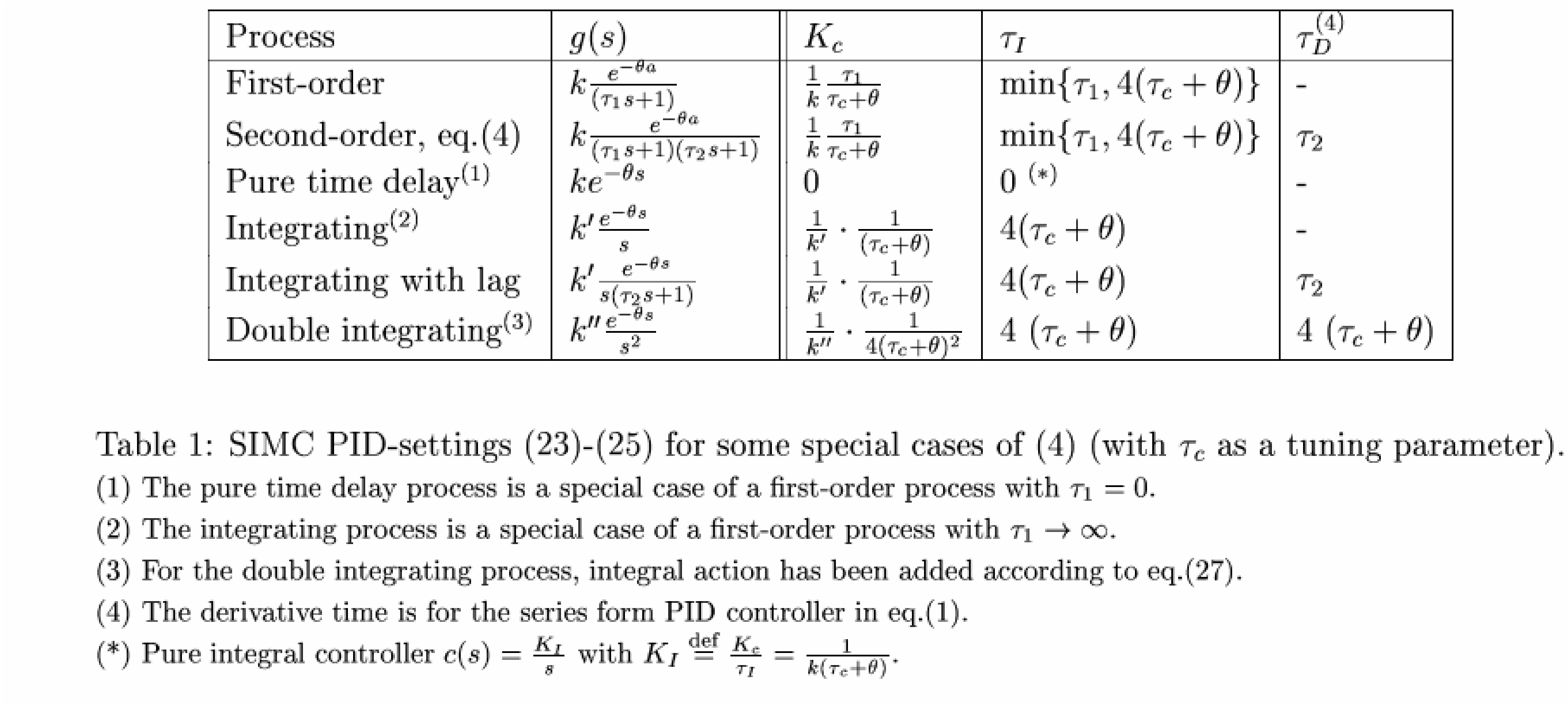
$$\tau_D = \tau_2 \tag{3}$$

Derivation:

1. First-order setpoint response with response time τ_c (IMC-tuning = “Direct synthesis”)
2. Reduce integral time to get better disturbance rejection for slow or integrating process (but avoid slow cycling $\Rightarrow \tau_I \geq \frac{4}{k'\ K_c}$)

One tuning parameter: c

Some special cases



Process	$g(s)$	K_c	τ_I	$\tau_D^{(4)}$
First-order	$k\frac{e^{-\theta a}}{(\tau_1 s+1)}$	$\frac{1}{k}\frac{\tau_1}{\tau_c+\theta}$	$\min\{\tau_1, 4(\tau_c+\theta)\}$	-
Second-order, eq.(4)	$k\frac{e^{-\theta a}}{(\tau_1 s+1)(\tau_2 s+1)}$	$\frac{1}{k}\frac{\tau_1}{\tau_c+\theta}$	$\min\{\tau_1, 4(\tau_c+\theta)\}$	τ_2
Pure time delay(1)	$ke^{-\theta s}$	0	0 (*)	-
Integrating(2)	$k'\frac{e^{-\theta s}}{s}$	$\frac{1}{k'}\cdot\frac{1}{(\tau_c+\theta)}$	$4(\tau_c+\theta)$	-
Integrating with lag	$k'\frac{e^{-\theta s}}{s(\tau_2 s+1)}$	$\frac{1}{k'}\cdot\frac{1}{(\tau_c+\theta)}$	$4(\tau_c+\theta)$	τ_2
Double integrating(3)	$k''\frac{e^{-\theta s}}{s^2}$	$\frac{1}{k''}\cdot\frac{1}{4(\tau_c+\theta)^2}$	$4\ (\tau_c+\theta)$	$4\ (\tau_c+\theta)$

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

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

(2) The integrating process is a special case of a first-order process with $\tau_1 \to \infty$.

(3) For the double integrating process, integral action has been added according to eq.(27).

(4) The derivative time is for the series form PID controller in eq.(1).

(*) Pure integral controller $c(s) = \frac{K_I}{s}$ with $K_I \overset{\text{def}}{=} \frac{K_c}{\tau_I} = \frac{1}{k(\tau_c+\theta)}$.

One tuning parameter: $_c$

DERIVATIVE ACTION ?

First order with delay plant ($\tau_2 = 0$) with $\tau_c = \theta$:

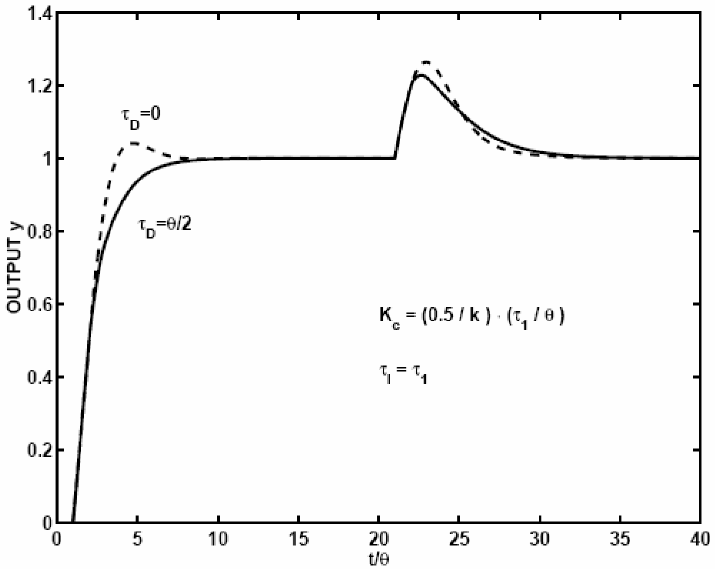


Figure 5: Setpoint change at $t = 0$. Load disturbance of magnitude 0.5 occurs at $t = 20$.

- Observe: Derivative action (solid line) has only a minor effect.
- Conclusion: Use second-order model (and derivative action) only when $\tau_2 > \theta$ (approximately)

Note: Derivative action is commonly used for temperature control loops. Select $_D$ equal to time constant of temperature sensor

Selection of tuning parameter $_c$

Two cases

1. Tight control: Want “fastest possible control” subject to having good robustness

2. Smooth control: Want “slowest possible control” subject to having acceptable disturbance rejection

TIGHT CONTROL

TUNING FOR FAST RESPONSE WITH GOOD ROBUSTNESS

$$\text{SIMC}: \quad \tau_c = \theta \tag{4}$$

Gives:

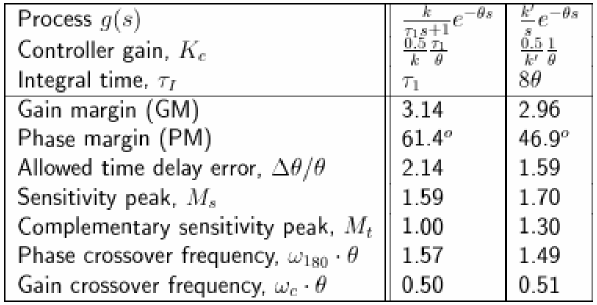
$$K_c = \frac{0.5}{k}\frac{\tau_1}{\theta} = \frac{0.5}{k'} \cdot \frac{1}{\theta} \tag{5}$$

$$\tau_I = \min\{\tau_1, 8\theta\} \tag{6}$$

$$\tau_D = \tau_2 \tag{7}$$

Try to memorize!

Gain margin about 3



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

Table 1: Robustness margins for first-order and integrating delay process using SIMC-tunings in (5) and (6) ($\tau_c = \theta$). The same margins apply to second-order processes if we choose $\tau_D = \tau_2$.

TIGHT CONTROL

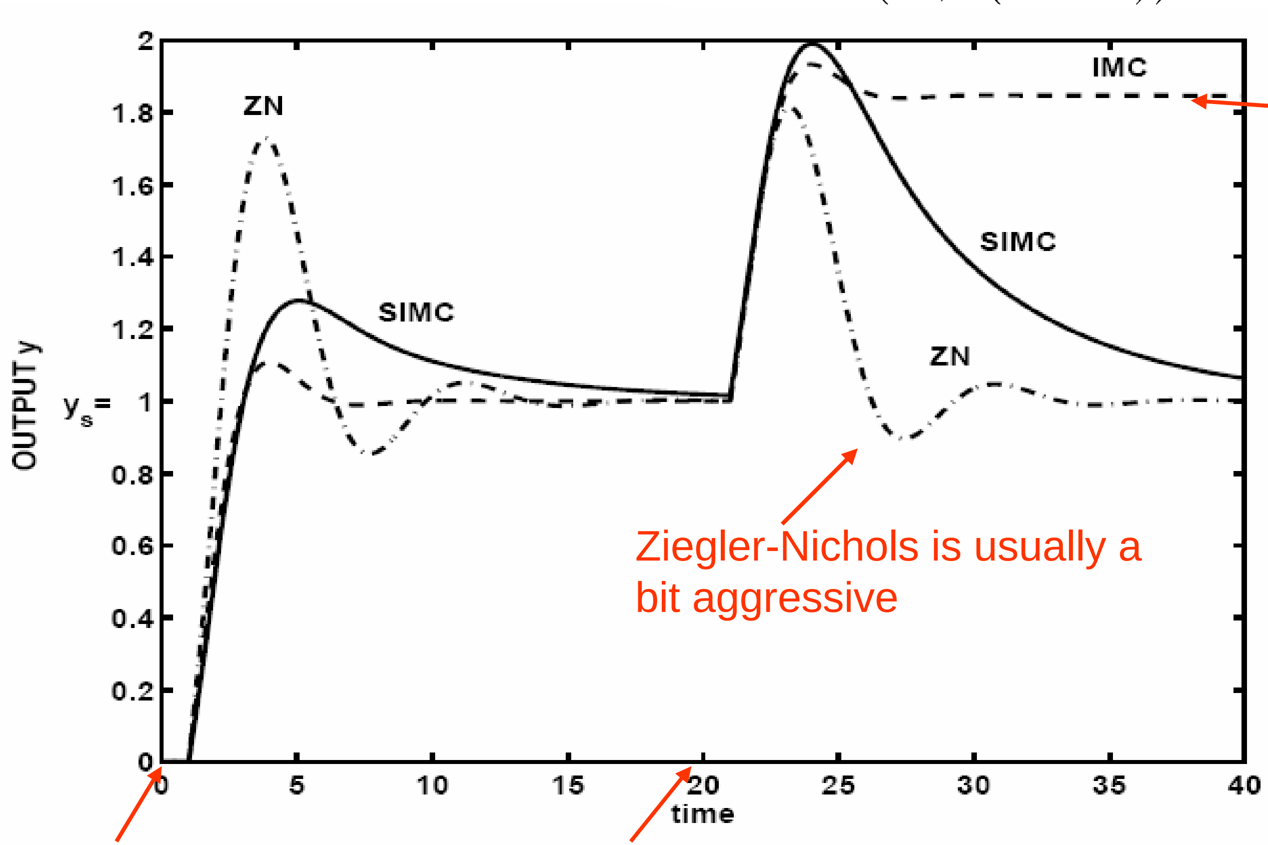
Example. Integrating process with delay=1. G(s) = e^{-s}/s.

Model: k’=1,   =1, $_1$=∞

SIMC-tunings with $_c$ with = =1:

$$K_c = \frac{1}{k'}\frac{1}{\tau_c+\theta} = 1\cdot\frac{1}{1+1} = 0.5$$

$$\tau_I = \min(\tau_1, 4(\tau_c+\theta)) = \min(\infty, 8) = 8$$



IMC has $_I$=∞

Ziegler-Nichols is usually a bit aggressive

Setpoint change at t=0

Input disturbance at t=20

SMOOTH CONTROL

Minimum controller gain:

$$|c(j\omega)| \geq \frac{|u_0(j\omega)|}{|y_{\max}|}$$

Industrial practice: Variables (instrument ranges) often scaled such that

$$|u_0| \approx |y_{\max}| = \text{max range} \quad \text{(span)}$$

Minimum controller gain is then

$$|c(j\omega)| \geq 1 \Rightarrow \quad \boxed{K_c \geq 1} \text{ for PID-control}$$

Minimum gain for smooth control ⇒
Common default factory setting K_c=1 is reasonable !

Level control is often difficult...

- Typical story:
  - Level loop starts oscillating
  - Operator detunes by decreasing controller gain
  - Level loop oscillates even more
  - ……
- ???
- Explanation: Level is by itself unstable and requires control.

How avoid oscillating levels?

- Simplest: Use P-control only (no integral action)
- If you insist on integral action, then make sure the controller gain is sufficiently large
- If you have a level loop that is oscillating then use *Sigurds rule* (can be derived):

To avoid oscillations, increase $K_c \cdot \tau_I$ by factor

$$f = 0.1 \cdot (P_0/\tau_{I0})^2$$

where

P_0 = period of oscillations [s]

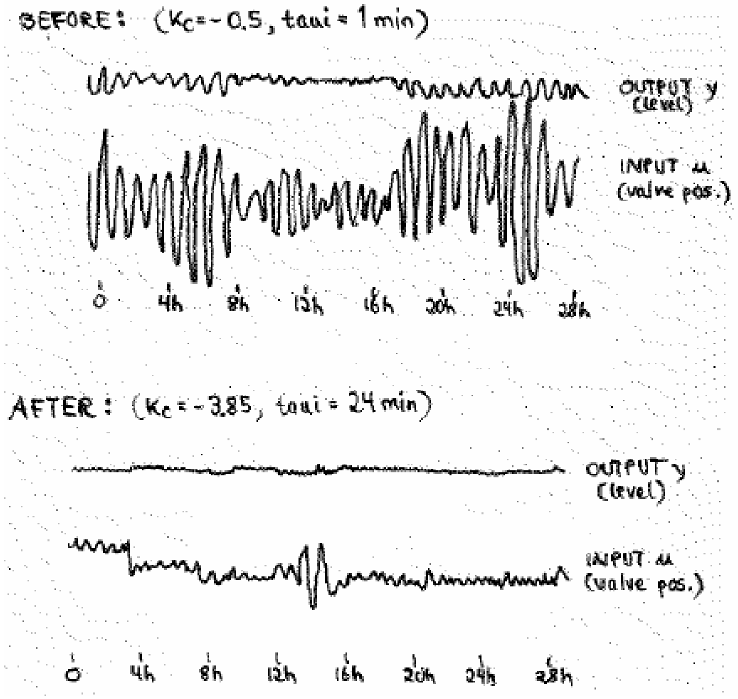
τ_{I0} = original integral time [s]

APPLICATION: RETUNING FOR INTEGRATING PROCESS

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

Real Plant data:

$$\text{Period of oscillations } P_0 = 0.85h = 51min \Rightarrow f = 0.1 \cdot (51/1)^2 = 260$$



Conclusion PID tuning

SIMC tuning rules

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

1. Tight control: Select $\tau_c = \theta$ corresponding to

$$K_{\mathrm{c,max}} = \frac{0.5}{k'} \frac{1}{\theta}$$

2. Smooth control. Select $K_c \geq$ $K_{\mathrm{c,min}} = \frac{|u_0|}{|y_{\mathrm{max}}|}$

Note: Having selected K_c (or τ_c), the integral time τ_I should be selected as given above

CONCLUSION

- It is simple (one single rule for all processes)
- It is excellent for teaching (analytical)
- It works very well for all of “our” processes