PID Tuning using the SIMC rules

Sigurd Skogestad
NTNU, Trondheim, Norway

MODEL

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

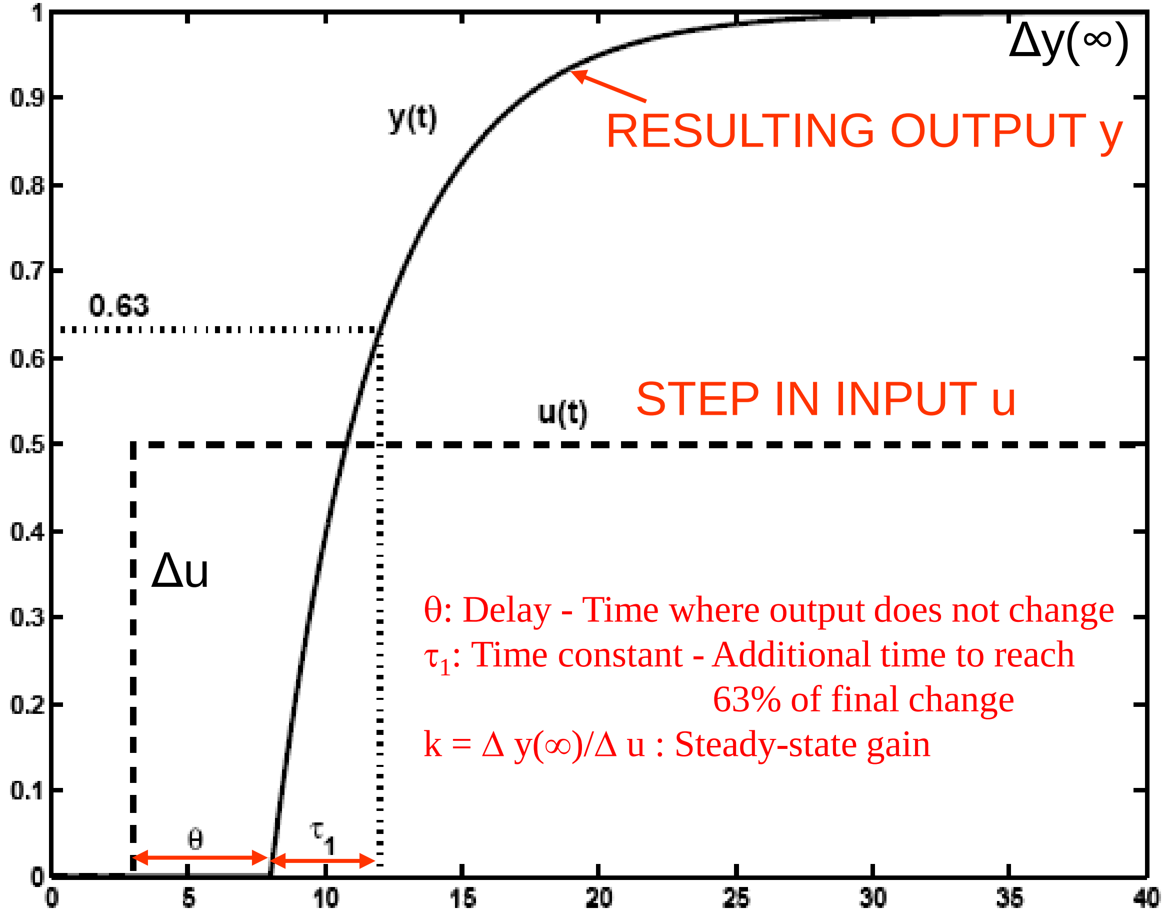
$$G(s) = \frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)} e^{-\theta s}$$

  - Recommend: Use second-order model (PID control) only if $\tau_2 > \theta$

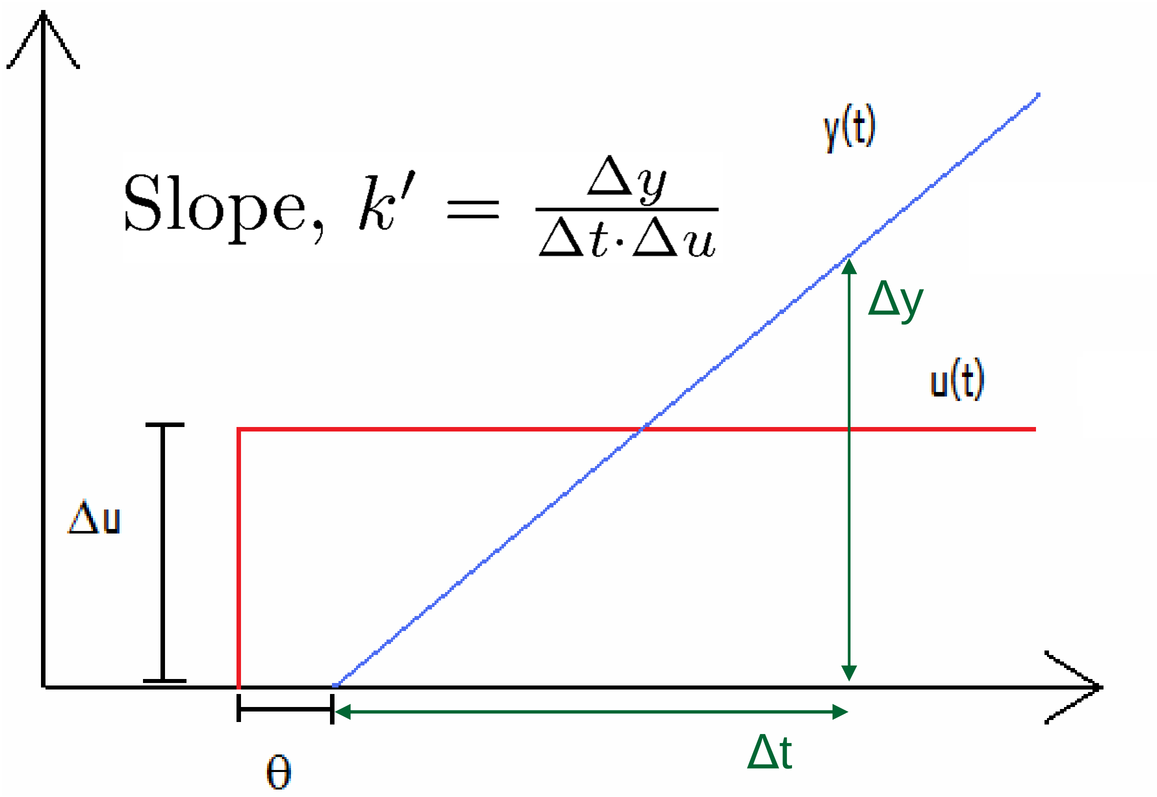
1. Step response experiment

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

MODEL, Approach 1



Step response integrating process



2. 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 the “effective” delay θ
- Use second-order model only if $\tau_2 > \theta$

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

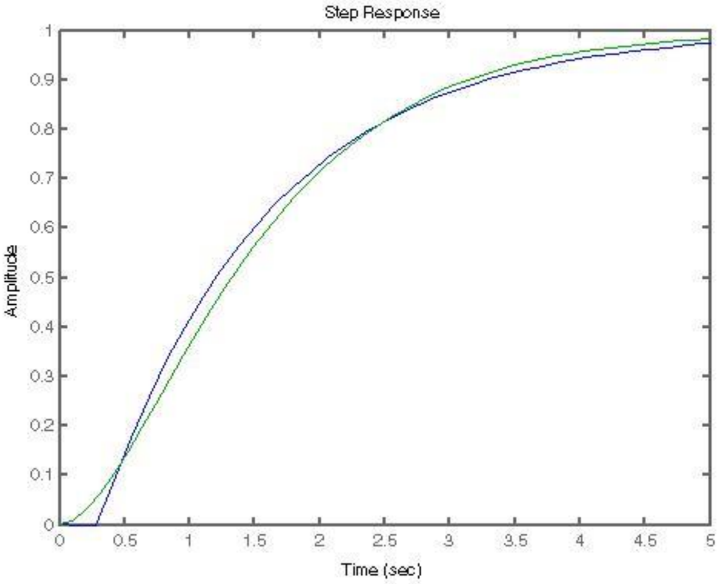
+ inverse reponse time constant(s)

+ 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



The second-order process

$$g_0(s) = \frac{1}{(1s+1)(0.6s+1)}$$

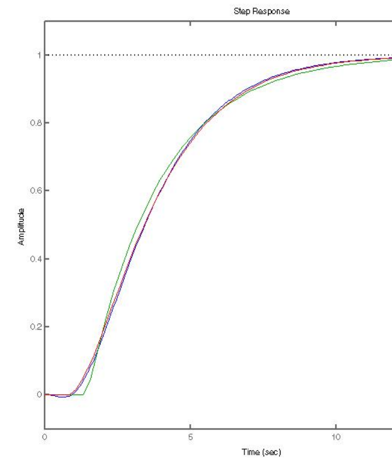
Half rule

is approximated as a first-order with delay process

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

with

$$k = 1; \quad \tau_1 = 1 + 0.6/2 = 1.3; \quad \theta = 0.6/2 = 0.3;$$



Example 2

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

half rule

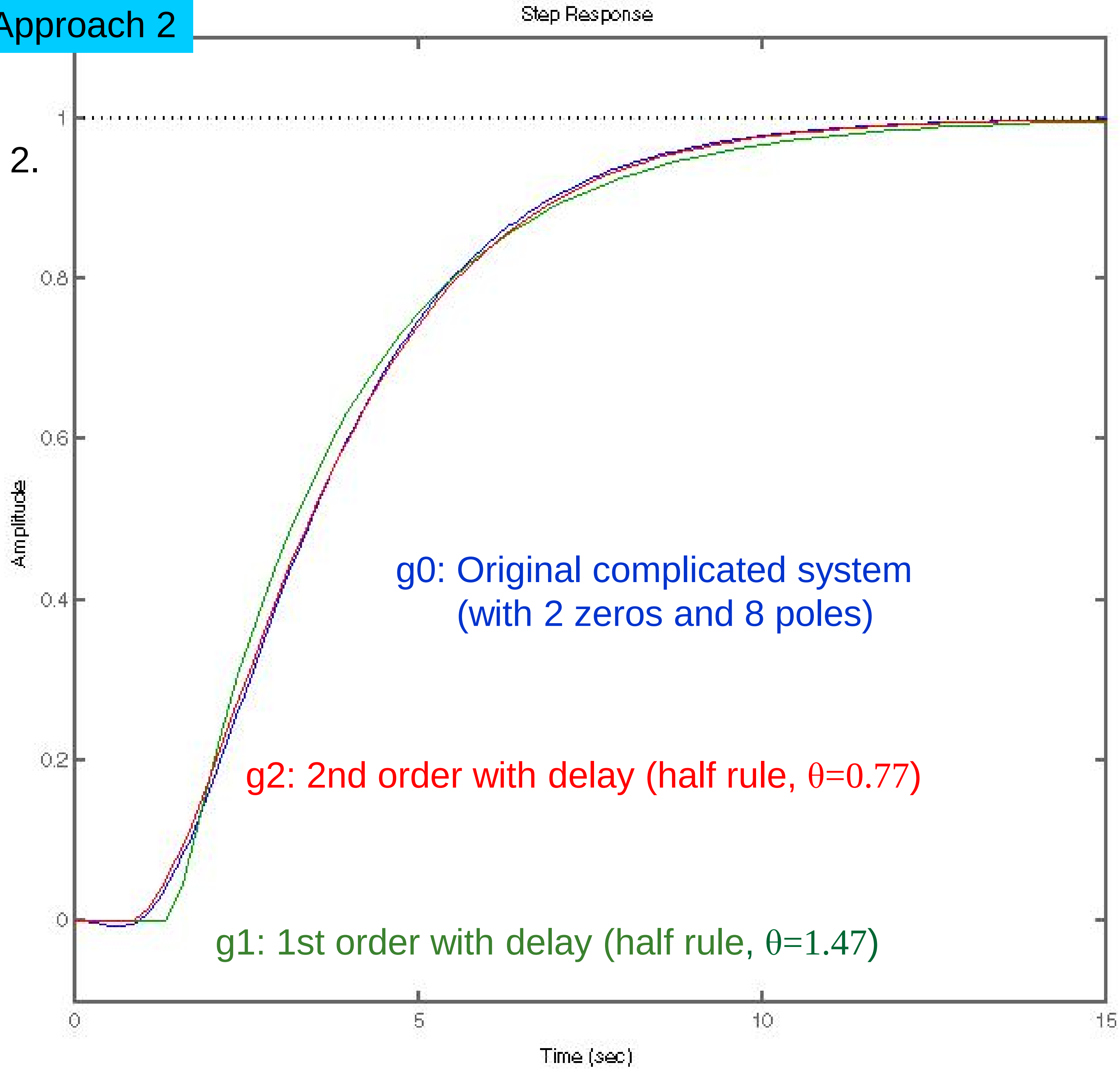
is approximated as a first-order delay process with

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

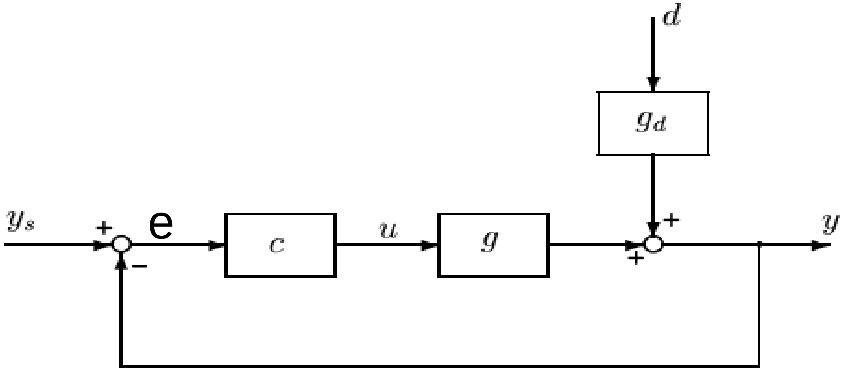
or as a second-order delay process with

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

Example 2.


Step Response
1
0.8
0.6
0.4
0.2
0
Amplitude
0
5
10
15
Time (sec)
g0: Original complicated system
(with 2 zeros and 8 poles)
g2: 2nd order with delay (half rule, θ=0.77)
g1: 1st order with delay (half rule, θ=1.47)

PID controller



- Time domain ("ideal" PID)

$$u(t) = u_0 + K_c' \left(e(t) + \frac{1}{\tau_I'} \int_0^t e(t^*)dt^* + \tau_D' \frac{de(t)}{dt} \right)$$

- Laplace domain ("ideal"/"parallel" form)

$$c(s) = K_c'(1 + \frac{1}{\tau_I' s} + \tau_D' s)$$

- For our purposes. Simpler with cascade form

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

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

- Usually τ_D=0. Then the two forms are identical.
- Only two parameters left (K_c and τ_I)
- How difficult can it be to tune???
  - Surprisingly difficult without systematic approach!

Let's start with the CONCLUSION

Tuning of PID controllers

- SIMC tuning rules ("Skogestad IMC")(*)
- Main message: Can usually do much better by taking a systematic approach
- Key: Look at <u>initial part</u> of step response

  Initial slope: $k' = k/\tau_1$
- One tuning rule!

For cascade-form PID controller:

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

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

$$\tau_D = \tau_2$$

- $\tau_c \geq -\theta$: desired closed-loop response time (tuning parameter)
- For robustness select: $\tau_c \geq \theta$

Reference: S. Skogestad, "Simple analytic rules for model reduction and PID controller design", *J.Proc.Control*, Vol. 13, 291-309, 2003
(Also reprinted in MIC)
(*) "Probably the best simple PID tuning rules in the world"

Derivation of SIMC-PID tuning rules

- PI-controller (based on first-order model)

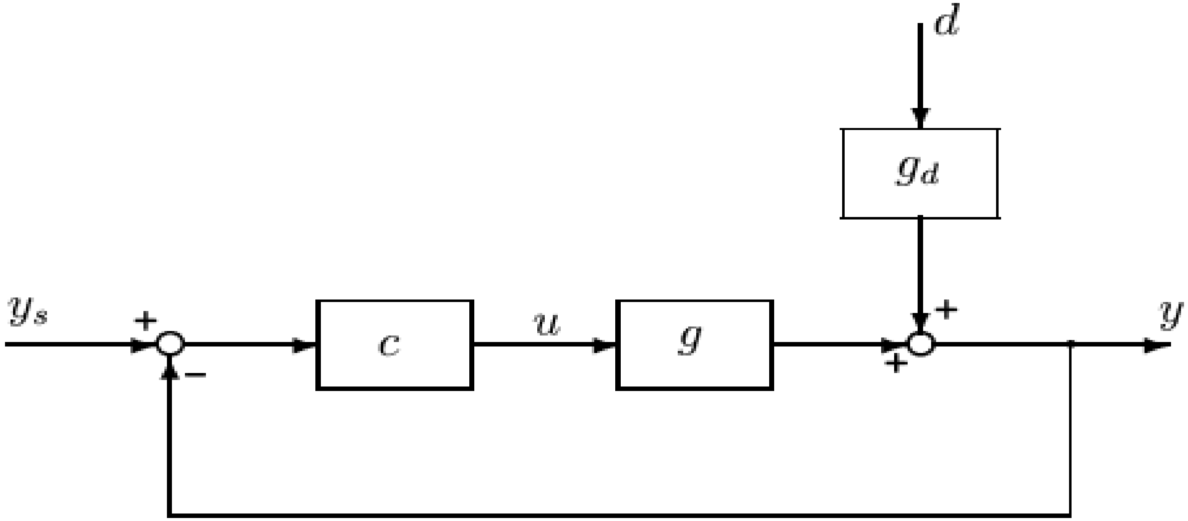
$$c(s) = K_c\left(1 + \frac{1}{\tau_I s}\right) = K_c \frac{\tau_I s + 1}{\tau_I s}$$

- For second-order model add D-action.

  For our purposes, simplest with the "series" (cascade) PID-form:

$$c(s) = K_c \frac{(\tau_I s + 1)(\tau_D s + 1)}{\tau_I s} \qquad (1)$$

Basis: Direct synthesis (IMC)



Closed-loop response to setpoint change

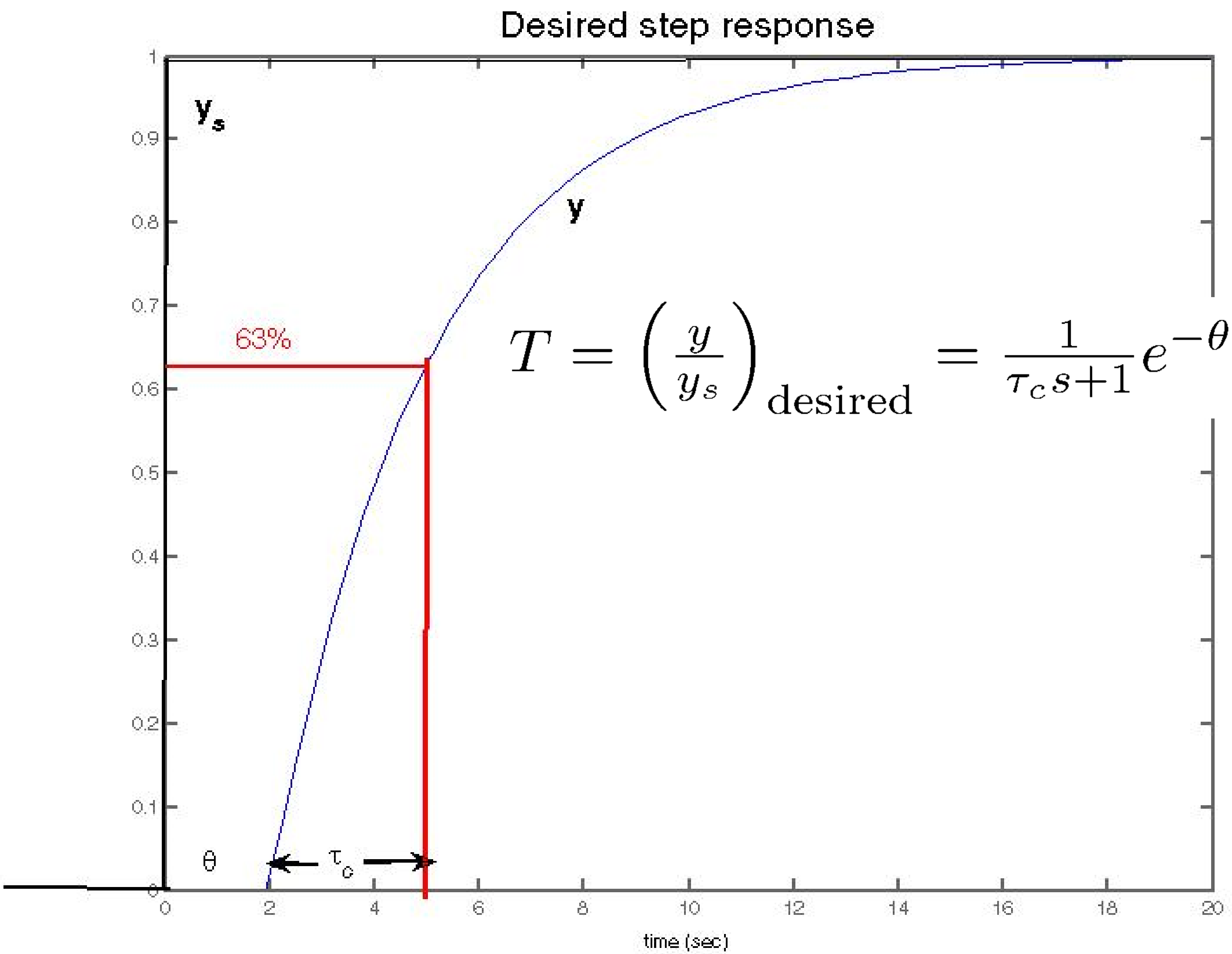
$$y = T\ y_s;\ \ T(s) = \frac{gc}{1+gc}$$

Idea: Specify desired response: $(y/y_s)_{\text{desired}} = T$

and from this get the controller. ……. Algebra:

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

SIMC-tunings



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

NOTE: Setting the steady-state gain = 1 in T will result in integral action in the controller!

IMC Tuning = Direct Synthesis

Algebra:

- 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

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

- τ_c is the sole tuning parameter

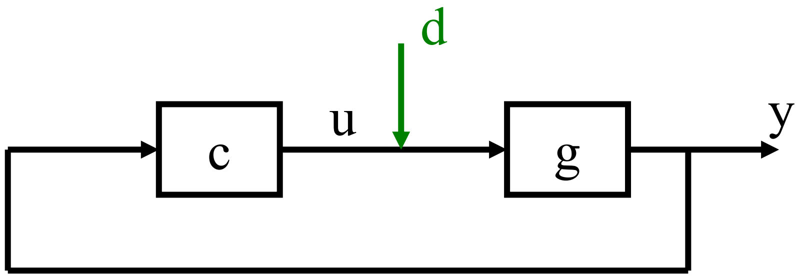
Surprisingly, this PID-controller is generally better, or at least more robust, than the Smith Predictor controller from which it was derived.

Reference: Chriss Grimholt and Sigurd Skogestad. "Should we forget the Smith Predictor?" (2018)

In 3rd IFAC conference on Advances in PID control, Ghent, Belgium, 9-11 May 2018. *In IFAC papers Online (2018)* .

Integral time

- Found: Integral time = dominant time constant ($\tau_I = \tau_1$)
- Works well for setpoint changes
- Needs to be modified (reduced) for integrating disturbances



Example. “Almost-integrating process” with disturbance at input:

$G(s) = e^{-s}/(30s+1)$

Original integral time $\tau_I = 30$ gives poor disturbance response

Try reducing it!

Integral time

- Want to reduce the integral time for "integrating" processes, but to avoid "slow oscillations" we must require:

$$\tau_I \geq 4(\tau_C + \theta)$$

- Derivation:

$$G(s) = k\frac{e^{-\theta s}}{\tau_1 s+1} \approx \frac{k'}{s} \text{ where } k' = \frac{k}{\tau_1};\ C(s) = K_c\left(1 + \frac{1}{\tau_I s}\right)$$

Closed-loop poles:

$$1 + GC = 0 \Rightarrow 1 + \frac{k'}{s}K_c\left(1 + \frac{1}{\tau_I s}\right) = 0 \Rightarrow \tau_I s^2 + k'K_c\tau_I s + k'K_c = 0$$

To avoid oscillations we must not have complex poles:

$$B^2 - 4AC \geq 0 \Rightarrow k'^2 K_c^2 \tau_I^2 - 4k'K_c\tau_I \geq 0 \Rightarrow k'K_c\tau_I \geq 4 \Rightarrow \tau_I \geq \frac{4}{k'K_c}$$

Inserted SIMC-rule for $K_c = \frac{1}{k'}\frac{1}{\tau_c+\theta}$ then gives

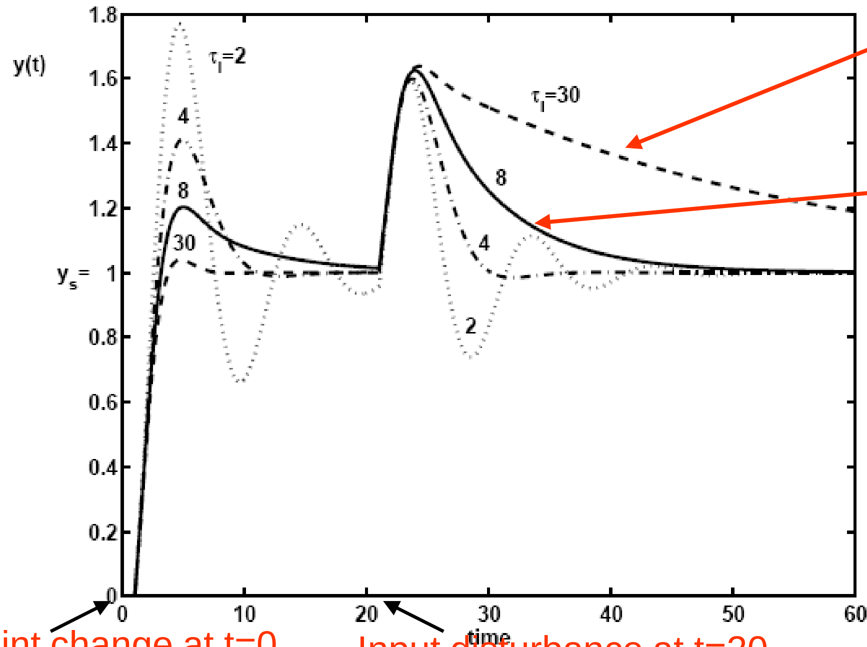
$$\tau_I \geq 4(\tau_c + \theta)$$

Avoid slow oscillations: $k'K_C\tau_I \geq 4$

Integral Time

$\tau_I = \tau_1$

Reduce τ_I to this value:
$\tau_I = 4\ (\tau_c + \theta) = 8\ \theta$



Setpoint change at t=0

Input disturbance at t=20

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

Conclusion: SIMC-PID Tuning Rules

For cascade form PID controller:

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

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

$$\tau_D = \tau_2 \quad (3)$$

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: τ_c

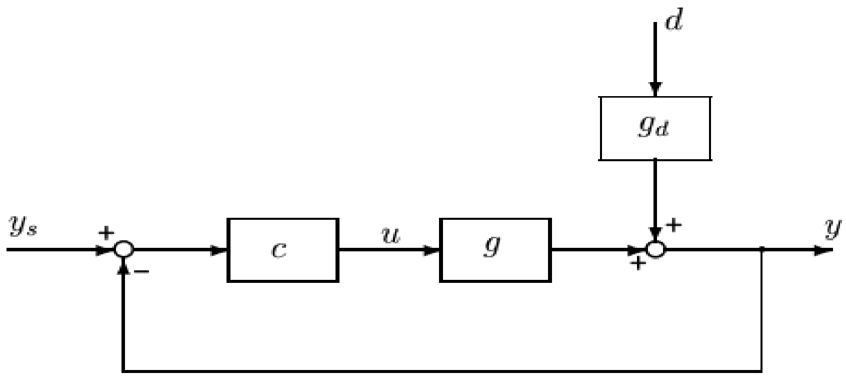
Example 2. SIMC PI and PID tunings

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

```
s=tf('s')
g=(-0.3*s+1)*(0.08*s+1)/((2*s+1)*(s+1)*(0.4*s+1)*(0.2*s+1)*(0.05*s+1)^3)
k=1;
tau1=2.5, tau2=0, theta=1.47, tauc=theta % 1st order
%tau1=2, tau2=1.2, theta=0.77, tauc=theta % 2nd order
```

Note: Tau2>theta , so 2nd order and PID is recommended

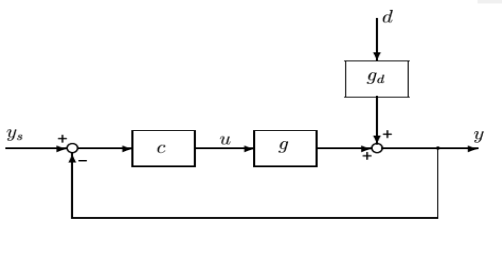
```
Kc=(1/k)*tau1/(tauc+theta)       % Kc.   PI: 0.85  PID: 1.30
taui=min(tau1,4*(tauc+theta))    % taui. PI: 2.50  PID: 2
taud=tau2;                       % taud. PI: 0     PID: 1.2
cpi=Kc*(1+1/(taui*s));
cd=(taud*s+1)/(0.1*taud*s+1);
cpid=cpi*cd;
L = cpid*g
S=inv(1+L)
%setpoint response
Ty=g*cpi*S, Ty=minreal(Ty); % without D-action on setpoint
Tuy=cpi*S, Tuy=minreal(Tuy); % without D-action on setpoint
%Input disturbance
gd=g;
Td=gd*S; Td=minreal(Td);
Tud=-gd*cpid*S; Tud=minreal(Tud);
Typi=Ty; Tdpi=Td; Tuypi=Tuy; Tudpi=Tud;
%Typid=Ty; Tdpid=Td; Tuypid=Tuy; Tudpid=Tud;

figure(1),step(Typi,'blue',Typid,'blue--',Tuypi,'red',Tuypid,'red--',15)
figure(2),step(Tdpi,'blue',Tdpid,'blue--',Tudpi,'red',Tudpid,'red--',15)
```

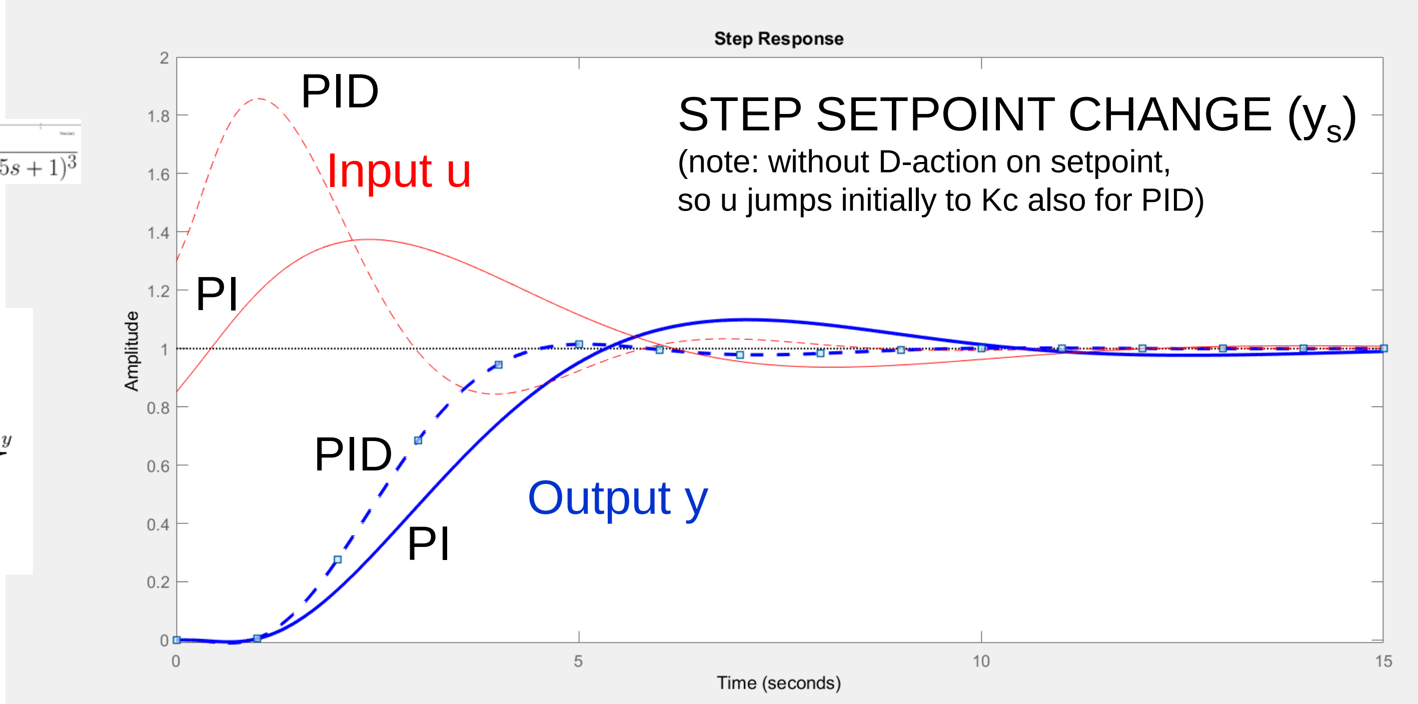


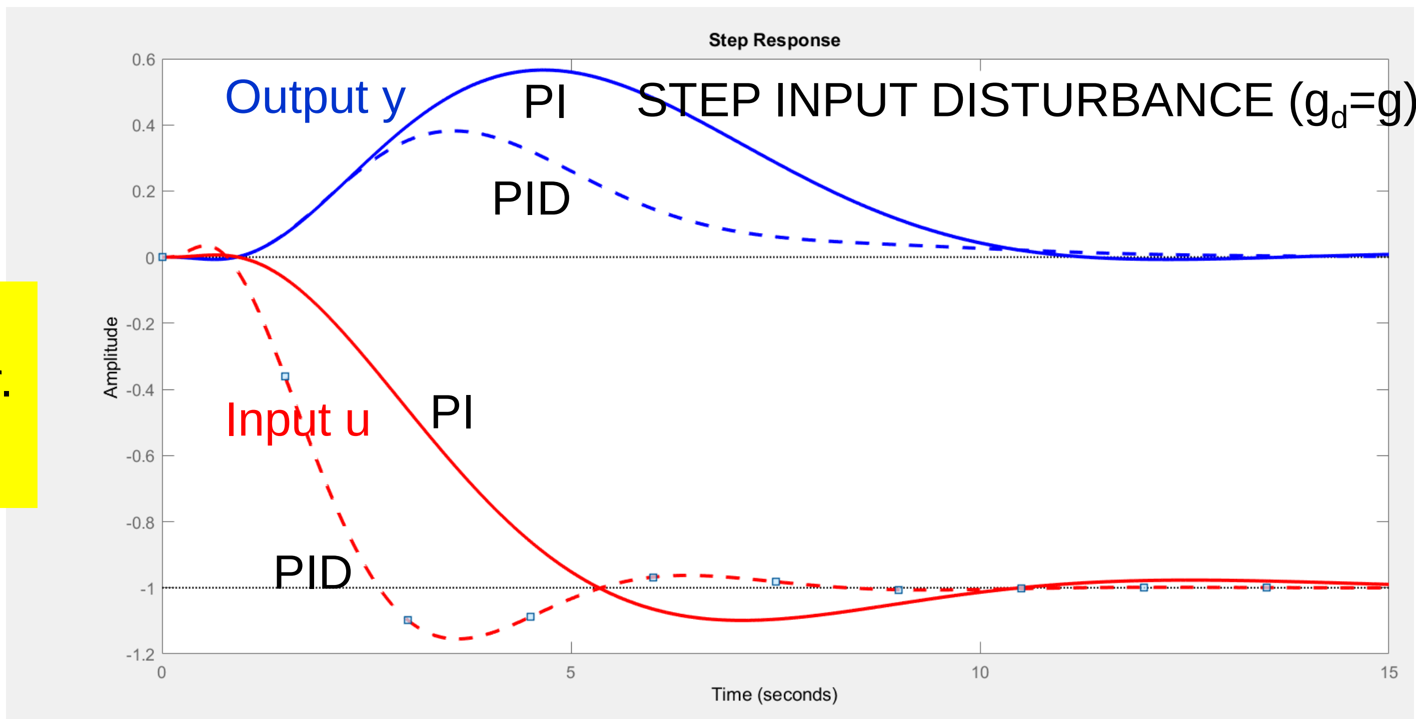
Example 2.

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

Comparison of PI and PID --







Conclusion:
PID is quite a lot better.
(expected since tau2=1.2
> theta=0.77)

SIMC-tunings

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: τ_c

DERIVATIVE ACTION ?

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

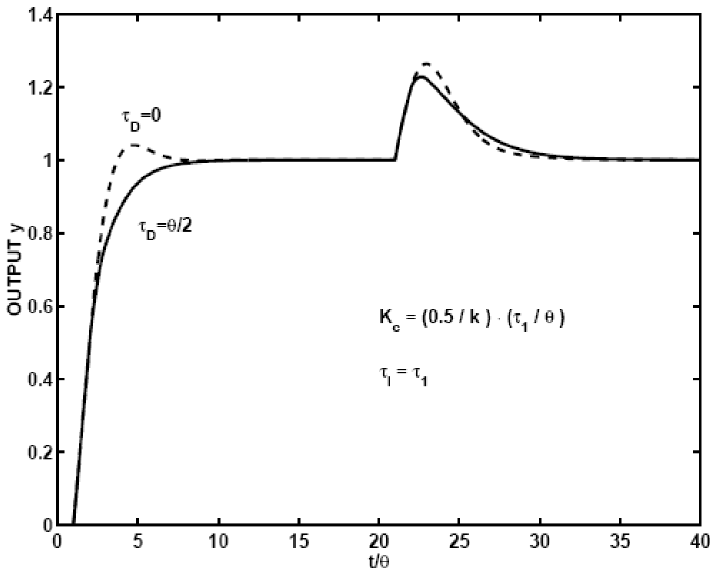


Figure 5: Setpoint change at $t = 0$. Load disturbance of magnitude 0.5 occurs at $t = 20$.

- Observe: Derivative action (solid line) has only a minor effect.

Conclusion D-action:

1. Use PID for dominant 2nd order processes with $\tau_2 > \theta$ (otherwise, add $\tau_2/2$ to effective delay θ and use PI)
   - Common rule: Select τ_D equal to τ_2 = time constant of temperature sensor

2. Use derivative action for unstable processes, for example, a double integrating process (not so common in process control).

3. Derivative action can help a little to speed up response for a process with time delay, but probably not worth it (see above with $\tau_D = \theta/2$).

6.3 Ideal PID controller

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

$$\text{Ideal PID}: \quad c'(s) = K_c'\left(1 + \frac{1}{\tau_I' s} + \tau_D' s\right) = \frac{K_c'}{\tau_I' s}\left(\tau_I'\tau_D' s^2 + \tau_I' s + 1\right) \tag{35}$$

we use the following translation formulas

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

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

Too complicated

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

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

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 $K_c = 0.5$, $\tau_I = 1$ and $\tau_D = 1$. The corresponding settings for the ideal PID controller in (35) are $K_c' = 1$, $\tau_I' = 2$ and $\tau_D' = 0.5$. The robustness margins with these settings are given by the first column in Table 2.*

Selection of tuning parameter τ_c

Two main cases

1. TIGHT CONTROL (τ_c small): Want "fastest possible control" subject to having good robustness
   - Want tight control of active constraints ("squeeze and shift")
2. SMOOTH CONTROL (τ_c large): Want "slowest possible control" subject to acceptable disturbance rejection
   - Want smooth control if fast setpoint tracking is not required, for example, levels and unconstrained ("self-optimizing") variables

TUNING FOR FAST RESPONSE WITH GOOD ROBUSTNESS

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

Gives:

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

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

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

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 $\tau_D = \tau_2$.

Typical closed-loop SIMC responses with the choice $\tau_c=\theta$

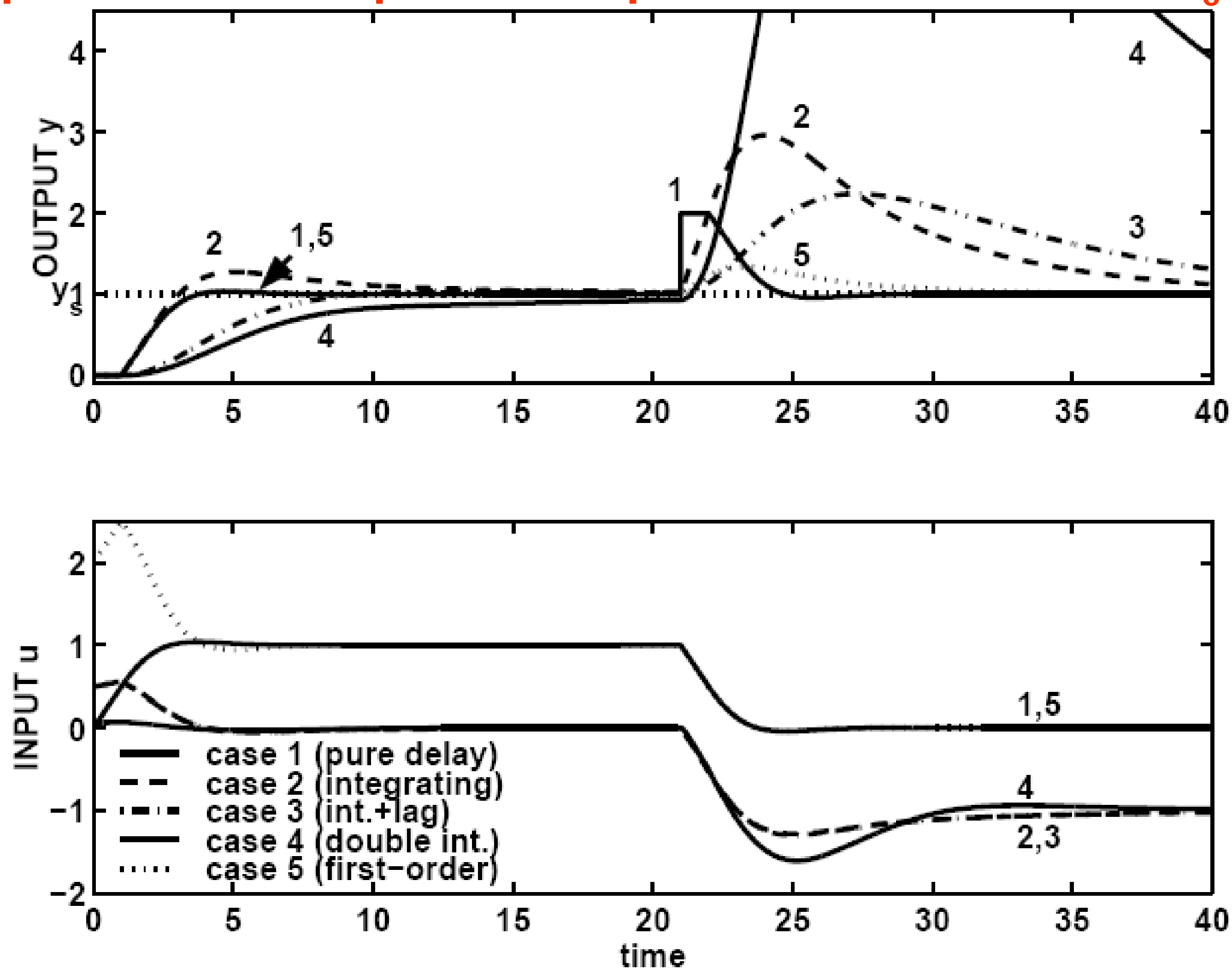
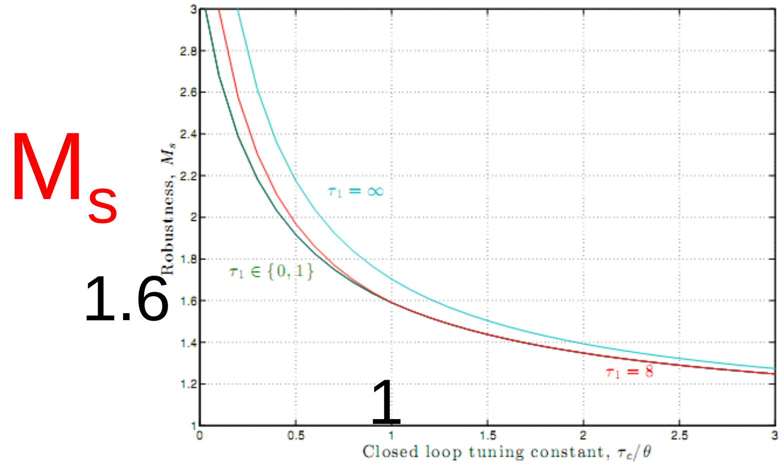
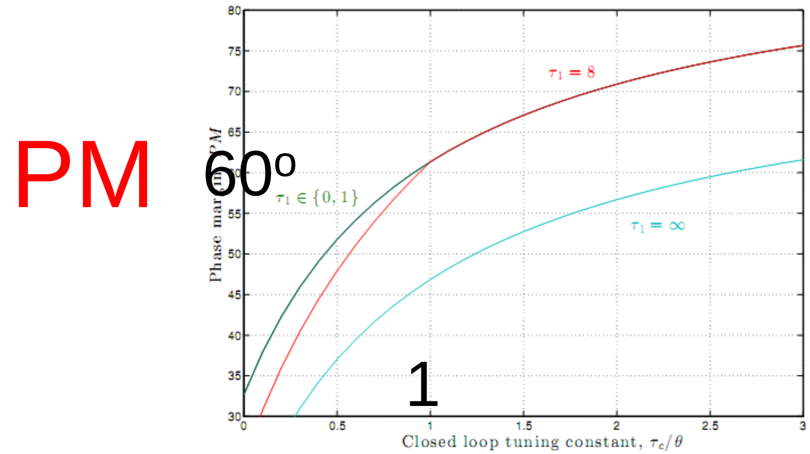
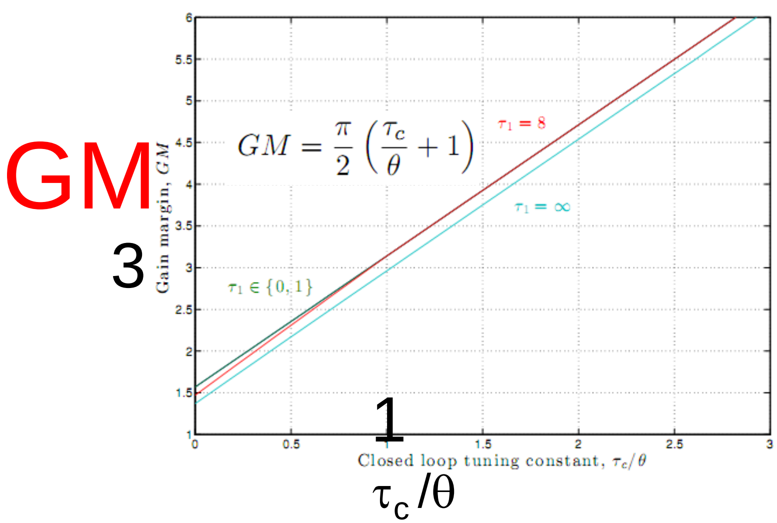


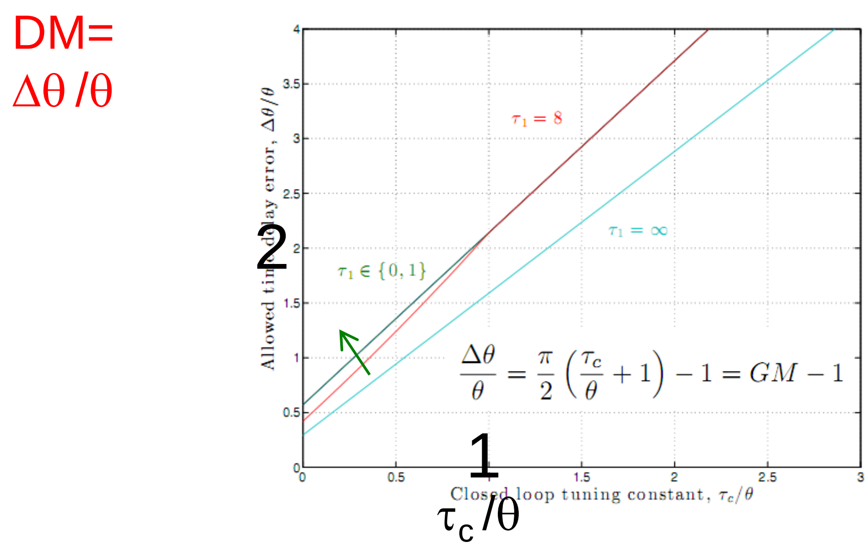
Figure 4: Responses using SIMC settings for the five time delay processes in Table 3 ($\tau_c = \theta$).
Unit setpoint change at $t = 0$; Unit load disturbance at $t = 20$.
Simulations are without derivative action on the setpoint.
Parameter values: $\theta = 1, k = 1, k' = 1, k'' = 1$.

SIMC: Tuning parameter (τ_c) correlates nicely with robustness measures









Tuning for smooth control

- Tuning parameter: τ_c = desired closed-loop response time
- Selecting $\tau_c=\theta$ if we need "tight control" of y.
- Other cases: "Smooth control" of y is sufficient, so select $\tau_c > \theta$ for
  - slower control
  - smoother input usage
    - less disturbing effect on rest of the plant
  - less sensitivity to measurement noise
  - better robustness
- Question: Given that we require some disturbance rejection.
  - What is the largest possible value for τ_c ?
  - Or equivalently: What is the smallest possible value for K_c?
  - ANSWER:

$K_{c,min} = u_d/y_{max}$.

u_d = input change to reject disturbance (steady-state)

- *May obtain u_d from historical data!*

y_{max} = maximum *desired* output deviation

From K_c we can get τ_c and then corresponding τ_I using SIMC tuning rule

«Proof»: Imagine using P-control only. Then we get at steady-state $u = K_c y_{ss}$ where y_{ss} is the steady-state offset. With I-action we have no offset but the peak value of y will be close to y_{ss}
More detailed proof: S. Skogestad, ``Tuning for smooth PID control with acceptable disturbance rejection'', *Ind.Eng.Chem.Res*, **45** (23), 7817-7822 (2006).

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

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

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

u0= input change required to reject disturbance
ymax = largest allowed output change

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

3. Derivative time: Only for dominant second-order processes

Level control

- Level control often causes problems
- 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.

Level control: Can have both fast and slow oscillations

- Slow oscillations (K_c too low): $P > 3\tau_I$
- Fast oscillations (K_c too high): $P < 3\tau_I$

Here: Consider the very common slow oscillations

Avoid slow oscillations: $k' K_C \tau_I \geq 4$

How avoid slowly 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]

$0.1 \approx 1/\pi^2$

Avoid slow oscillations: $k' K_C \tau_I \geq 4$

Case study oscillating level

- We were called upon to solve a problem with oscillations in a distillation column
- Closer analysis: Problem was oscillating reboiler level in upstream column
- Use of Sigurd's rule solved the problem

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:

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