The improved SIMC method for PI controller tuning

Chriss Grimholt
Sigurd Skogestad
NTNU, Trondheim, Norway

Reference: C. Grimholt and S. Skogestad, “The improved SIMC method for PI controller tuning”,
IFAC-conference PID’12, Brescia, Italy, March 2012

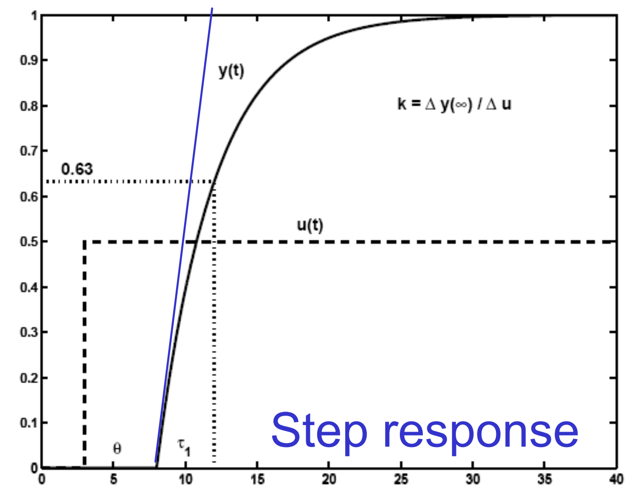
SIMC PI tuning rule*

- Look at initial part of step response, Initial slope: $k' = k/\tau_1$
- One tuning rule! Easily memorized

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

$\tau_c \geq -\theta$: Desired closed-loop response time (tuning parameter)

•For robustness select: $\tau_c \geq \theta$



Questions:
1. How good is really this rule?
2. Can it be improved?

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 rule in the world"

1. How good is really the SIMC rule?

Need to compare with:

- Optimal PI-controller
  for class of first-order with delay processes

Optimal controller

- Tradeoff between
  - Output performance
  - Robustness
  - Input usage
  - Noise sensitivity

High controller gain (“tight control”)

Low controller gain (“smooth control”)

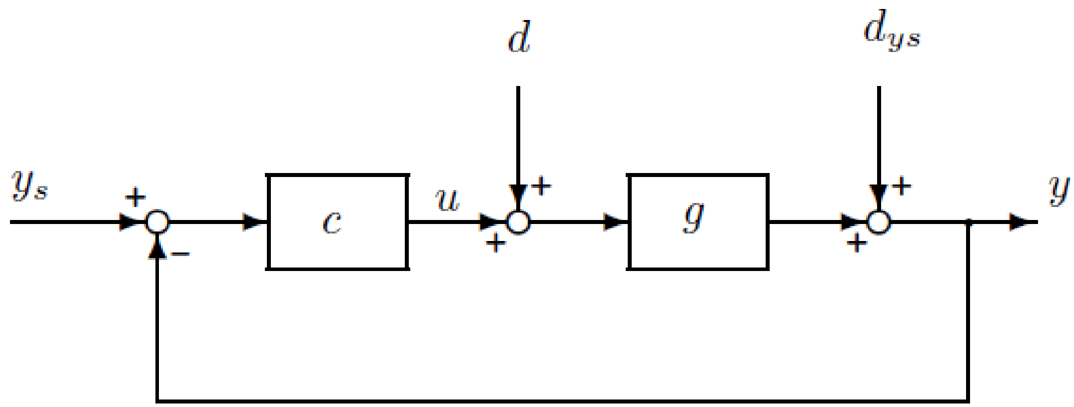
- Quantification
  - Output performance:
    - Frequency domain: weighted sensitivity $||W_pS||$
    - Time domain: IAE or ISE for setpoint/disturbance
  - Robustness: M_s, GM, PM, Delay margin
  - Input usage: $||KSG_d||$, ISE or TV for step response
  - Noise sensitivity: $||KS||$, etc.

Our choice:

J = avg. IAE for setpoint/disturbance

M_s = peak sensitivity

Output performance (J)



$$\boxed{J(c) = 0.5\frac{\mathrm{IAE}_{ys}(c)}{\mathrm{IAE}^{o}_{ys}} + 0.5\frac{\mathrm{IAE}_{d}(c)}{\mathrm{IAE}^{o}_{d}}}$$

IAE^{o}_{ys}: PI-optimal for setpoint ($M_s = 1.59$)

IAE^{o}_{d}: PI-optimal for disturbance ($M_s = 1.59$)

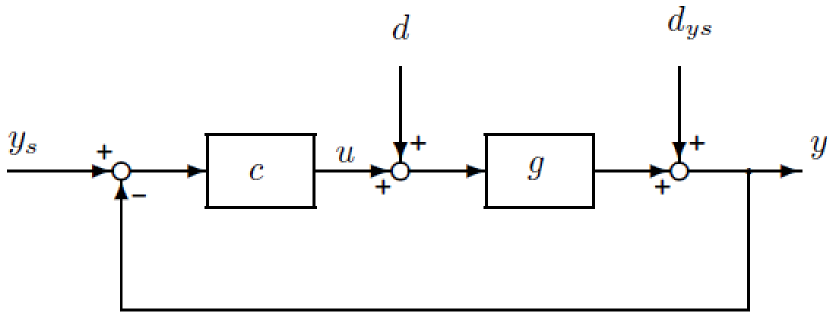
IAE = Integrated absolute error = $\int|y-y_s|dt$, for step change in y_s or d

Cost J is independent of:

1. process gain k
2. disturbance magnitude
3. unit for time

NTNU
Norwegian University of Science and Technology

Optimal PI-controller: Minimize J for given M_s



$$J(c) = 0.5\frac{\mathrm{IAE}_{ys}(c)}{\mathrm{IAE}^o_{ys}} + 0.5\frac{\mathrm{IAE}_d(c)}{\mathrm{IAE}^o_d}$$

$\min_c J(c)|_{M_s=m}$

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

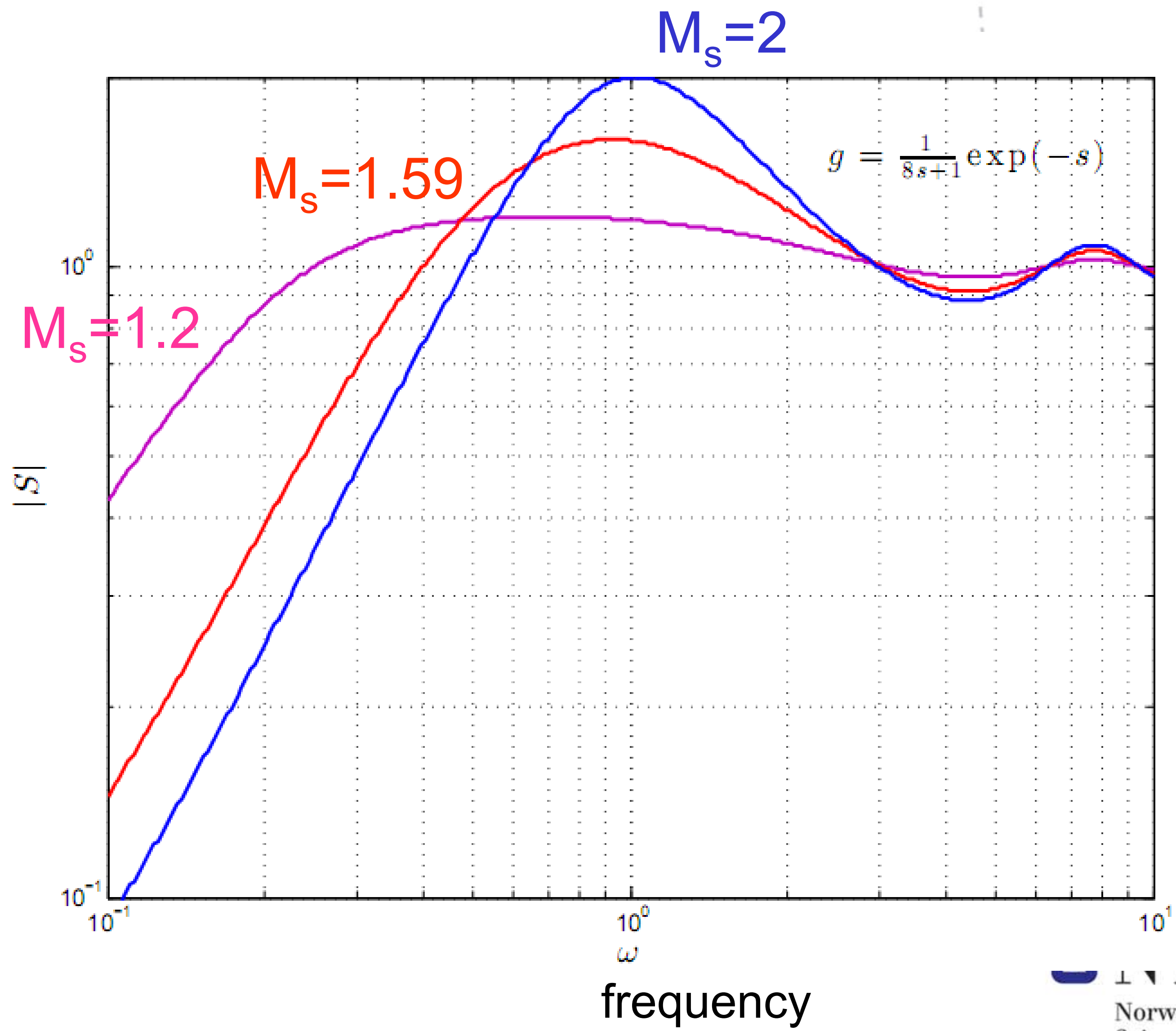
First-order with delay processes: $g(s) = \frac{k}{\tau_1 s+1}e^{-\theta s}$

$\theta = 1, \tau_1/\theta = [0, \infty]$

$m = [..., 1.2, 1.59, 1.7, 2....]$

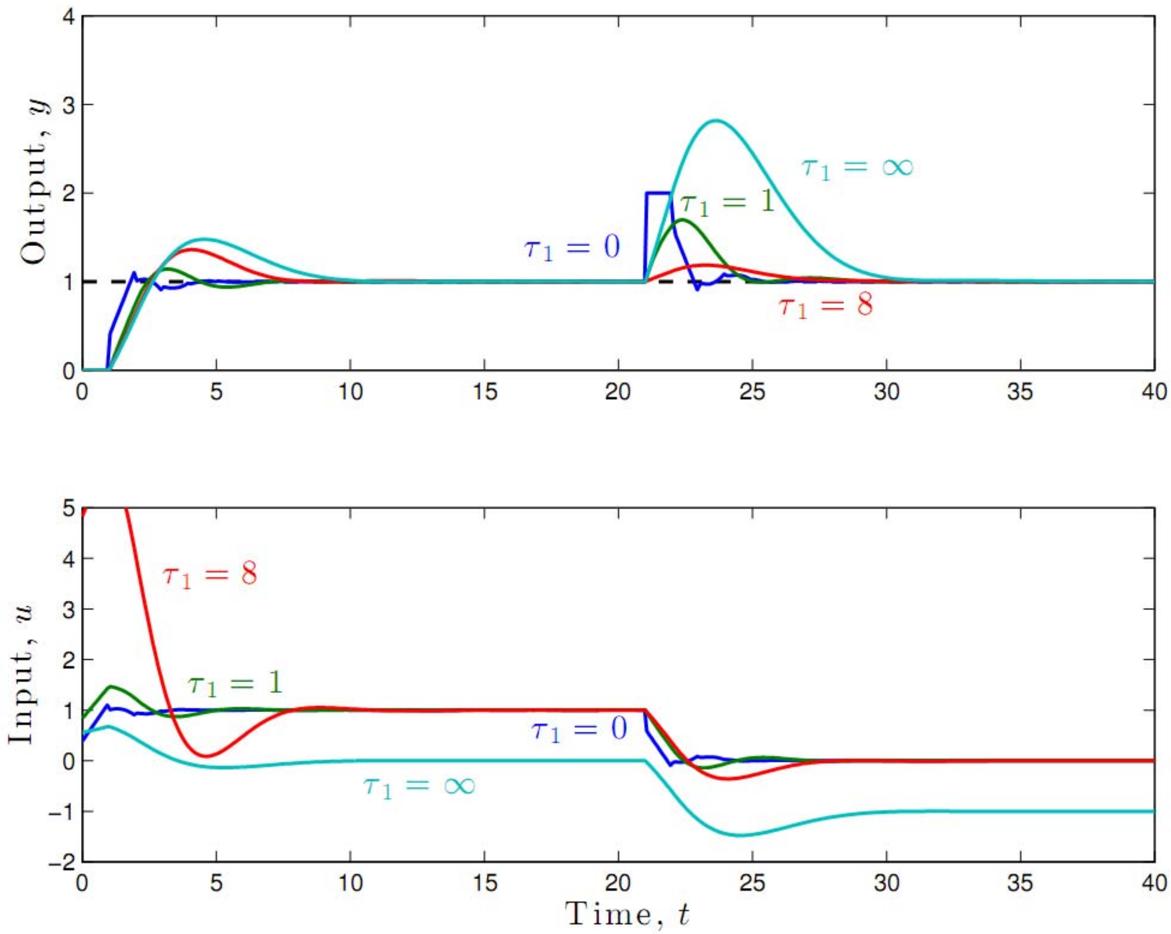
NTNU
Norwegian University of Science and Technology

|S|



NTNU
Norwegian University of Science and Technology

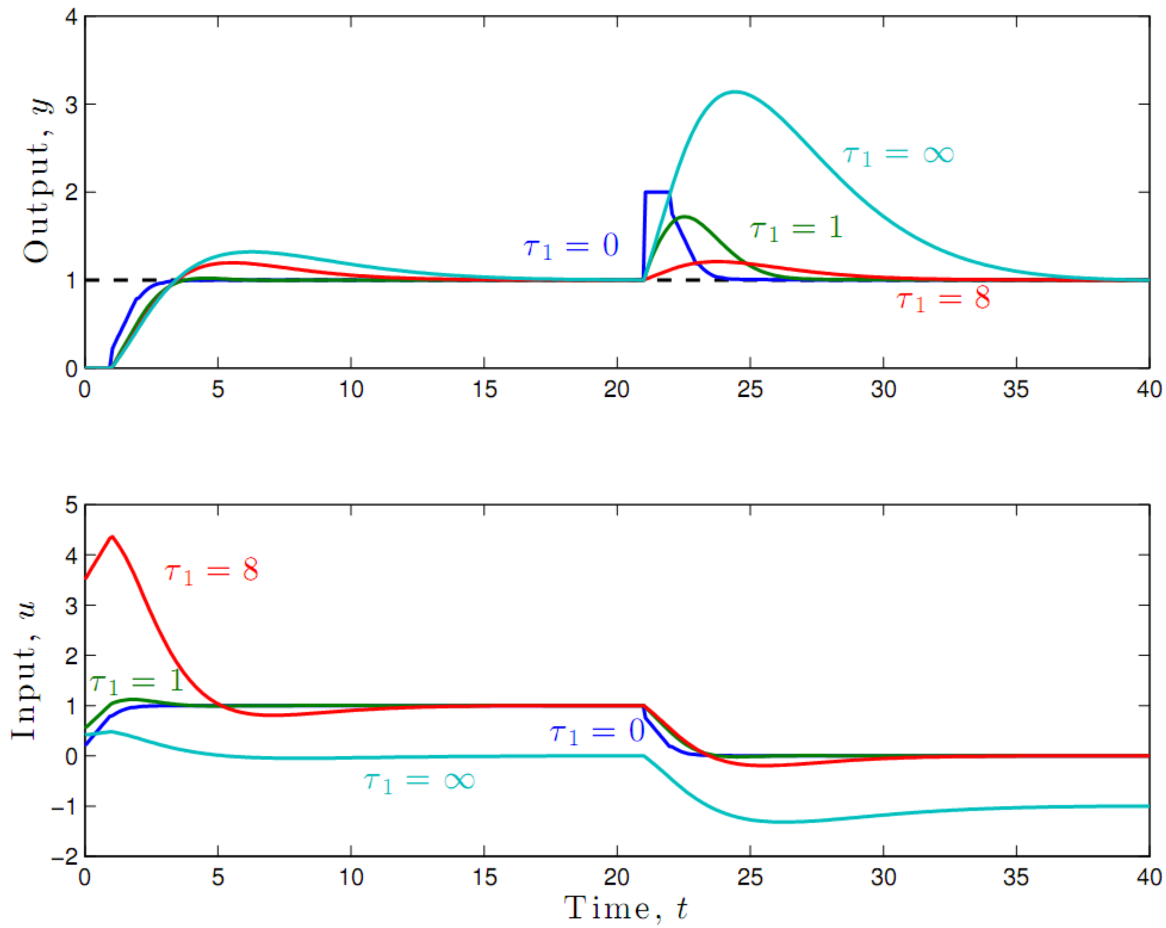
M_s=2



Setpoint change at t=0, Input disturbance at t=20, $g(s)=k\, e^{-\theta s}/(\tau_1 s+1)$, Time delay θ=1

NTNU
Norwegian University of Science and Technology

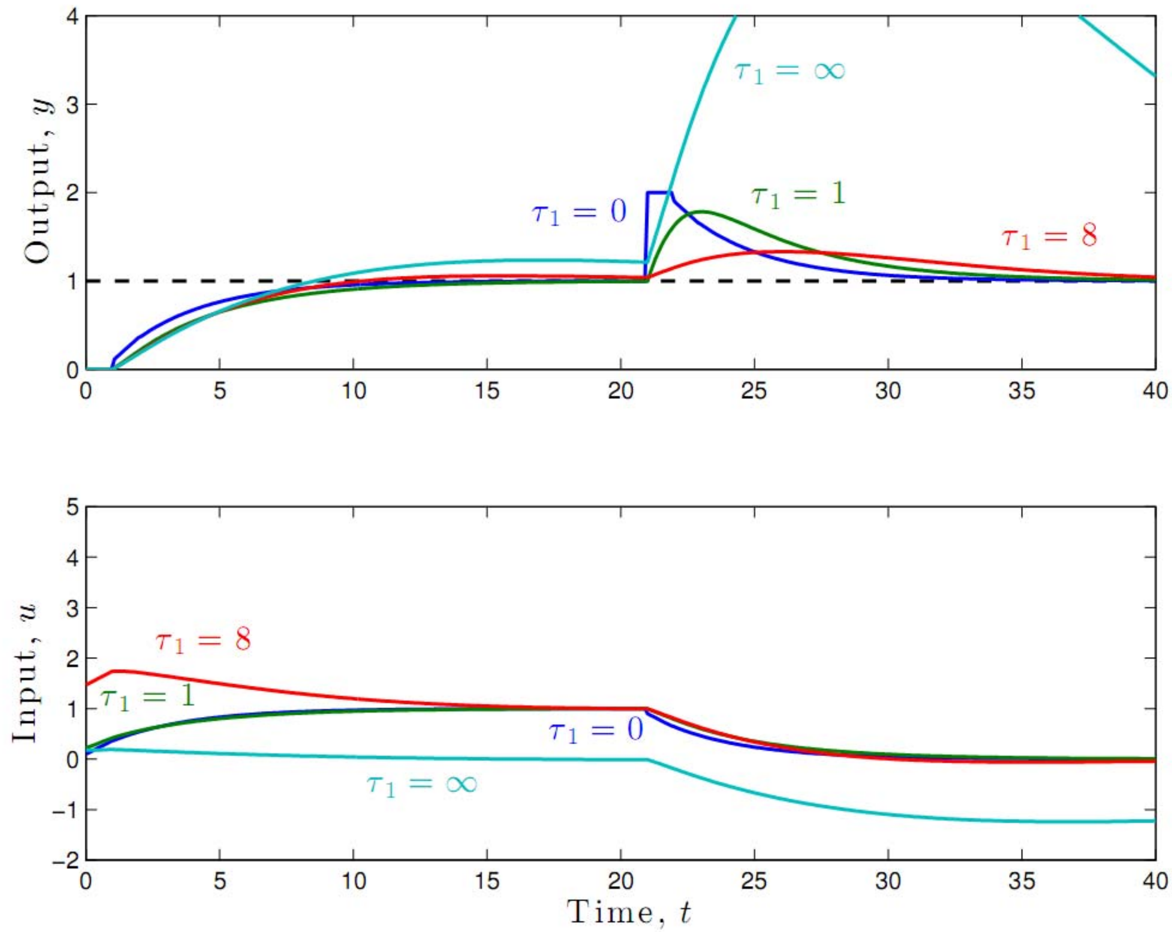
M_s=1.59



Setpoint change at t=0, Input disturbance at t=20,
g(s)=k $e^{-\theta s}/(\tau_1 s+1)$, Time delay θ=1

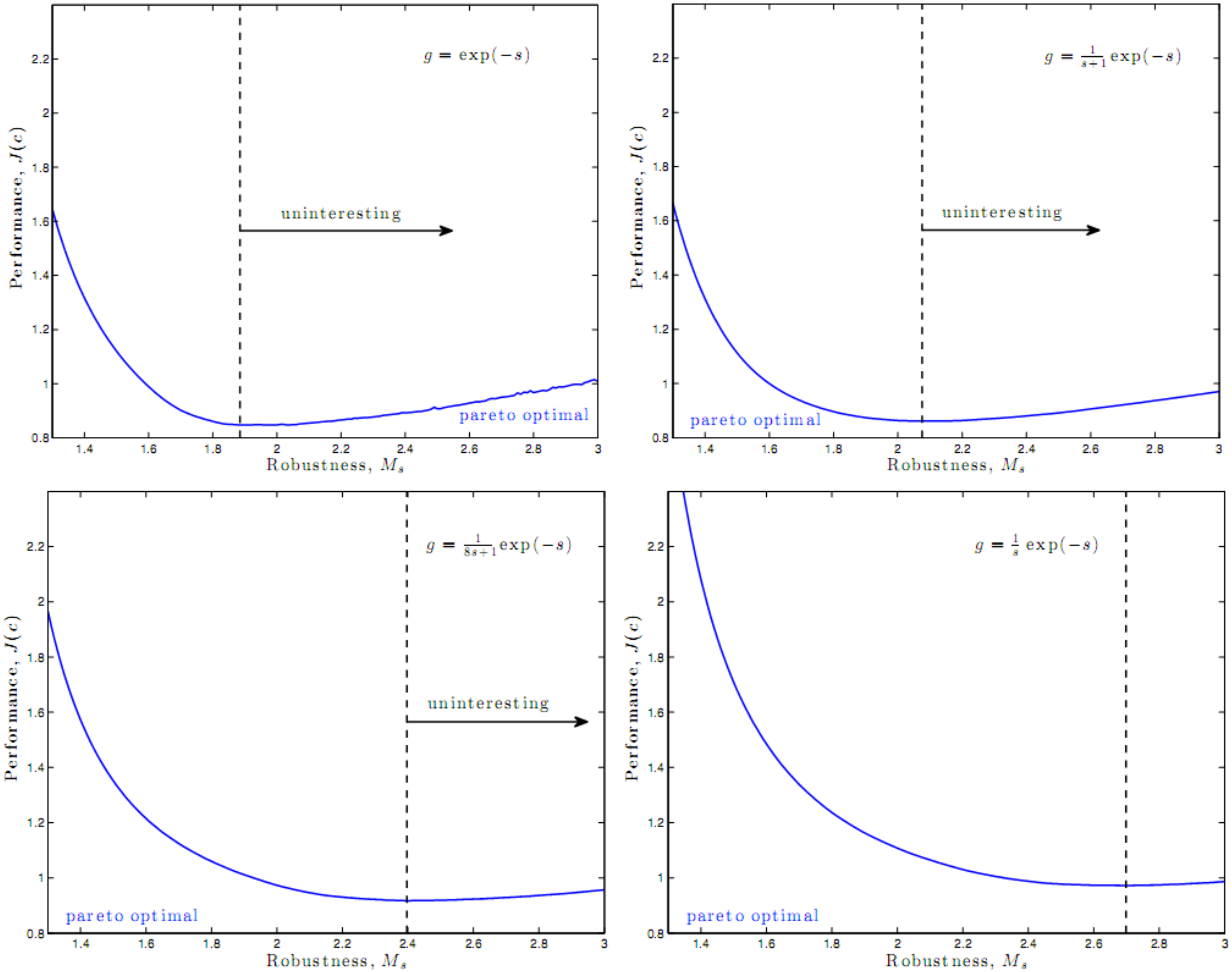
NTNU
Norwegian University of Science and Technology

M_s=1.2



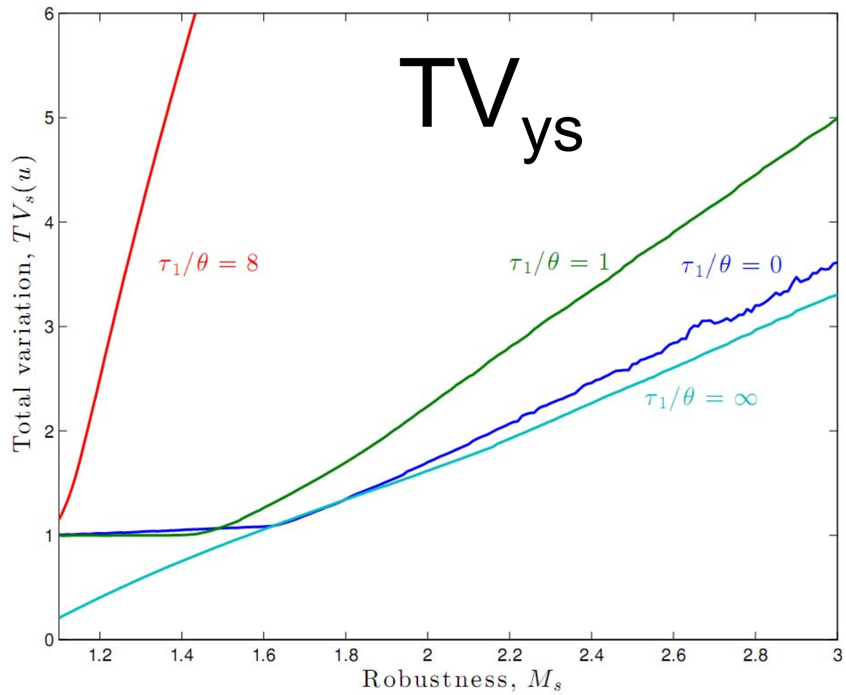
Setpoint change at t=0, Input disturbance at t=20,
g(s)=k $e^{-\theta s}/(\tau_1 s+1)$, Time delay θ=1

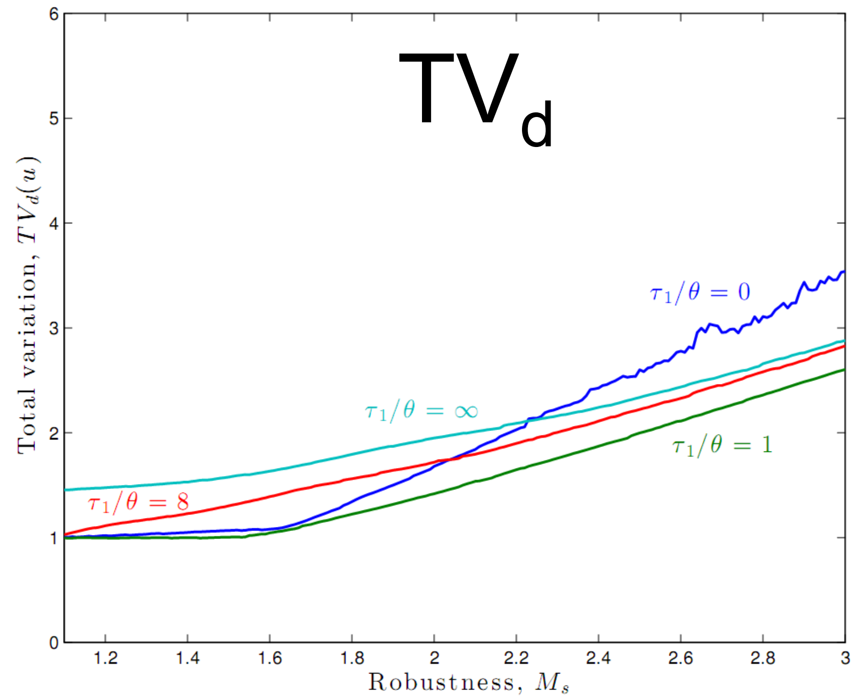
Optimal performance (J) vs. M_s



$$J(c) = 0.5\frac{\mathrm{IAE}_{ys}(c)}{\mathrm{IAE}^o_{ys}} + 0.5\frac{\mathrm{IAE}_d(c)}{\mathrm{IAE}^o_d}$$

Input usage (TV) increases with M_s

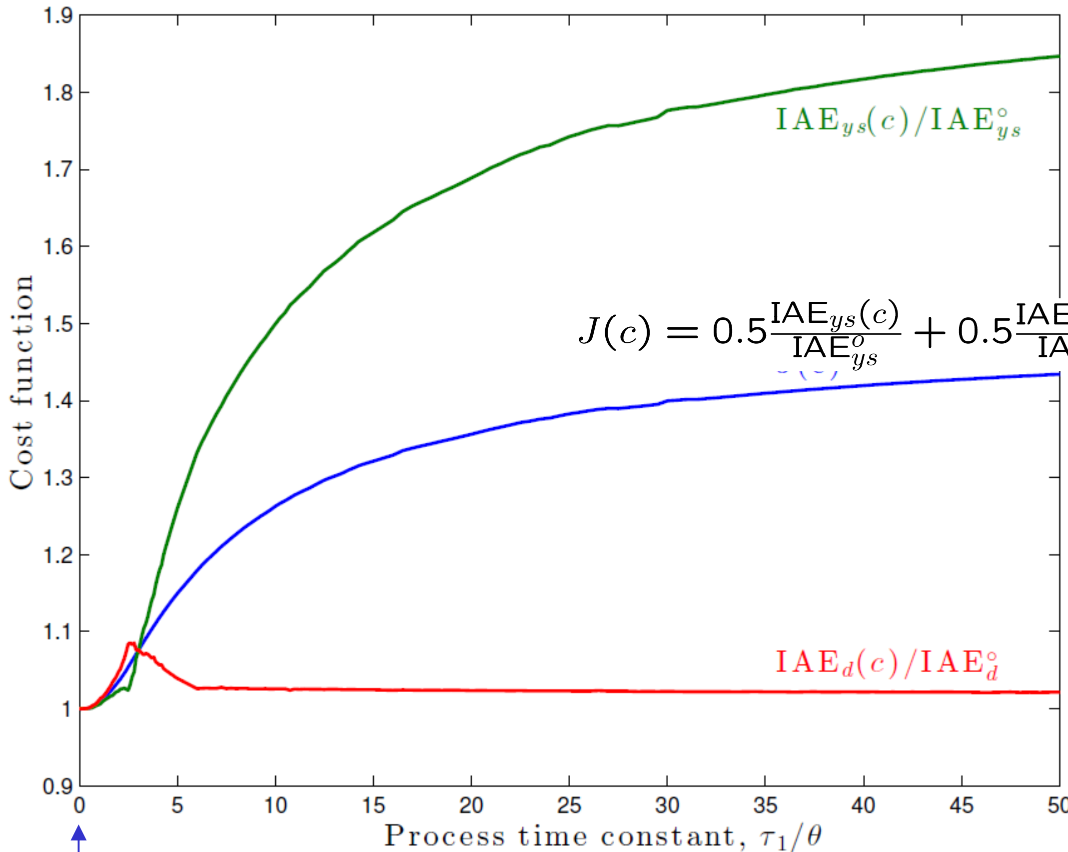




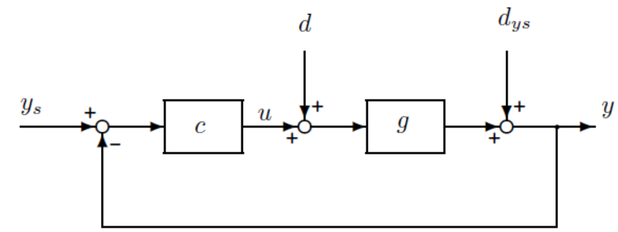
$$\mathrm{TV}(u) = \int_0^\infty \left|\frac{du}{dt}\right| dt = \Sigma_{i=1}^{\infty} |u_i - u_{i-1}|$$

NTNU
Norwegian University of Science and Technology

Setpoint / disturbance tradeoff



$$J(c) = 0.5\frac{\mathrm{IAE}_{ys}(c)}{\mathrm{IAE}^o_{ys}} + 0.5\frac{\mathrm{IAE}_{d}(c)}{\mathrm{IAE}^o_{d}}$$



Optimal controller:
Emphasis on disturbance d

Pure time delay process: J=1, No tradeoff
(since setpoint and disturbance the same)

NTNU
Norwegian University of Science and Technology

Setpoint / disturbance tradeoff

Table 1. Optimal PI-controllers ($M_s = 1.59$) and corresponding IAE-values for four processes.

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

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

Optimal setpoint:
No integral action

NTNU
Norwegian University of
Science and Technology

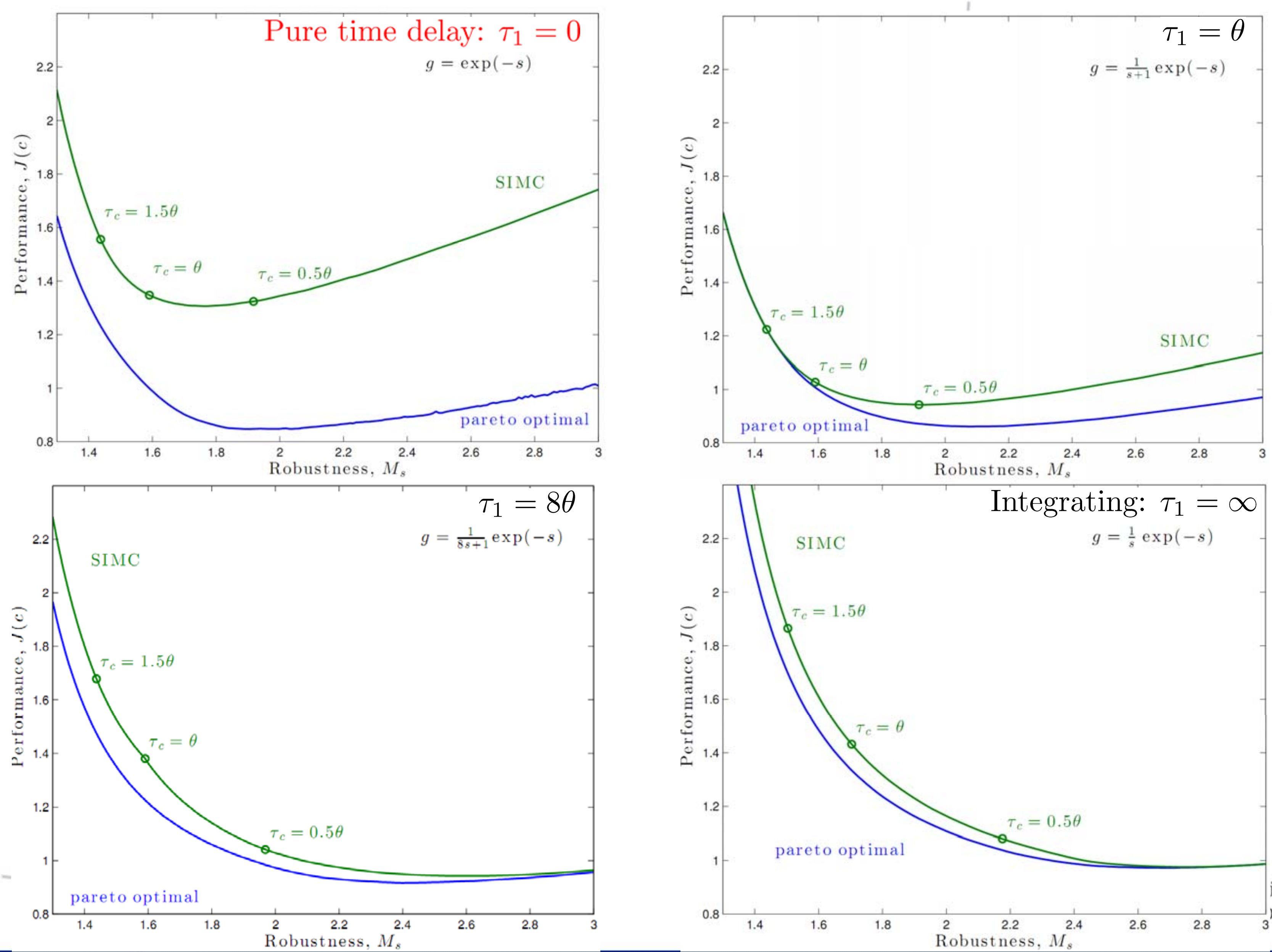
Comparison with SIMC

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

Tuning parameter: τ_c

Tight control with good robustness: Select $\tau_c = \theta$ (effective delay)
- Gives M_s between 1.59 and 1.7

Comparison of J vs. M_s for optimal and SIMC for 4 processes



Conclusion (so far):
How good is really the SIMC rule?

- Varying τ_C gives (almost) Pareto-optimal tradeoff between performance (J) and robustness (M_s)
- $\tau_C = \theta$ is a good ”default” choice
- Not possible to do much better with any other PI-controller!

- Exception: Time delay process

NTNU
Norwegian University of Science and Technology

2. Can the SIMC-rule be improved?

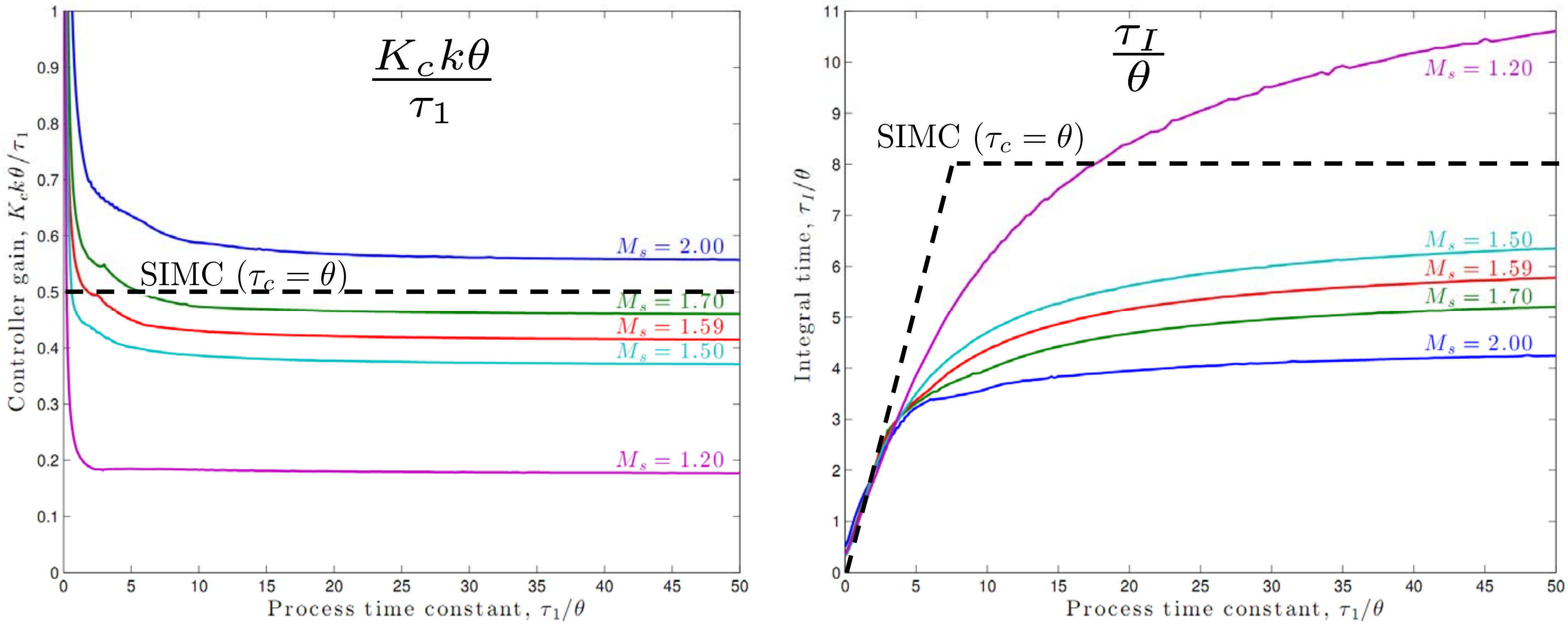
Yes, possibly for time delay process

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

Tuning parameter: τ_c

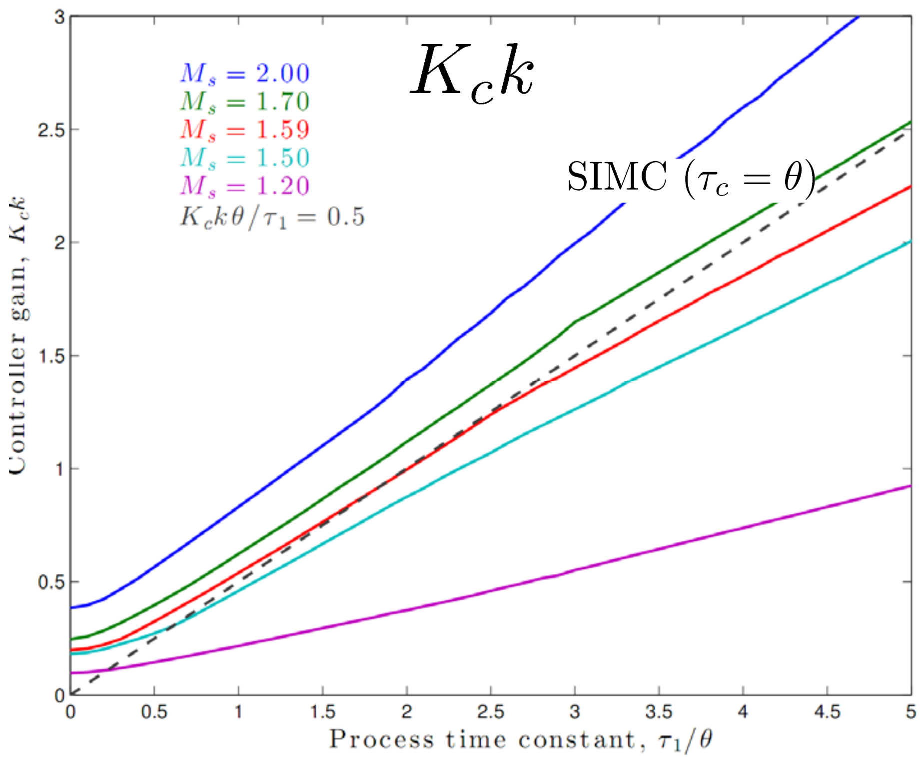
Time delay process, $g = k' e^{-\theta s}$:
SIMC-rule gives integrating controller ($\tau_I = \tau_1 = 0$)

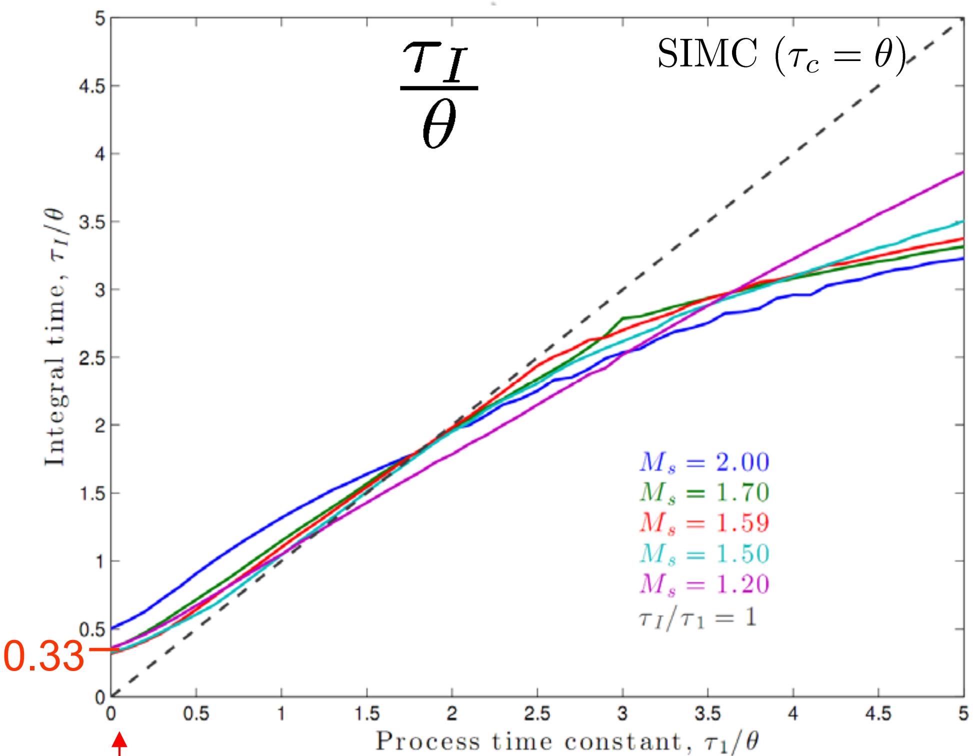
Optimal PI-settings



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

Optimal PI-settings (small τ_1)





Time-delay process
SIMC: $\tau_I = \tau_1 = 0$

NTNU
Norwegian University of Science and Technology

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

Improved SIMC-rule: Replace τ_1 by $\tau_1+\theta/3$

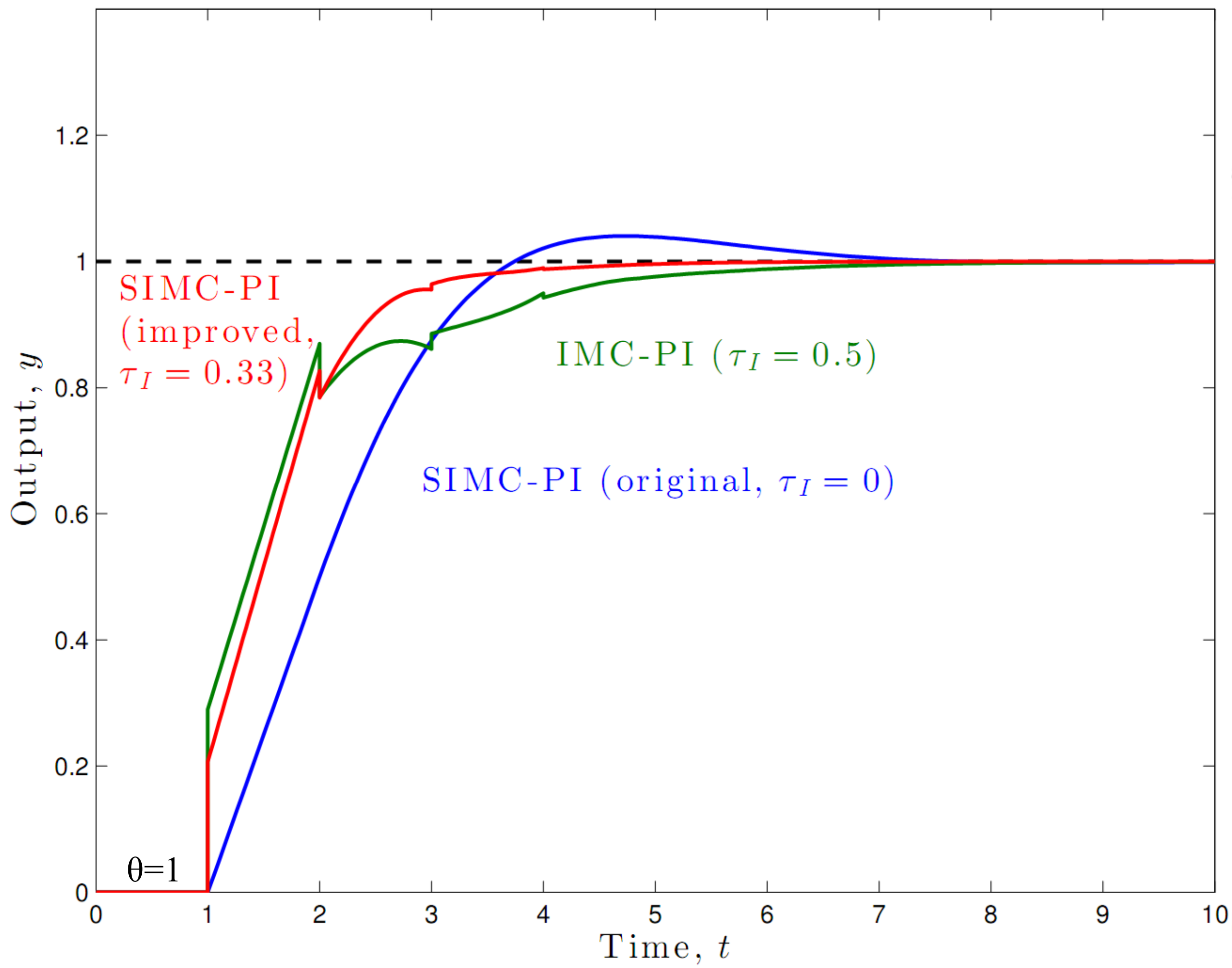
$$
\boxed{
\begin{aligned}
&K_c = \frac{1}{k} \cdot \frac{\tau_1 + \frac{\theta}{3}}{(\theta + \tau_c)} \\
&\tau_I = \min\left(\tau_1 + \frac{\theta}{3}, 4(\tau_c + \theta)\right)
\end{aligned}
}
$$

Tuning parameter: τ_c

Time delay process ($\tau_1 = 0$): $\tau_I = \frac{\theta}{3}$

NTNU
Norwegian University of Science and Technology

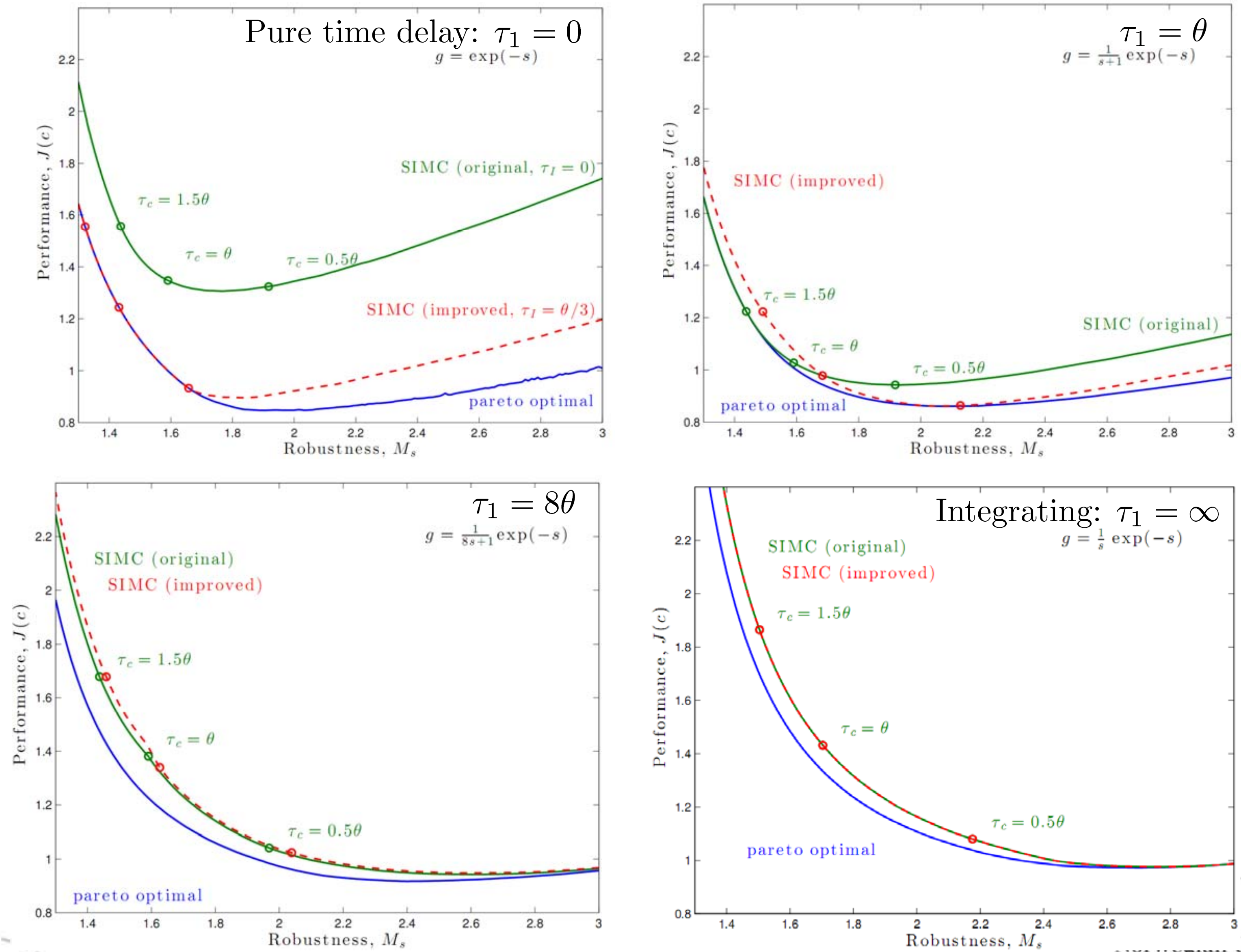
Step response for time delay process



NTNU
Norwegian University of Science and Technology

Time delay process: Setpoint and disturbance response same

Comparison of J vs. Ms for optimal and SIMC for 4 processes



University of
Science and Technology

Conclusion

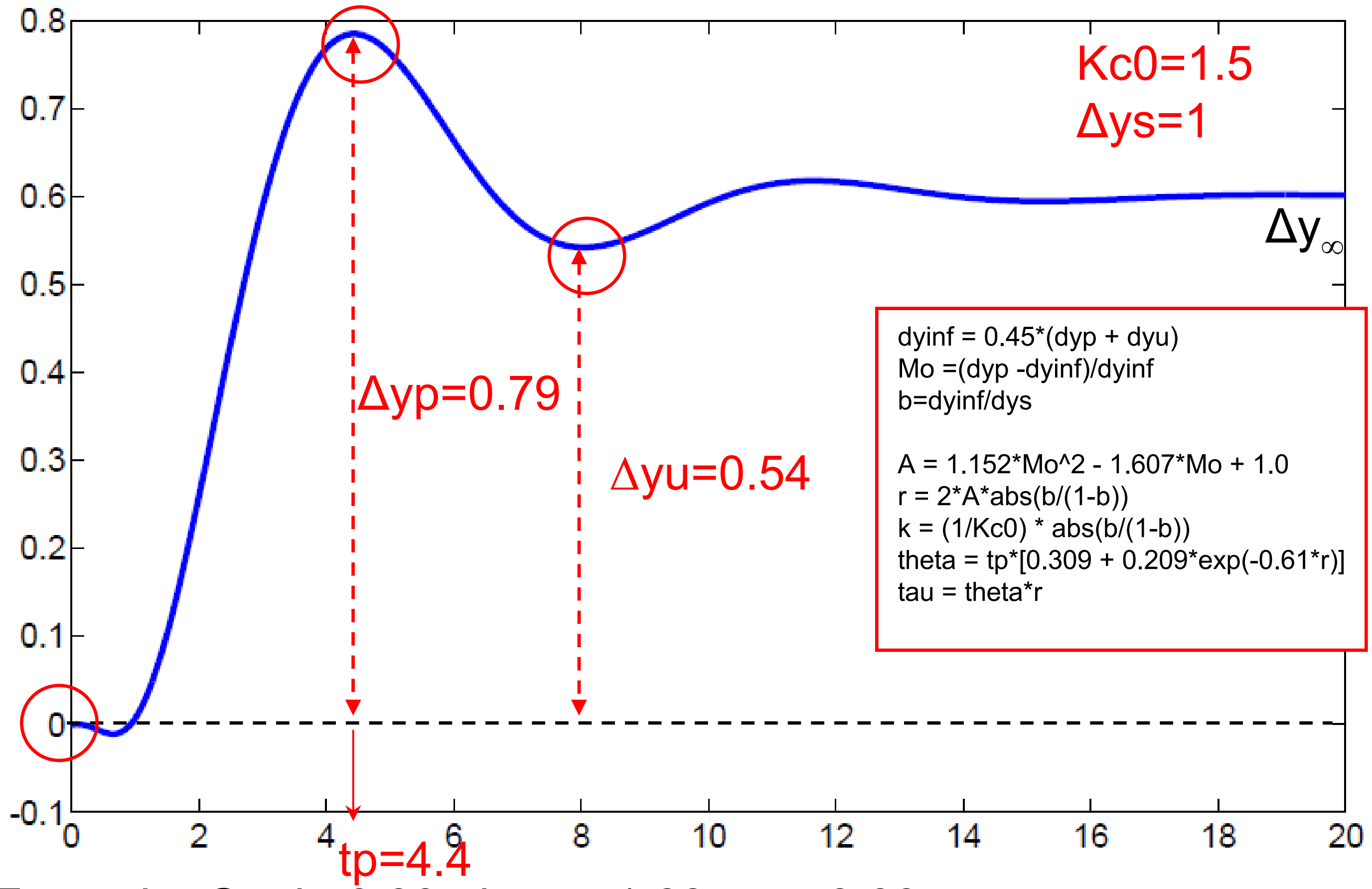
Questions:

1. How good is really the SIMC-rule?
   – Answer: Pretty close to optimal, except for time delay process
2. Can it be improved?
   – Yes, to improve for time delay process: Replace τ_1 by $\tau_1+\theta/3$ in rule to get ”Improved-SIMC”

- Not possible to do much better with any other PI-controller!

Reference: C. Grimholt and S. Skogestad, “The improved SIMC method for PI controller tuning”, IFAC-conference PID’12, Brescia, Italy, March 2012

Model from closed-loop response with P-controller



Example: Get k=0.99, theta =1.68, tau=3.03

Ref: Shamssuzzoha and Skogestad (JPC, 2010)
+ modification by C. Grimholt (Project, NTNU, 2010; see also new from PID-book 2011)