PID tuning using the SIMC rule

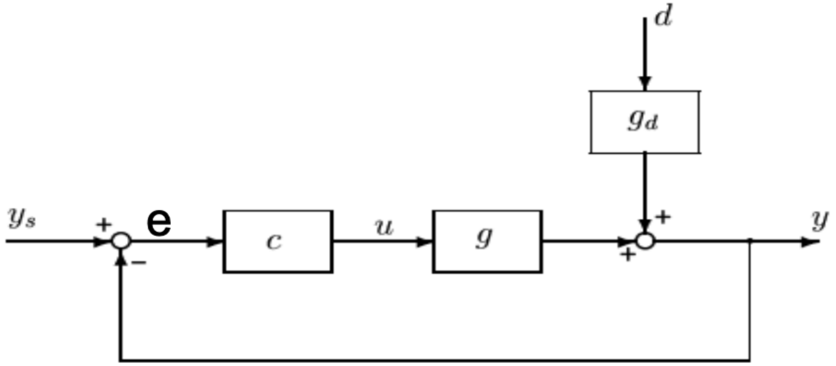
Sigurd Skogestad

NTNU

Lecture outline

- SIMC rule for first order systems
- Closed loop tuning
- Half-rule for higher order models

PID controller



- Time domain ("ideal" PID)

$$u(t) = u_0 + K_c \left(e(t) + \frac{1}{\tau_I} \int_0^t e(\xi)\, d\xi + \tau_D \frac{de}{dt} \right)$$

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

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

- Usually $\tau_D = 0$. Only two parameters left (K_C and τ_I)…
- How difficult can it be?
  - Surprisingly difficult without systematic approach!

PID tuning – wikipedia - 2023

Choosing a tuning method

Method	Advantages	Disadvantages
Manual tuning	No mathematics required; online.	Requires experienced personnel. [citation needed]
Ziegler–Nichols [b]	Proven method; online.	Process upset, some trial-and-error, very aggressive tuning. [citation needed]
Tyreus Luyben	Proven method; online.	Process upset, some trial-and-error, very aggressive tuning [citation needed]
Software tools	Consistent tuning; online or offline - can employ computer-automated control system design (*CAutoD*) techniques; may include valve and sensor analysis; allows simulation before downloading; can support non-steady-state (NSS) tuning.	Some cost or training involved.[21]
Cohen–Coon	Good process models.	Some mathematics; offline; only good for first-order processes. [citation needed]
Åström-Hägglund	Can be used for auto tuning; amplitude is minimum so this method has lowest process upset	The process itself is inherently oscillatory.[citation needed]
SIMC	Has tuning parameter, analytically derived, works also on delay processes(where ZN does not work)	The process must not be oscillatory

No, less agressive Version of ZN

BAD method

??

https://en.wikipedia.org/wiki/PID_controller#Loop_tuning

Hm…. SIMC tuning is not mentioned – maybe someone can update wiki

NTNU

PID tuning – wikipedia - 2024

Choosing a tuning method

Method	Advantages	Disadvantages
Manual tuning	No mathematics required; online.	This is an iterative, experience-based, trial-and-error procedure that can be relatively time consuming. Operators may find "bad" parameters without proper training.[22]
Ziegler–Nichols	Online tuning, with no tuning parameter therefore easy to deploy.	Process upsets may occur in the tuning, can yield very aggressive parameters. Does not work well with time-delay processes. [citation needed]
Tyreus Luyben	Online tuning, an extension of the Ziegler–Nichols method, that is generally less aggressive.	Process upsets may occur in the tuning; operator needs to select a parameter for the method which requires insight.
Software tools	Consistent tuning; online or offline – can employ computer-automated control system design (*CAutoD*) techniques; may include valve and sensor analysis; allows simulation before downloading; can support non-steady-state (NSS) tuning.	"Black box tuning" that requires specification of an objective describing the optimal behaviour.
Cohen–Coon	Good process models[citation needed].	Offline; only good for first-order processes. [citation needed]
Åström-Hägglund	Unlike the Ziegler–Nichols method this will not introduce a risk of loop instability. Little prior process knowledge required.[23]	May give excessive derivative action and sluggish response. Later extensions resolve these issues, but require a more complex tuning procedure.[23]
Simple control rule (SIMC)	Analytically derived, works on time delayed processes, has an additional tuning parameter that allows additional flexibility. Tuning can be performed with step-response model.[22]	Offline method; cannot be applied to oscillatory processes. Operator must choose the additional tuning parameter.[22]

Still a BAD method, Delete

https://en.wikipedia.org/wiki/PID_controller#Loop_tuning

SIMC tuning is on the list !! …. Thank you (I don't know who)

Excellent work – especially considering that it was published only 3 years after the PID controller came on the market (Taylor Model 100 Fullscope, 1939)

***Trans. ASME*, 64, 759-768 (Nov. 1942).**

Optimum Settings for Automatic Controllers

By J.G. ZIEGLER[1] and N. B. NICHOLS[2] • ROCHESTER, N. Y.

In this paper, the three principle control effects found in present controllers are examined and practical names and units of measurement are proposed for each effect.

varying its output air pressure, repositions a diaphragm-operated valve. The controller may be measuring temperature, pressure, level, or any other variable, but we will completely divorce the

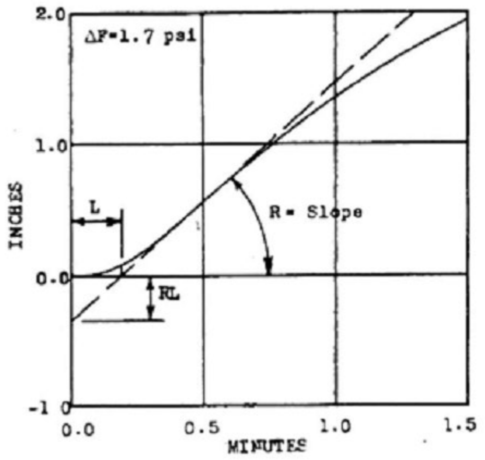


FIG. 8 REACTION CURVE

Reset-Rate Determination From Reaction Curve. Since the period of oscillation at the ultimate sensitivity proves to be 4 times the lag. A substitution of 4 *L* for P_u in previous equations for optimum reset rate gives an equation expressing this reset rate in terms of lag. For a controller with proportional reset responses, the optimum settings become

$$\textit{Sensitivity} = \frac{0.9}{R_1 L}\ \textit{psi per in.}$$

$$\text{Reset Rate} = \frac{0.3}{L}\ \text{per min}$$

At these settings the period will be about 5.7L, having been increased, by both the lowering of sensitivity and the addition of automatic reset.

My notation:
Model: $R = k', L = \theta$
PI-settings:
$K_c = \frac{0.9}{k'}\frac{1}{\theta}$
$\tau_I = 3.3\theta$

Disadvantages Ziegler-Nichols:
1. Aggressive settings
2. No tuning parameter
3. Poor for processes with large time delay (θ)

Comment:
Similar to SIMC for integrating process with τ_c=0 (aggressive!):
K_c = 1/k' 1/θ
τ_I = 4 θ

AMERICAN CONTROL CONFERENCE
San Diego, California
June 6-8, 1984

IMPLICATIONS OF INTERNAL MODEL CONTROL FOR PID CONTROLLERS

Manfred Morari
Sigurd Skogestad

Daniel F. Rivera

California Institute of Technology
Department of Chemical Engineering
Pasadena, California 91125

University of Wisconsin
Department of Chemical Engineering
Madison, Wisconsin 53706

252 *Ind. Eng. Chem. Process Des. Dev.* **1986**, *25*, 252–265

Internal Model Control. 4. PID Controller Design

Daniel E. Rivera, Manfred Morari,* and Sigurd Skogestad

Chemical Engineering, 206-41, California Institute of Technology, Pasadena, California 91125

For a large number of single input–single output (SISO) models typically used in the process industries, the Internal Model Control (IMC) design procedure is shown to lead to PID controllers, occasionally augmented with a first-order lag. These PID controllers have as their only tuning parameter the closed-loop time constant or, equivalently, the closed-loop bandwidth. On-line adjustments are therefore much simpler than for general PID controllers. As a special case, PI- and PID-tuning rules for systems modeled by a first-order lag with dead time are derived analytically. The superiority of these rules in terms of both closed-loop performance and robustness is demonstrated.

252 *Ind. Eng. Chem. Process Des. Dev.* **1986**, *25*, 252–265

Internal Model Control. 4. PID Controller Design

Daniel E. Rivera, Manfred Morari,* and Sigurd Skogestad

Chemical Engineering, 206-41, California Institute of Technology, Pasadena, California 91125

For a large number of single input–single output (SISO) models typically used in the process industries, the Internal Model Control (IMC) design procedure is shown to lead to PID controllers, occasionally augmented with a first-order lag. These PID controllers have as their only tuning parameter the closed-loop time constant or, equivalently, the ... al PID controllers. As a ... dead time are derived ... ›ustness is demonstrated.

Table I. IMC-Based PID Controller Parameters[a]

	model	$y/y_s = \tilde{g}_+ f$	controller	$k_c k$	τ_I	τ_D	τ_F	comments
A	$\frac{k}{\tau s+1}$	$\frac{1}{\epsilon s+1}$	$\frac{1}{k}\frac{\tau s+1}{\epsilon s}$	$\frac{\tau}{\epsilon}$	τ	–	–	–
B	$\frac{k}{(\tau_1 s+1)(\tau_2 s+1)}$	$\frac{1}{\epsilon s+1}$	$\frac{(\tau_1 s+1)(\tau_2 s+1)}{k\epsilon s}$	$\frac{\tau_1+\tau_2}{\epsilon}$	$\tau_1+\tau_2$	$\frac{\tau_1\tau_2}{\tau_1+\tau_2}$	–	–
C	$\frac{k}{\tau^2 s^2+2\zeta\tau s+1}$	$\frac{1}{\epsilon s+1}$	$\frac{\tau^2 s^2+2\zeta\tau s+1}{k\epsilon s}$	$\frac{2\zeta\tau}{\epsilon}$				
D	$k\frac{-\beta s+1}{\tau s+1}$	$\frac{-\beta s+1}{\epsilon s+1}$	$\frac{\tau s+1}{k(\beta+\epsilon)s}$	$\frac{\tau}{\beta+\epsilon}$				
E	$k\frac{-\beta s+1}{\tau s+1}$	$\frac{-\beta s+1}{(\beta s+1)(\epsilon s+1)}$	$\frac{\tau s+1}{ks(\beta\epsilon s+2\beta+\epsilon)}$	$\frac{\tau}{2\beta+\epsilon}$				
F	$k\frac{-\beta s+1}{\tau^2 s^2+2\zeta\tau+1}$	$\frac{-\beta s+1}{\epsilon s+1}$	$\frac{\tau^2 s^2+2\zeta\tau s+1}{k(\beta+\epsilon)s}$	$\frac{2\zeta\tau}{\beta+\epsilon}$				
G	$k\frac{-\beta s+1}{\tau^2 s^2+2\zeta\tau s+1}$	$\frac{-\beta s+1}{(\beta s+1)(\epsilon s+1)}$	$\frac{\tau^2 s^2+2\zeta\tau s+1}{k(\beta\epsilon s+2\beta+\epsilon)s}$	$\frac{2\zeta\tau}{2\beta+\epsilon}$				
H	$\frac{k}{s}$	$\frac{1}{\epsilon s+1}$	$\frac{1}{k\epsilon}$	$\frac{1}{\epsilon}$				
I	$\frac{k}{s}$	$\frac{2\epsilon+1}{(\epsilon s+1)^2}$	$\frac{2\epsilon s+1}{k\epsilon^2 s}$	$\frac{2}{\epsilon}$				
J	$\frac{k}{s(\tau s+1)}$	$\frac{1}{\epsilon s+1}$	$\frac{\tau s+1}{k\epsilon}$	$\frac{1}{\epsilon}$	–	τ	–	–
K	$\frac{k}{[illegible]}$	$\frac{2\epsilon s+1}{[illegible]}$	$\frac{(\tau s+1)(2\epsilon s+1)}{[illegible]}$	$\frac{2\epsilon+\tau}{[illegible]}$	$2\epsilon+\tau$	$\frac{2\epsilon\tau}{[illegible]}$	–	(6)

Table II. IMC-Based PID Parameters for $g(s) = ke^{-\theta s}/(\tau s + 1)$ and Practical Recommendations for ϵ/θ

controller	kk_c	τ_I	τ_D	recom-mended ϵ/θ (> $0.1\tau/\theta$ *always*)
PID	$(2\tau + \theta)/(2\epsilon + \theta)$	$\tau + (\theta/2)$	$\tau\theta/(2\tau + \theta)$	>0.8
PI	$\theta/\tau = 0.1$	1.54		>1.7
improved PI	$(2\tau + \theta)/2\epsilon$	$\tau + (\theta/2)$		>1.7

Disadvantage IMC-PID (=Lambda tuning):
1. Many rules
2. Poor disturbance response for «slow» processes (with large τ_1/θ)

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.

NTNU

ELSEVIER

Journal of Process Control 13 (2003) 291–309

JOURNAL OF PROCESS CONTROL

www.elsevier.com/locate/jprocont

Simple analytic rules for model reduction and PID controller tuning☆

Sigurd Skogestad*

Department of Chemical Engineering, Norwegian University of Science and Technology, N-7491 Trondheim, Norway

Received 18 December 2001; received in revised form 25 June 2002; accepted 11 July 2002

Abstract

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

Keywords: Process control; Feedback control; IMC; PI-control; Integrating process; Time delay

1. Introduction

Although the proportional-integral-derivative (PID) controller has only three parameters, it is not easy, without a systematic procedure, to find good values (settings) for them. In fact, a visit to a process plant will usually show that a large number of the PID controllers are poorly tuned. The tuning rules presented in this paper have developed mainly as a result of teaching this material, where there are several objectives:

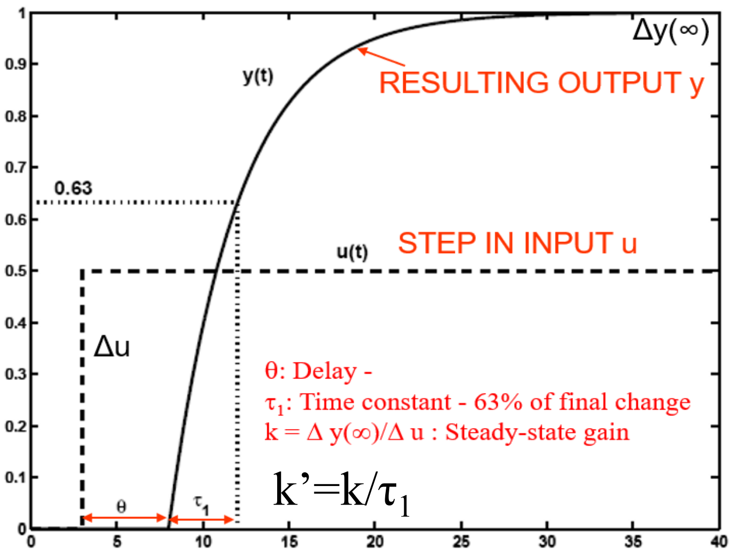
1. The tuning rules should be well motivated, and preferably model-based and analytically derived.
2. They should be simple and easy to memorize.
3. They should work well on a wide range of processes.

Step 2. Derive model-based controller settings. PI-settings result if we start from a first-order model, whereas PID-settings result from a second-order model.

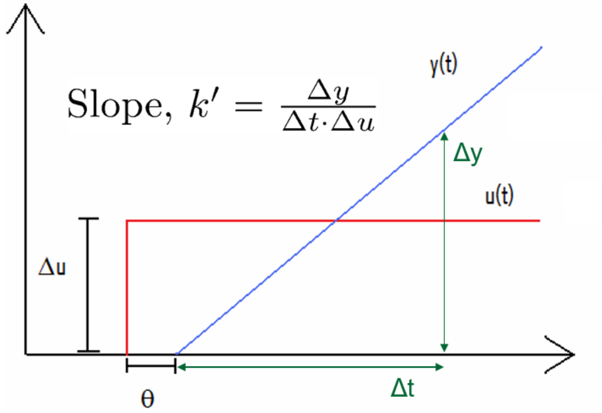
There has been previous work along these lines, including the classical paper by Ziegler amd Nichols [1], the IMC PID-tuning paper by Rivera et al. [2], and the closely related direct synthesis tuning rules in the book by Smith and Corripio [3]. The Ziegler–Nichols settings result in a very good disturbance response for integrating processes, but are otherwise known to result in rather aggressive settings [4,5], and also give poor performance for processes with a dominant delay. On the other hand, the analytically derived IMC-settings in [2] are known to result in a poor disturbance response for integrating processes (e.g., [6,7]), but are robust and

Summary SIMC PID tuning rule

$$u(t) = K_c e(t) + K_c \tau_D \frac{de(t)}{dt} + \overbrace{\underbrace{\frac{K_c}{\tau_I}\int_{t_0}^{t} e(t')dt' + u_0}_{\text{bias}=b}}^{u_I}$$



Step response integrating process



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

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

$$\tau_D = \tau_2$$

Only one tuning parameter:
Closed-loop time constant:

$$\tau_c \geq \theta$$

(gives Gain Margin>3)

+ Filter time constant, $\tau_F \leq \frac{\tau_c}{2}$

With anti windup

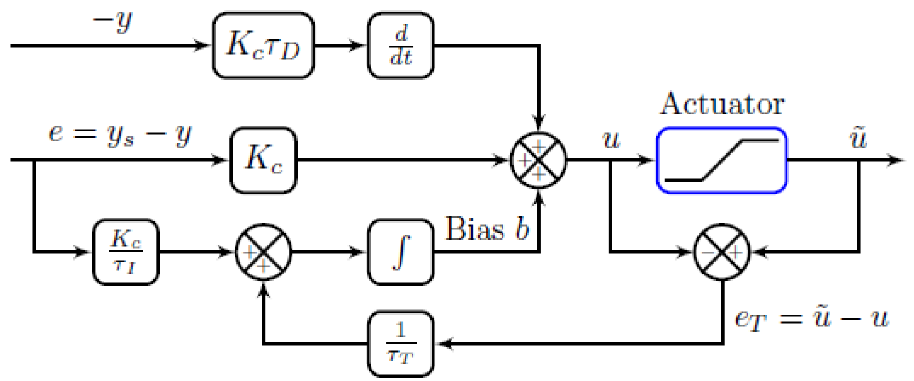


Figure 7: Recommended PID-controller implementation with anti-windup using tracking of the actual controller output ($\tilde{u}$), and without D-action on the setpoint. (Åström & Hägglund, 1988).

$\int$ = integral = $\frac{1}{s}$ in Laplace domain
$\frac{d}{dt}$ = derivative = s in Laplace domain
K_c = controller gain
τ_I = integral time [s, min]
τ_D = derivative time [s, min]
τ_T = tracking time constant for anti-windup [s, min]

Derivation of SIMC-PID tuning rules

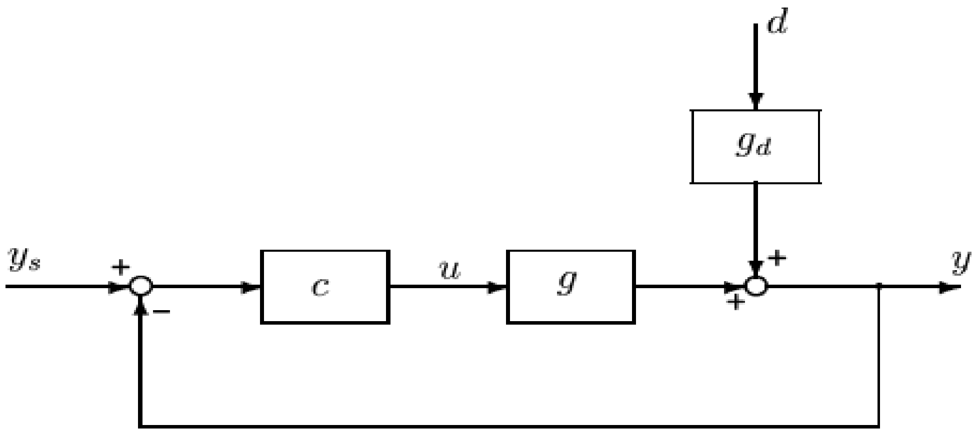
- PI-controller (based on first-order model)

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

- For second-order model add D-action.
  - For our purposes, simplest with the “series” (cascade) PID-form:

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

Basis: Direct synthesis (IMC)



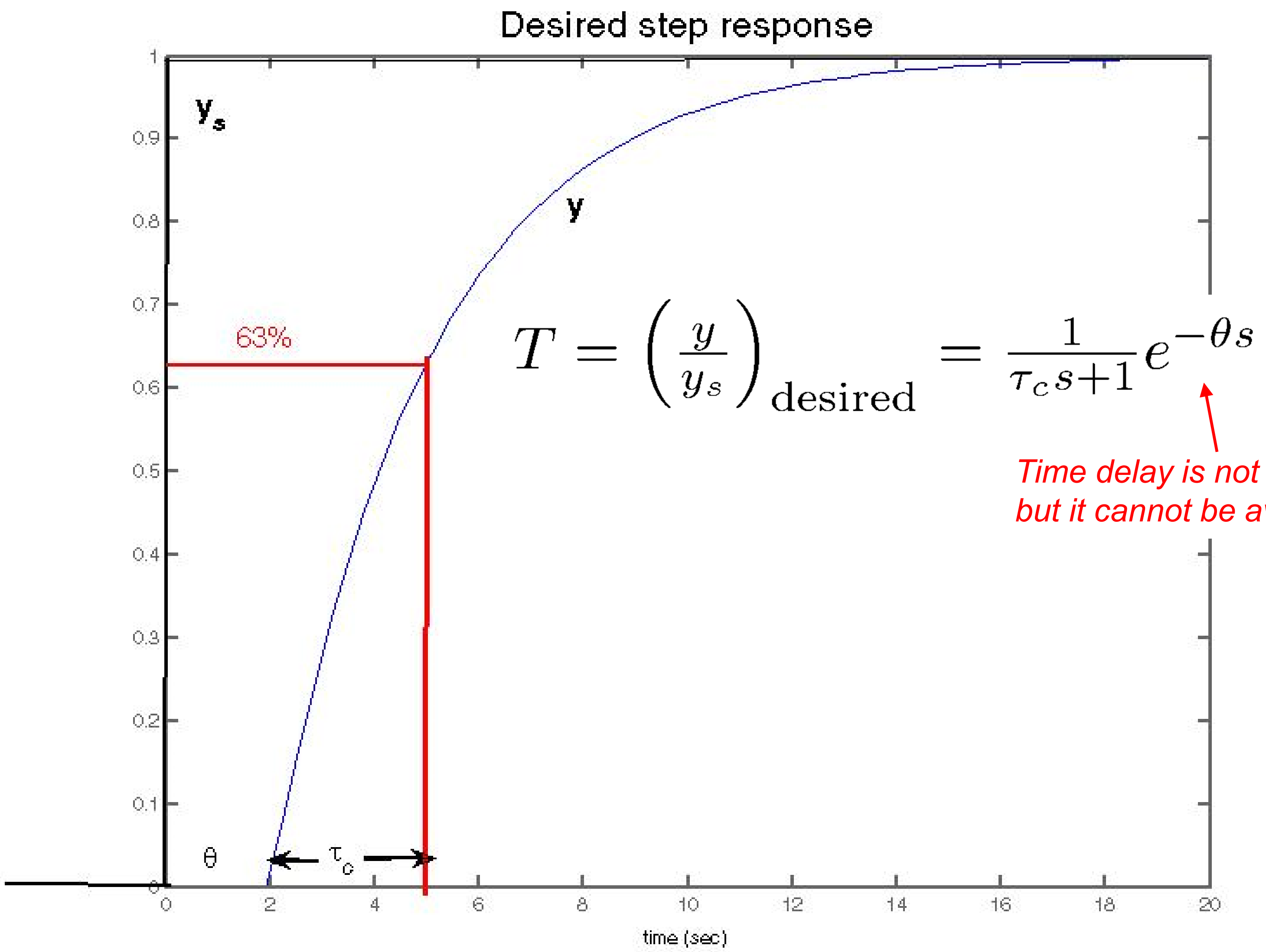
Closed-loop response to setpoint change:

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

Idea: specify desired response T and from this get the controller:

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

SIMC-tunings



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

Time delay is not really desired but it cannot be avoided

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

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

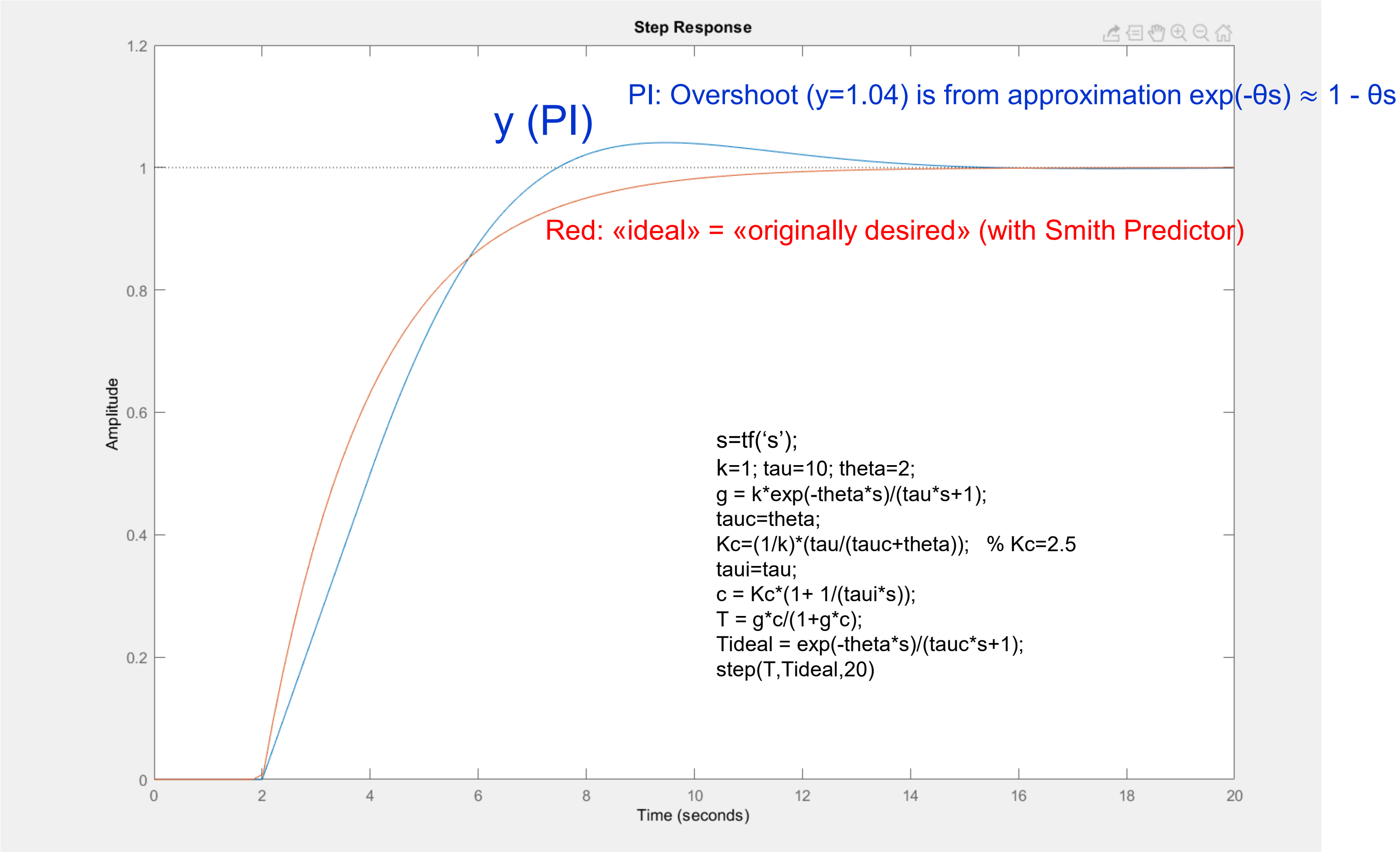
- τ_c is the sole tuning parameter

IMC-tuning is the same as "Lambda-tuning": τ_c is sometimes called λ

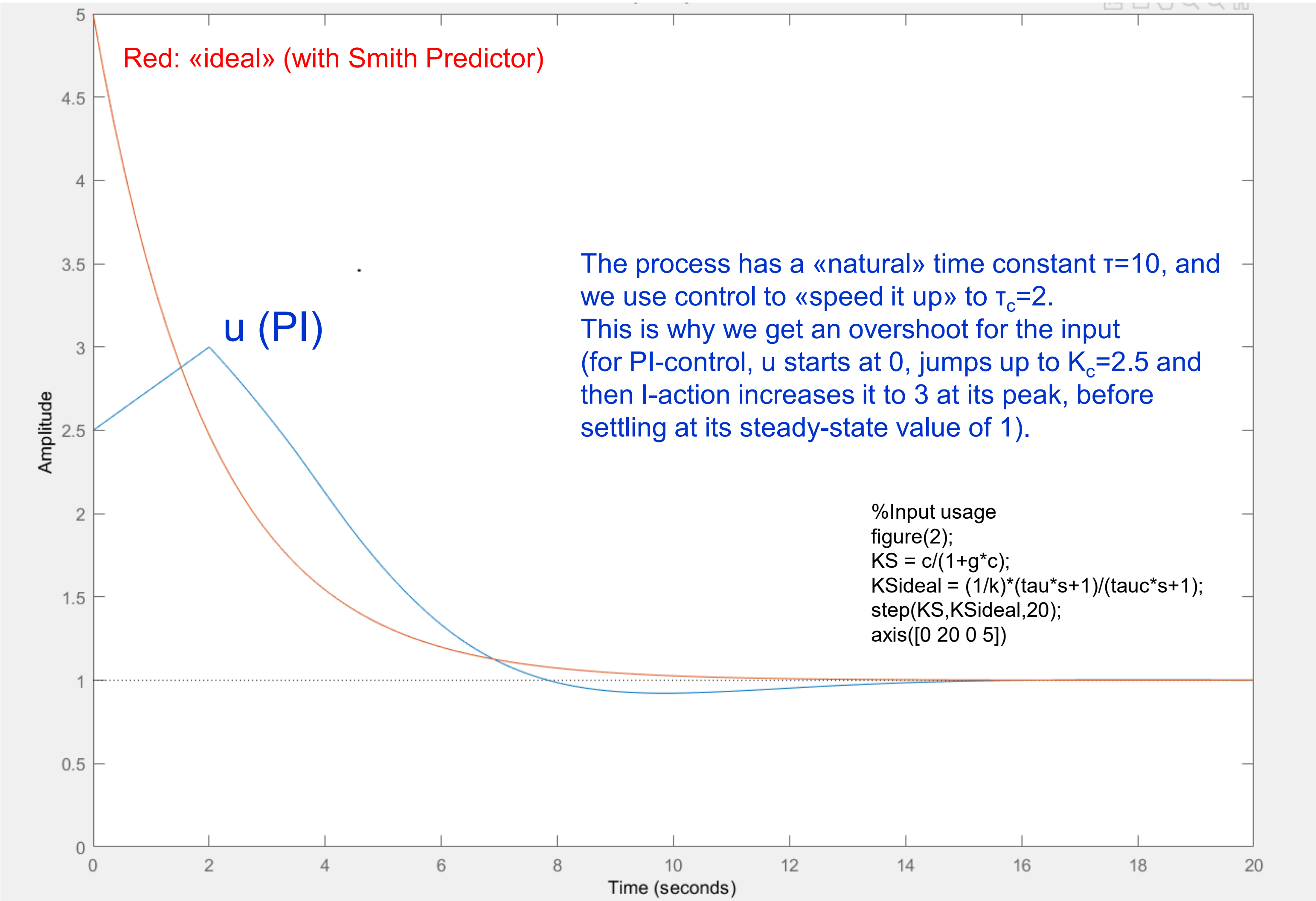
Surprisingly, this PID-controller is generally better, or at least more robust with respect to changes in the time delay θ, than the Smith Predictor controller from which it was derived. We are lucky ☺.

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)* .

Example step setpoint response (with choice $\tau_c = \theta = 2$)

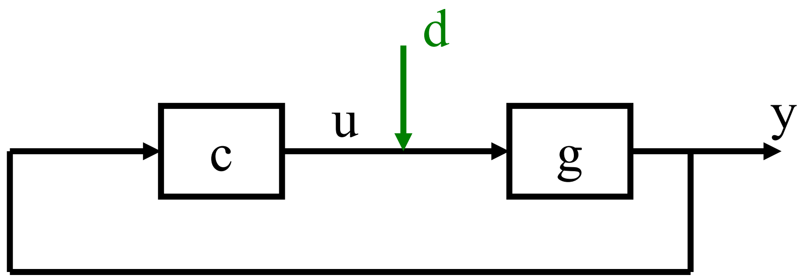


Input usage for setpoint response



Integral time

- Found: Integral time = dominant time constant ($\tau_I = \tau_1$)
- Gives P-controller for integrating process ($\tau_I = \infty$)
  - This works well for setpoint changes
  - But: τ_I 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!

Effect of decreasing τ_I

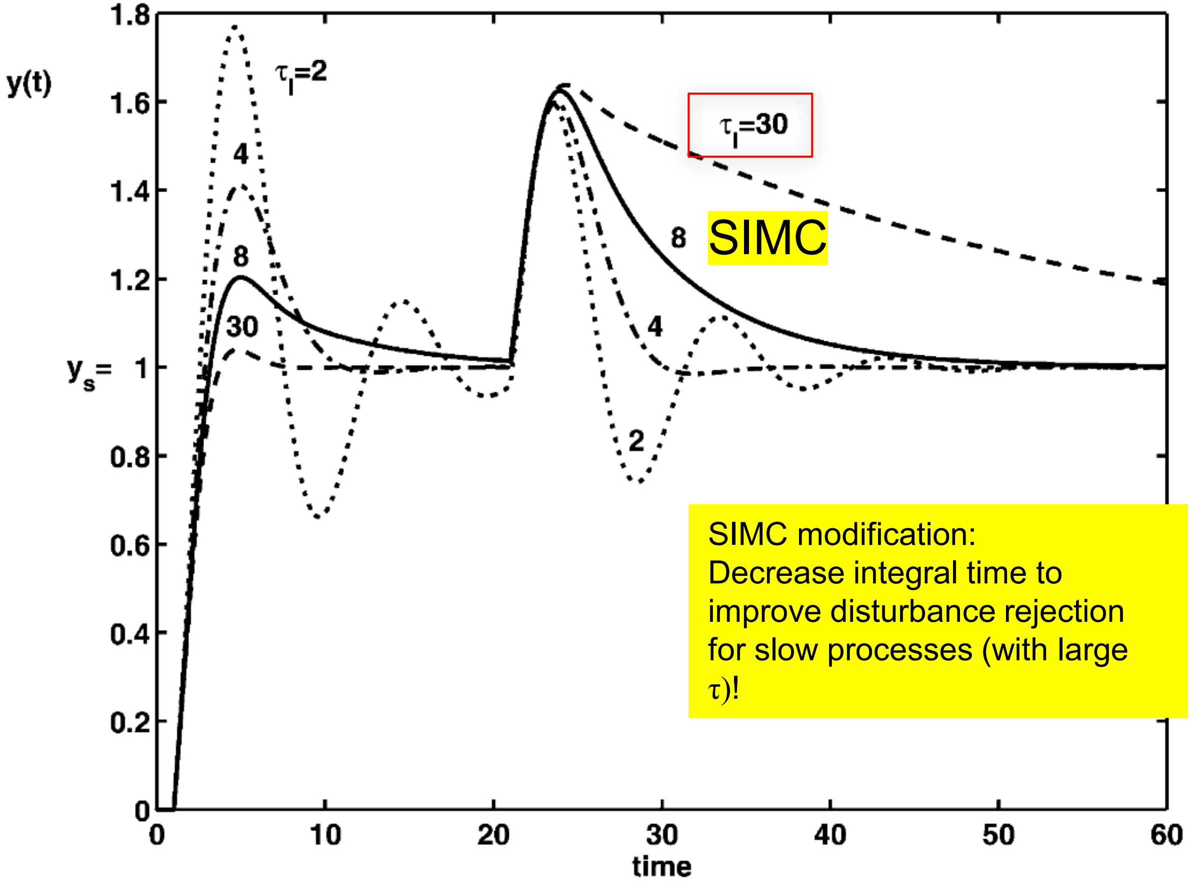


Fig. 3. Effect of changing the integral time τ_I for PI-control of "almost integrating" process $g(s) = e^{-s}/(30s+1)$ with $K_c = 15$. Unit setpoint change at $t = 0$; load disturbance of magnitude 10 at $t = 20$.

Integral time correction

- Want to reduce the integral time for "integrating" processes
- But to avoid "slow oscillations" (not caused by the delay θ) we must require $k'K_c\tau_I \geq 4$, which with the SIMC-rule for K_c gives:

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

- Proof:

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

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:

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

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

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

Conclusion: SIMC-PID Tuning Rules

For cascade form PID controller:

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

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

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

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: τ_c

Some special cases

Process	$g(s)$	K_c	τ_I	$\tau_D^{(4)}$
First-order	$k\frac{e^{-\theta s}}{(\tau_1 s+1)}$	$\frac{1}{k}\frac{\tau_1}{\tau_c+\theta}$	$\min\{\tau_1, 4(\tau_c+\theta)\}$	-
Second-order, eq.(4)	$k\frac{e^{-\theta s}}{(\tau_1 s+1)(\tau_2 s+1)}$	$\frac{1}{k}\frac{\tau_1}{\tau_c+\theta}$	$\min\{\tau_1, 4(\tau_c+\theta)\}$	τ_2
Pure time delay(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 $K_I \stackrel{\text{def}}{=} \frac{K_c}{\tau_I} = \frac{1}{k(\tau_c+\theta)}$.

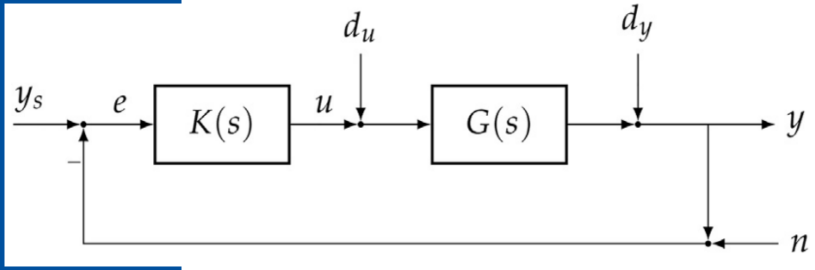
One tuning parameter: τ_c

(1)(*) Note that we get pure I-controller for static process with delay.

Choice of SIMC-tuning parameter τ_c.

1. Trade-off between robustness (Ms) and performance (J=IAE)

NTNU



2.1. Performance

In this paper, we quantify performance in terms of the integrated absolute error (IAE),

$$\text{IAE} = \int_0^\infty |y(t) - y_s(t)| dt. \tag{7}$$

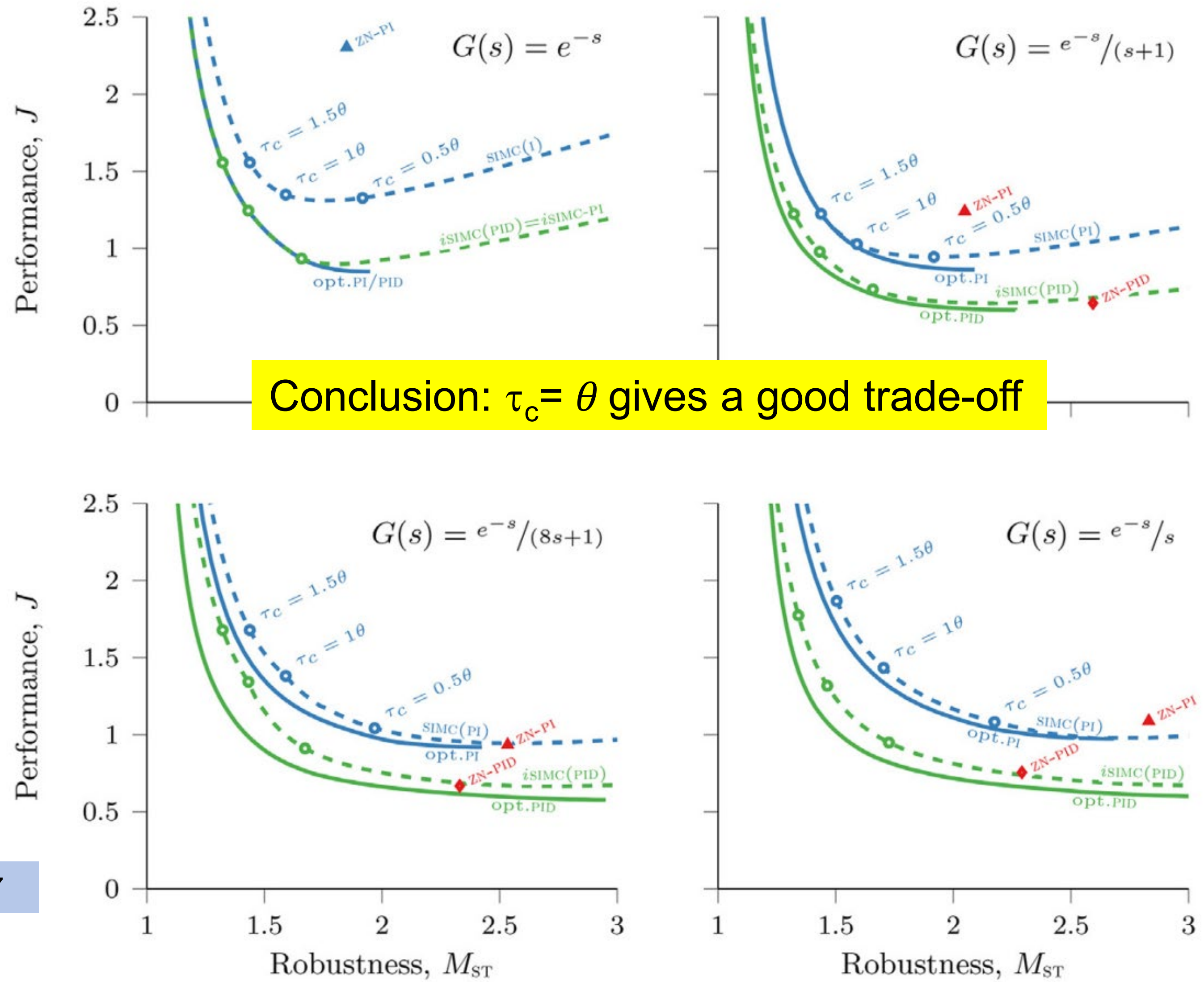
To balance the servo/regulatory trade-off we choose a weighted average of the IAE for a step input disturbance d_u (load disturbance) and a step output disturbance d_y:

$$J(p) = 0.5 \left(\frac{\text{IAE}_{dy}(p)}{\text{IAE}^\circ_{dy}} + \frac{\text{IAE}_{du}(p)}{\text{IAE}^\circ_{du}} \right) \tag{8}$$

where IAE°_{dy} and IAE°_{du} are weighting factors, and p denotes the controller parameters. Note that we do not consider setpoint responses, but instead output disturbances. For the system in Fig. 1,

Ms = Peak of |S(jω)| = 1/(smallest distance to (-1)-point). Want less than 1.7

C. Grimholt, S. Skogestad / Journal of Process Control 70 (2018) 36–46



Conclusion: $\tau_c = \theta$ gives a good trade-off

Optimal PI and PID control of first-order plus delay processes and evaluation of the original and improved SIMC rules

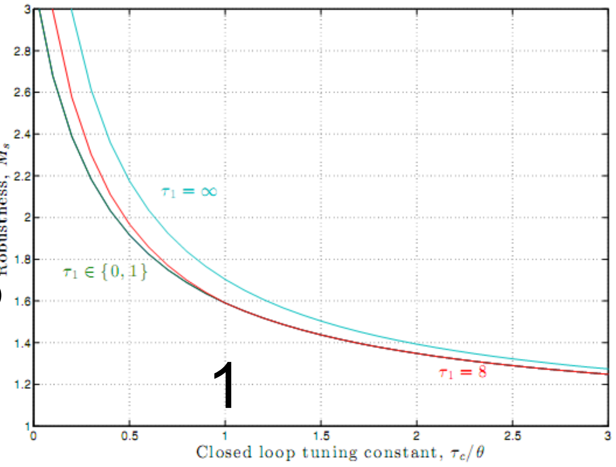
Chriss Grimholt, Sigurd Skogestad Journal of Process Control 70 (2018) 36–46

Choice of tuning SIMC-parameter τ_c

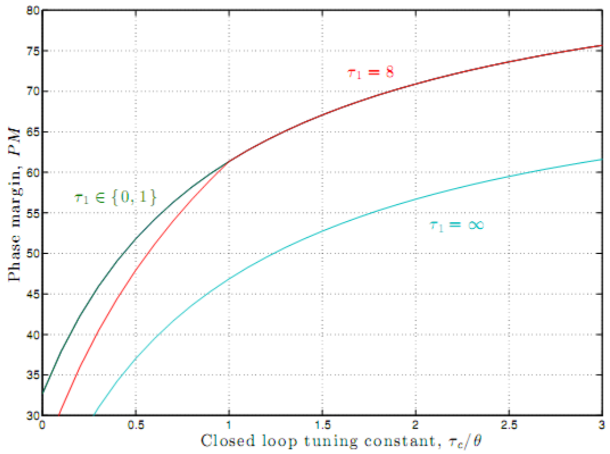
2. Relationship between τ_c and robustness (Ms, GM, PM, DM)

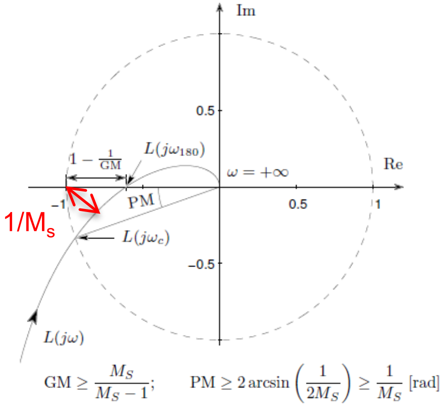
Conclusion: $\tau_c/\theta = 1$ gives a acceptable robustness (Ms=1.6, PM=60°, GM=3, DM=2)

M_s

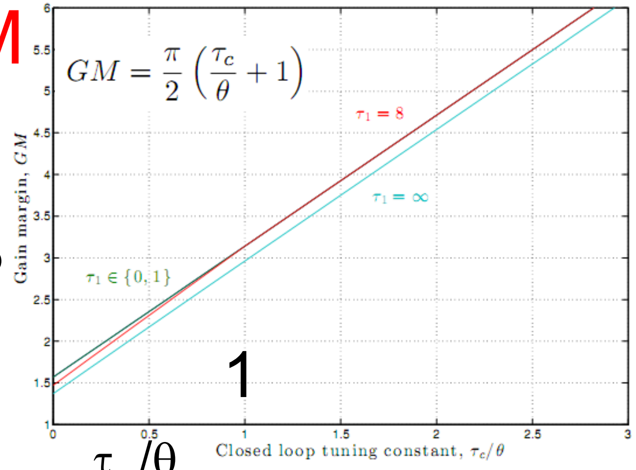


PM

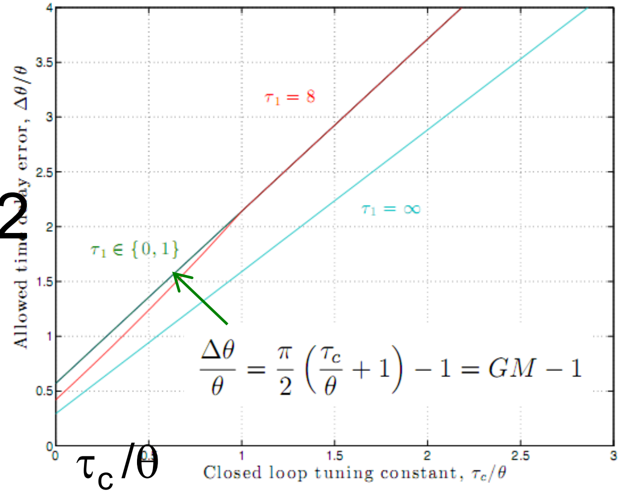




GM



DM= $\Delta\theta/\theta = PM/\omega_c$



SIMC: GM and DM increase linearly with τ_c

Chriss Grimholt and Sigurd Skogestad.
"Optimal PI-Control and Verification of the SIMC Tuning Rule".
Proceedings *IFAC conference on Advances in PID control* (PID'12), Brescia, Italy, 28-30 March 2012.

TUNING FOR FAST RESPONSE WITH GOOD ROBUSTNESS

$$\text{SIMC}: \quad \tau_c = \theta \qquad (4)$$

Gives:

$$K_c = \frac{0.5}{k}\frac{\tau_1}{\theta} = \frac{0.5}{k'}\cdot\frac{1}{\theta} \qquad (5)$$

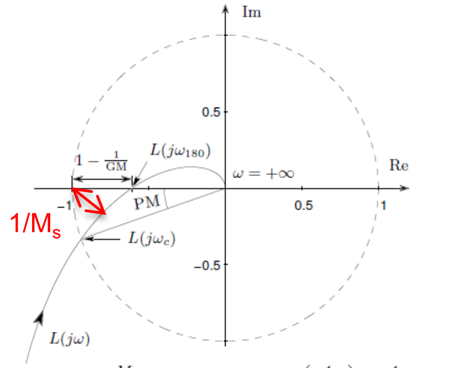
$$\tau_I = \min\{\tau_1, 8\theta\} \qquad (6)$$

$$\tau_D = \tau_2 \qquad (7)$$

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 $\tau_D = \tau_2$.



$\text{GM} \geq \frac{M_S}{M_S-1}; \quad \text{PM} \geq 2\arcsin\left(\frac{1}{2M_S}\right) \geq \frac{1}{M_S}$ [rad]

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

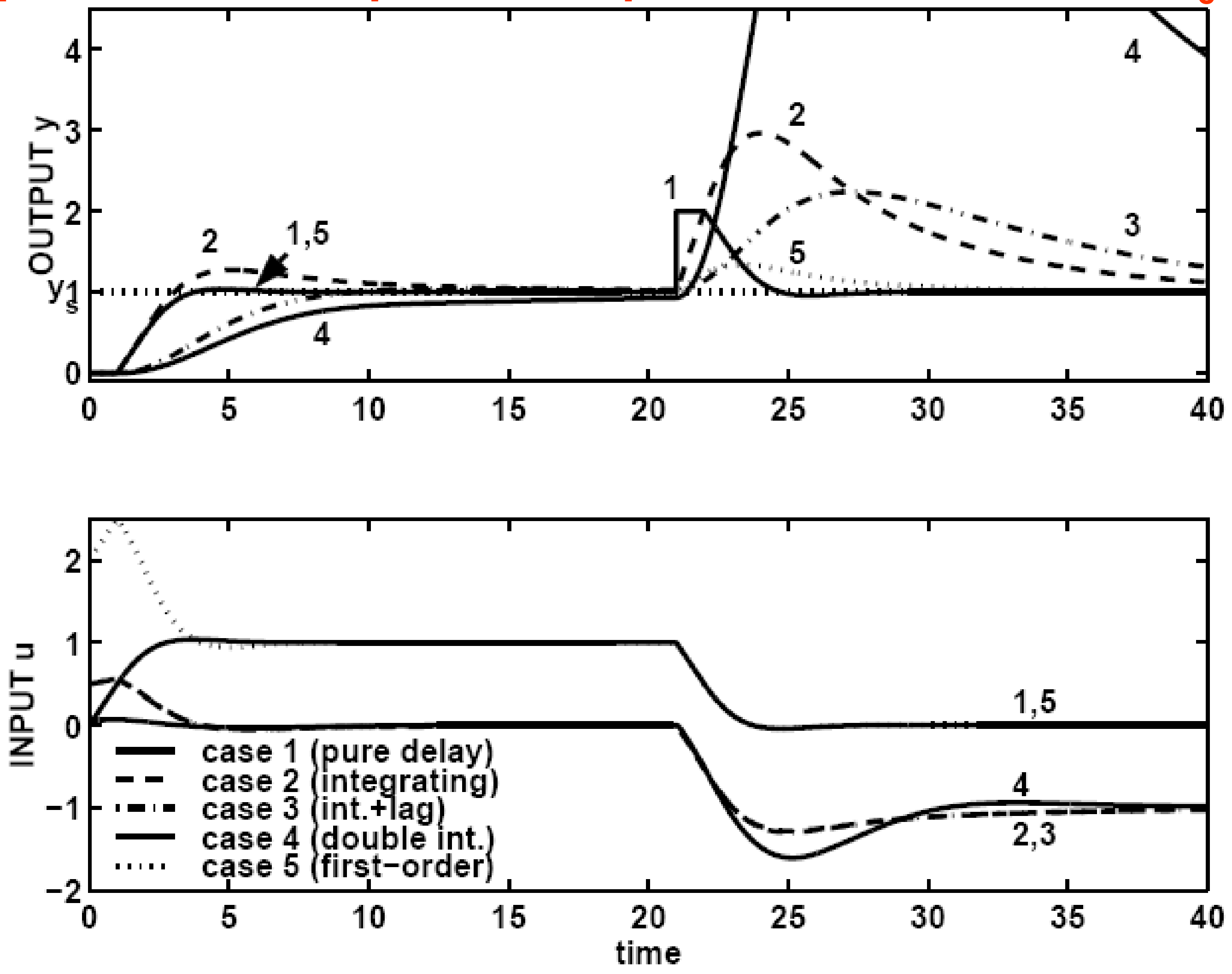


Figure 4: Responses using SIMC settings for the five time delay processes in Table 3 ($\tau_c = \theta$).

Unit setpoint change at $t = 0$; Unit load disturbance at $t = 20$.

Simulations are without derivative action on the setpoint.

Parameter values: $\theta = 1, k = 1, k' = 1, k'' = 1$.

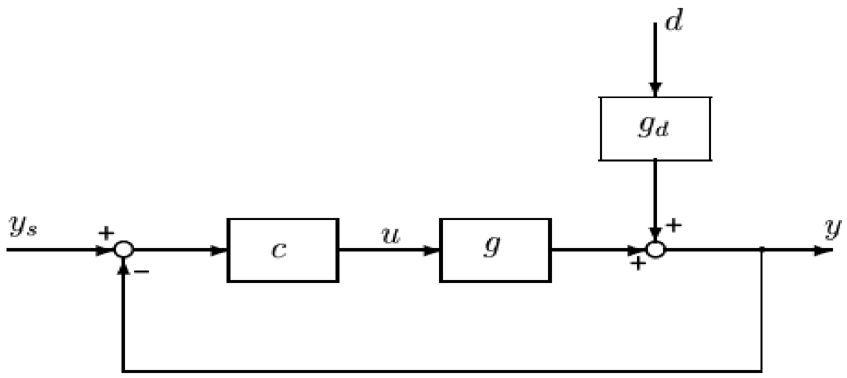
TIGHT CONTROL

- Example 2. Compare PI and PID

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

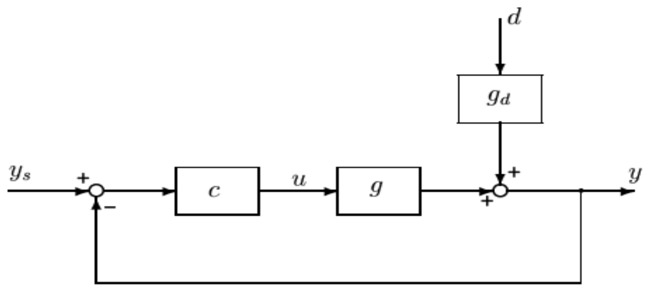
```
s=tf('s')
g=(-0.3*s+1)*(0.08*s+1)/((2*s+1)*(s+1)*(0.4*s+1)*(0.2*s+1)*(0.05*s+1)^3)
k=1;
tau1=2.5, tau2=0, theta=1.47, tauc=theta % 1st order
%tau1=2, tau2=1.2, theta=0.77, tauc=theta % 2nd order

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)
```

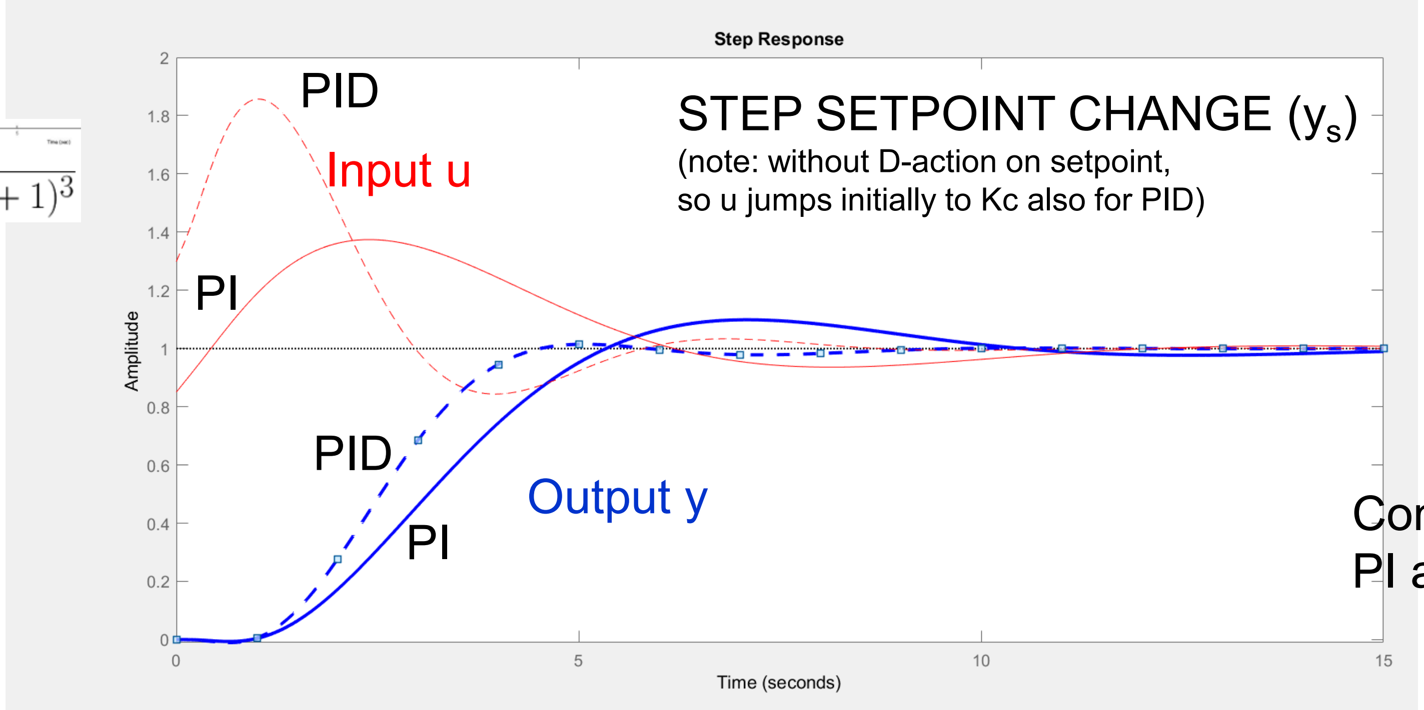
Note: tau2>theta , so 2nd order and PID gives perfromance improvement compared to PI



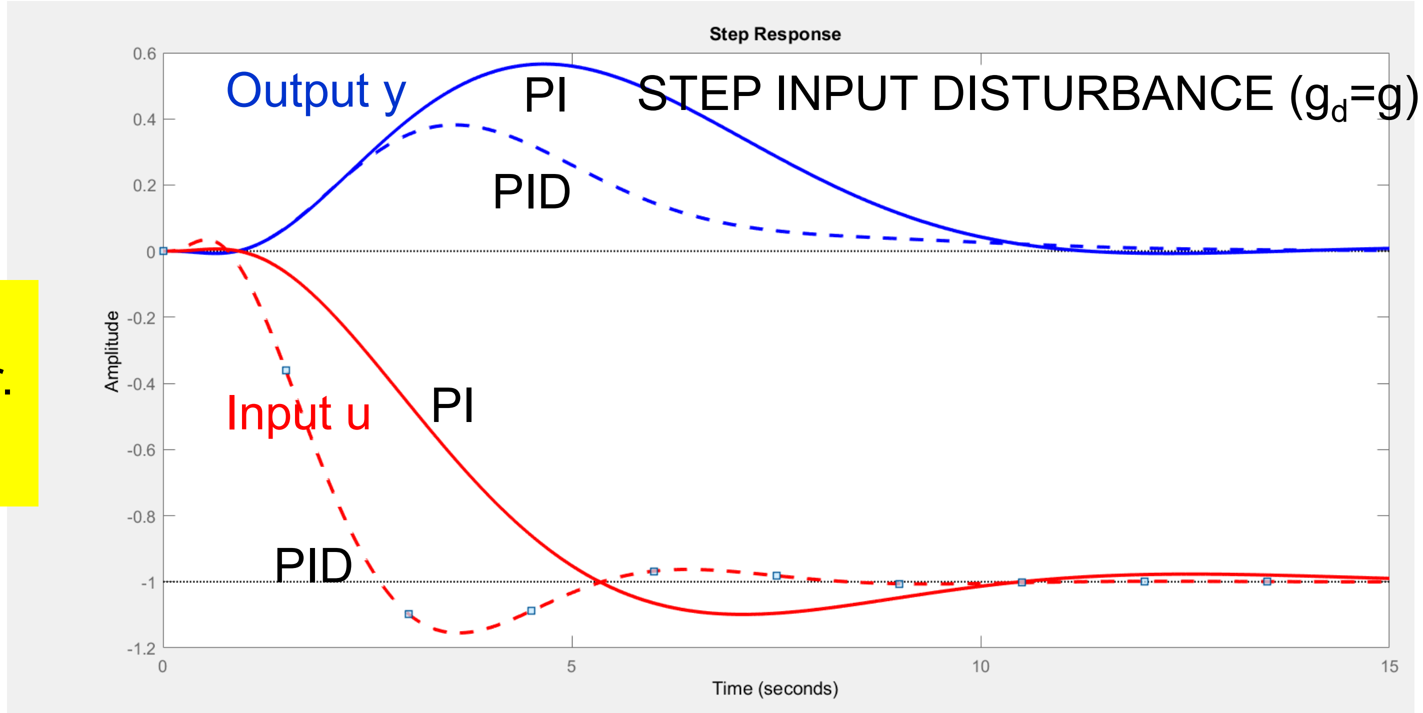
Example 2.

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





Comparison of PI and PID - -



Conclusion:

PID is quite a lot better.

(expected since tau2=1.2 > theta=0.77)

NTNU

DERIVATIVE ACTION ?

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

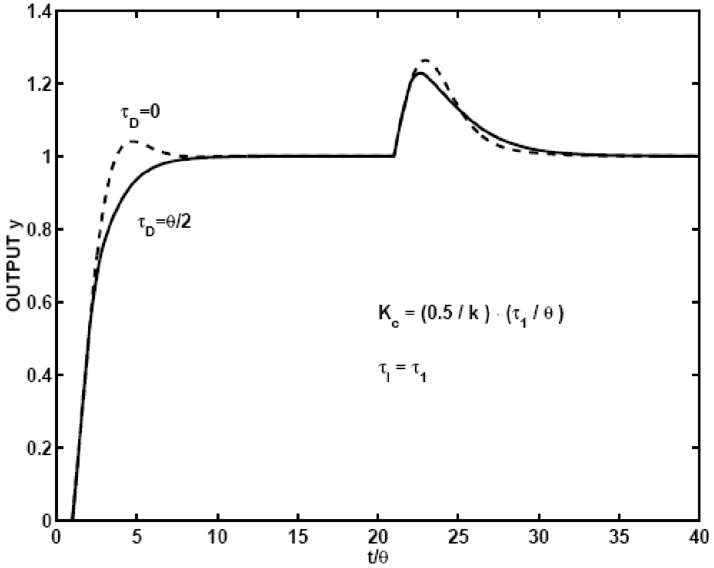


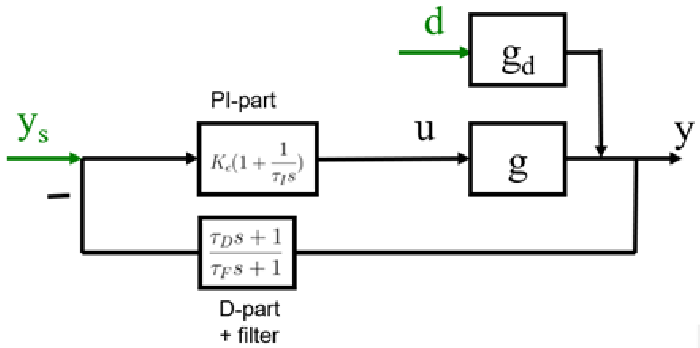
Figure 5: Setpoint change at $t = 0$. Load disturbance of magnitude 0.5 occurs at $t = 20$.

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

Conclusion D-action (for series-form PID):

1. Use PID with $\tau_D = \tau_2$ for dominant 2nd order processes with $\tau_2 > \theta$ (otherwise, add $\tau_2/2$ to effective delay θ and use PI)
2. Use derivative action (PID) for unstable processes, for example, a double integrating process (not so common in process control).
3. Derivative action (PID) can help a little to speed up response for a process with time delay (e.g. use $\tau_D = \theta/3$), but we then need to reduce τ_C (i.e., increase K_c) to get the performance benefit (e.g., reduce τ_C from θ to θ/2).
We did not do this in the above simulation, so this is why the benefit of D-action is small.
Example 3 (next slide): Compare SIMC-PI with PID with $\tau_D = \theta/3$ using $\tau_C = \theta$ (red curve, small benefit) and $\tau_C = \theta/2$ (yellow curve, less overshoot).
4. If you end up using a "large" τ_c, such that you have $\tau_c > 2\tau_D$ (approximately), then D-action is not helping much and you may consider PI-control instead. *Example: See above simulation which has $\tau_c = \theta$ and $\tau_D = \theta/2$, that is, $\tau_c = 2\tau_D$.*

Example 3 (disturbance response Dynea reactor):
(1) Effect of wrong tau-I, (2) Use of tau-D to reduce overshoot



$$g = 0.5\frac{e^{-4s}}{10s+1}$$

$$g_d = \frac{0.1}{200s+1}$$

SIMC $(\tau_c = \theta = 4)$:

$$K_c = 2.5, \tau_I = 10, \tau_D = \tau_F = 0$$

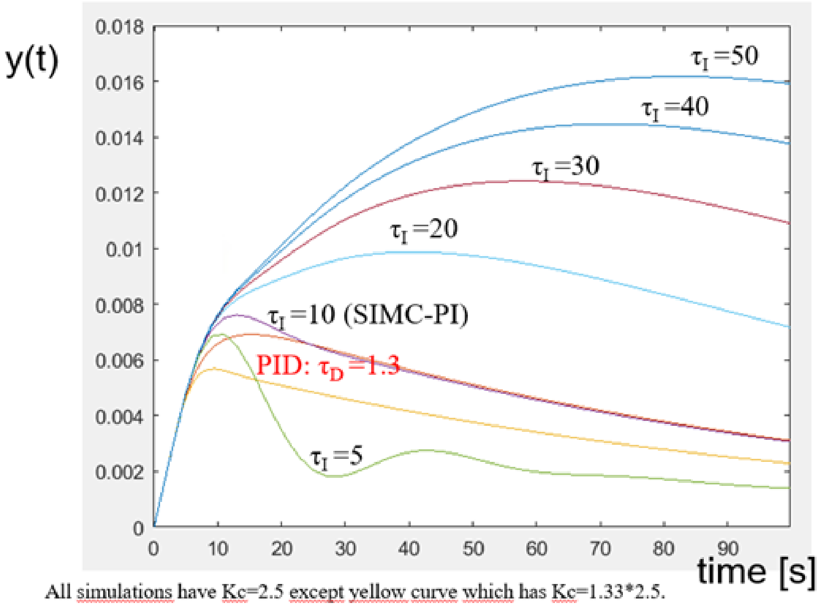
Response y(t) to step d=1.
Effect of changes in τ_I and τ_D

Conclusion:

- I-action is important: Follow SIMC-rule for τ_I
- D-action: Larger effect if we also increase K_c by 33% (yellow curve)

Note: The process (g) is fast compared to the disturbance dynamics (g_d). This is why the I-action is the most important.

SIMC-rule (from first-order process model g):
$K_c = \frac{1}{k}\frac{\tau}{\tau_c+\theta}$
$\tau_I = \min(\tau, 4(\tau_c+\theta))$



All simulations have Kc=2.5 except yellow curve which has Kc=1.33*2.5.
Red and yellow curves have D-action: τ_I=10, τ_D=θ/3=1.33, τ_F=τ_D/10=0.133

Note: Often D-action requires much more input usage - but in this example the input usage (not shown) is almost the same for all controllers. This is because the disturbance is so «slow» (almost integrating)

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

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

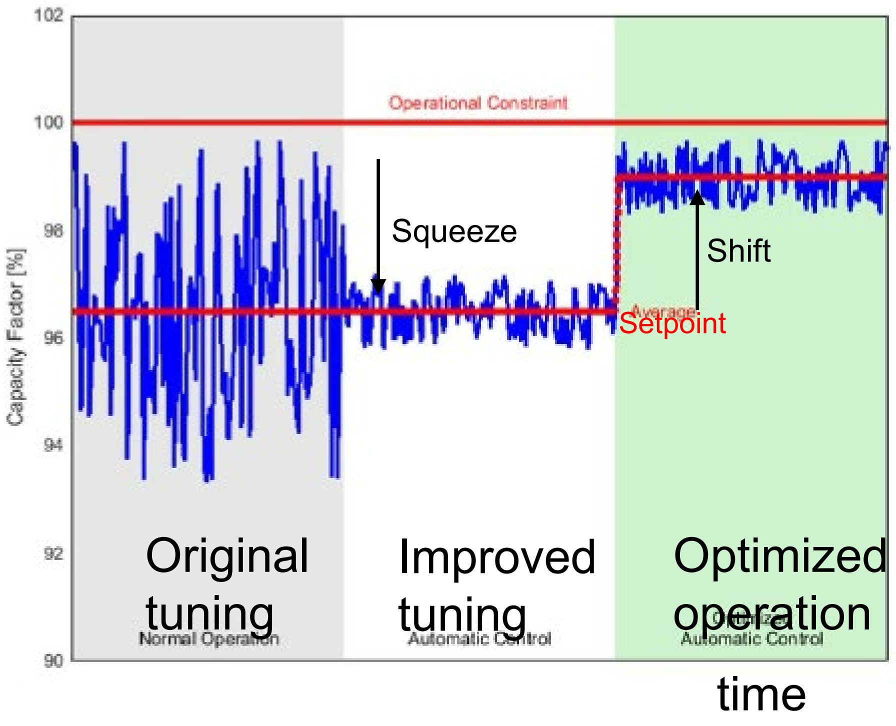
we use the following translation formulas

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

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.*

When do we need «tight control»? For hard constraints

«SQEEZE and SHIFT» RULE



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")
   - Select $\tau_c = \theta$ (effective delay)

2. SMOOTH CONTROL (τ_c large): Want "slowest possible control" subject to acceptable disturbance rejection
   - Prefer smooth control if fast control is not required

Tuning for smooth control

NTNU

- 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 ?
  - ANSWER: $\tau_{c,max} = 1/\omega_d$ (where ω_d is defined as the frequence where $|g_d(j\omega_d)| = y_{max}/d_{max}$)

Proof. y=Sgd d, where S=(1+L). Require |y|<ymax at all frequencies, so |S| < |gd| d/ymax at all frequencies.
The integral action takes care of most of the disturbance rejection, so usually, the «worst-case» frequency is where |S| reaches 1, which is approximately at wc=1/tauc.
So define wd as the frequency where (gd/g) d/ymax = 1 and we must require wc > wd or equivalently tauc < 1/wd. Thus we have tauc.maxc=1(wd.

This bound may be optimistic if there are disturbances with two or more «slow» poles, because then the worst-case frequency may be lower than wc.

Comment: An simpler (but sometimes conservative) answer is to select **Kc,min =|ud|/|ymax|** where |ud| is the input magnitude to reject the maximum disturbance. (Given Kc,min we may obtain the corresponding tauc,max using the SIMC-rule for Kc).
.
More detailed proof: S. Skogestad, ``Tuning for smooth PID control with acceptable disturbance rejection'', *Ind.Eng.Chem.Res*, **45** (23), 7817-7822 (2006).

NTNU

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.

NTNU

Level control: Can have both fast and slow oscillations

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

Fast oscillations: Caused by (effective) time delay
Here: Consider the common slow oscillations

P=period of oscillations = $2\pi/\omega$

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

Level control (integrating process): Can have both fast and slow oscillations

- Fast oscillations (K_c too high): $P < \pi \tau_I$
  - Caused by (effective) time delay
- Slow oscillations (K_c too low): $P > \pi \tau_I$
  - Caused by integral action in controller
  - Avoid slow oscillations: $k' K_c \tau_I \geq 4$.

P=period of oscillations = $2\pi/\omega$

NTNU

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]
>
> $0.1 \approx 1/\pi^2$

$$P_0 = \frac{2\pi}{\sqrt{1-\zeta^2}}\tau_0 = \frac{2\pi}{\sqrt{1-\zeta^2}}\sqrt{\frac{\tau_{I0}}{k' K_{c0}}} \approx 2\pi\sqrt{\frac{\tau_{I0}}{k' K_{c0}}} \qquad (39)$$

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

$$K_{c0} \cdot \tau_{I0} = (2\pi)^2 \frac{1}{k'}\left(\frac{\tau_{I0}}{P_0}\right)^2$$

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

$$\frac{K_c \tau_I}{K_{c0}\tau_{I0}} \geqslant \frac{1}{\pi^2} \cdot \left(\frac{P_0}{\tau_{i0}}\right)^2 \qquad (40)$$

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

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

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

Case study oscillating level

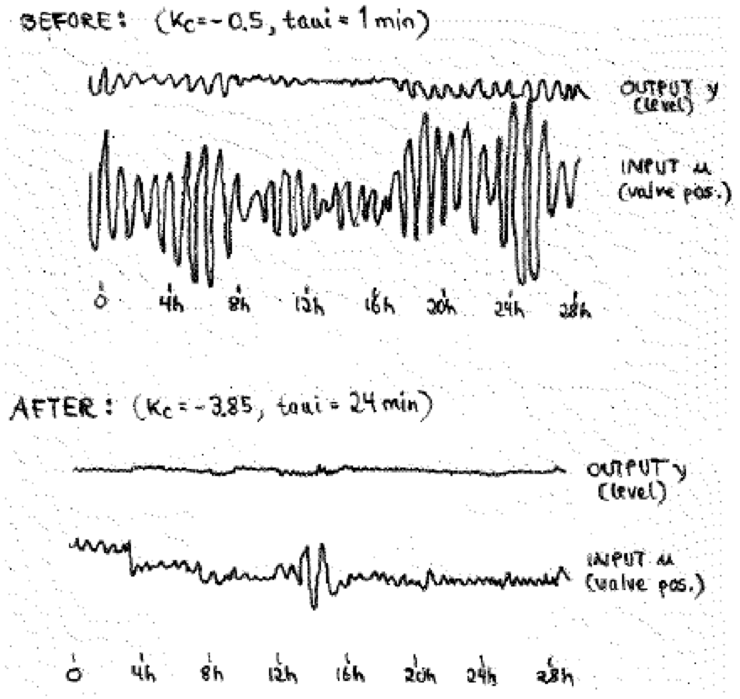
- We were called upon to solve a problem with oscillations in a distillation column
- Closer analysis: Problem was oscillating reboiler level in upstream column
- Use of Sigurd's rule solved the problem

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:

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



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

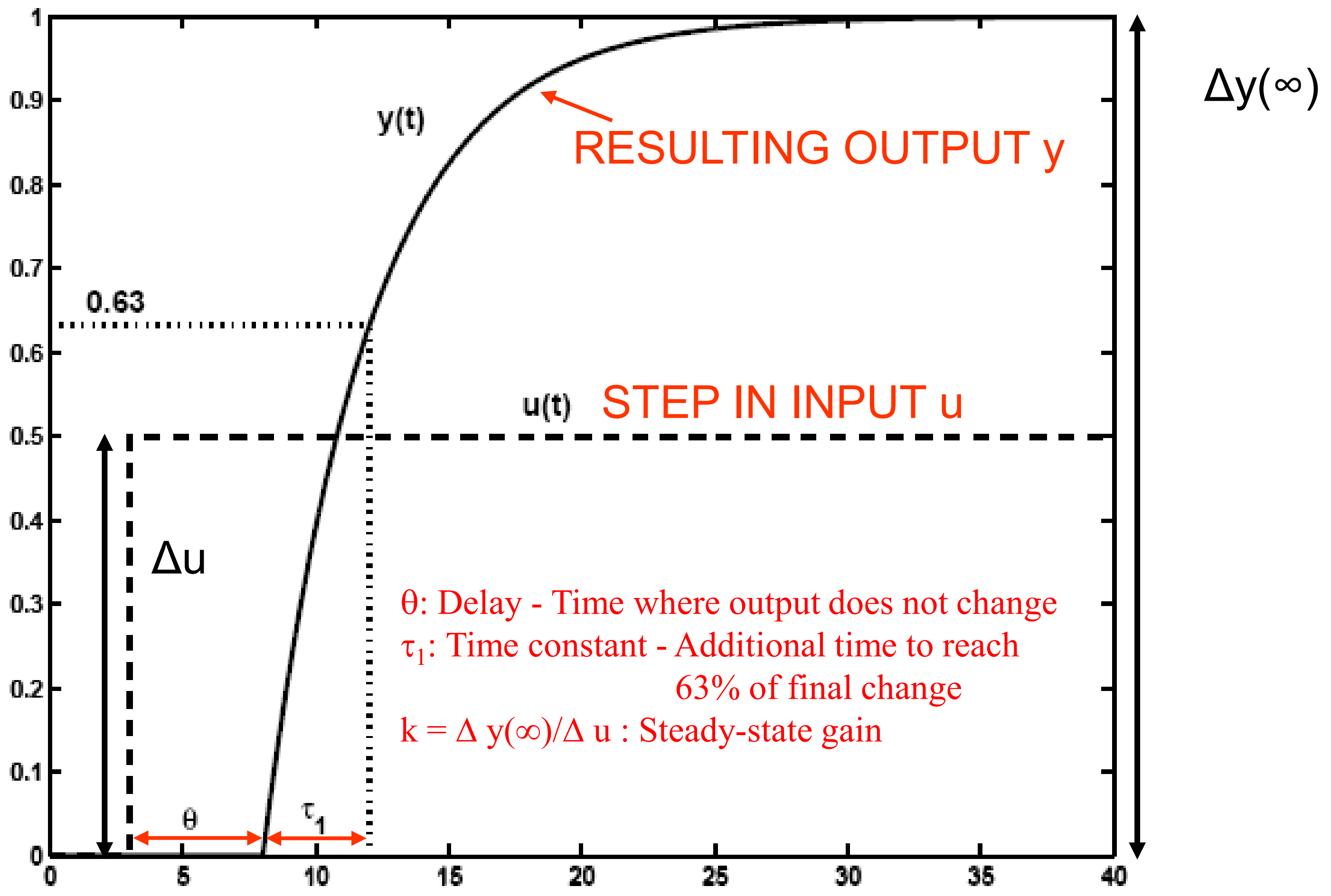
$$G(s) = \frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)} e^{-\theta s}$$

  – Recommend: Use second-order model only if $\tau_2 \geq \theta$

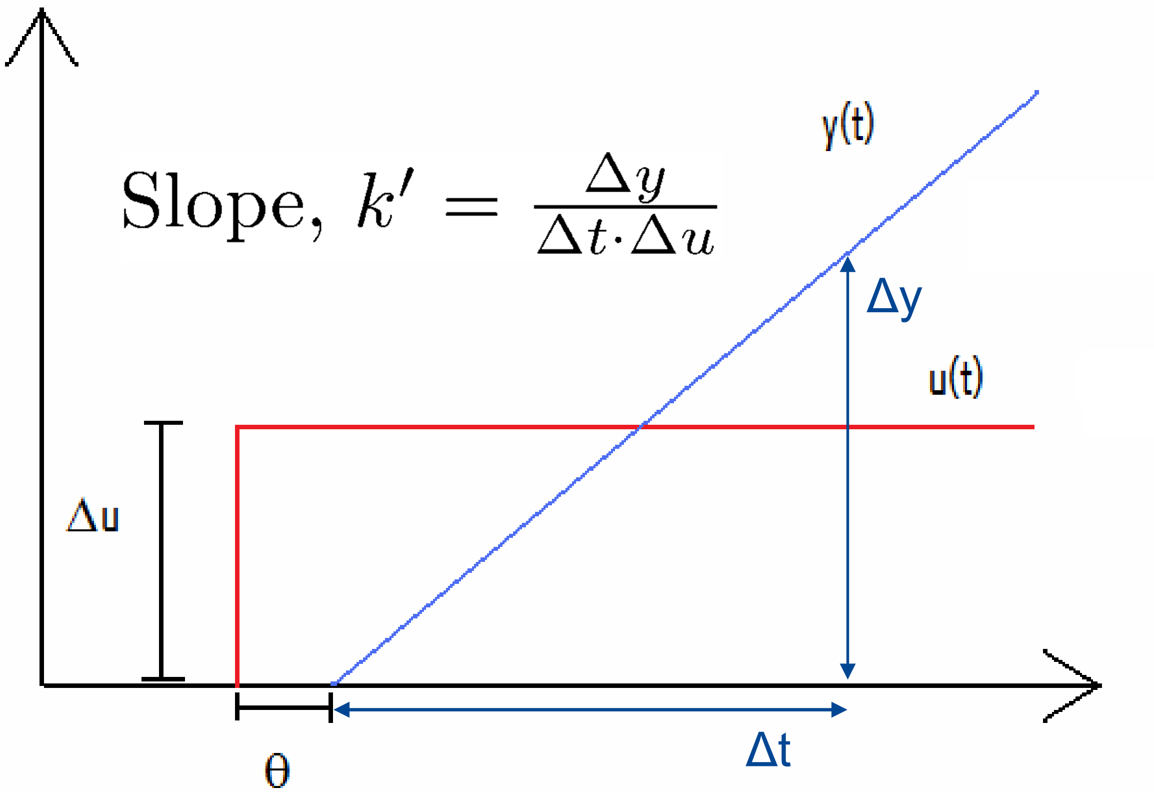
1. Step response experiment

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

1A. Open-loop setting



Step response of integrating process

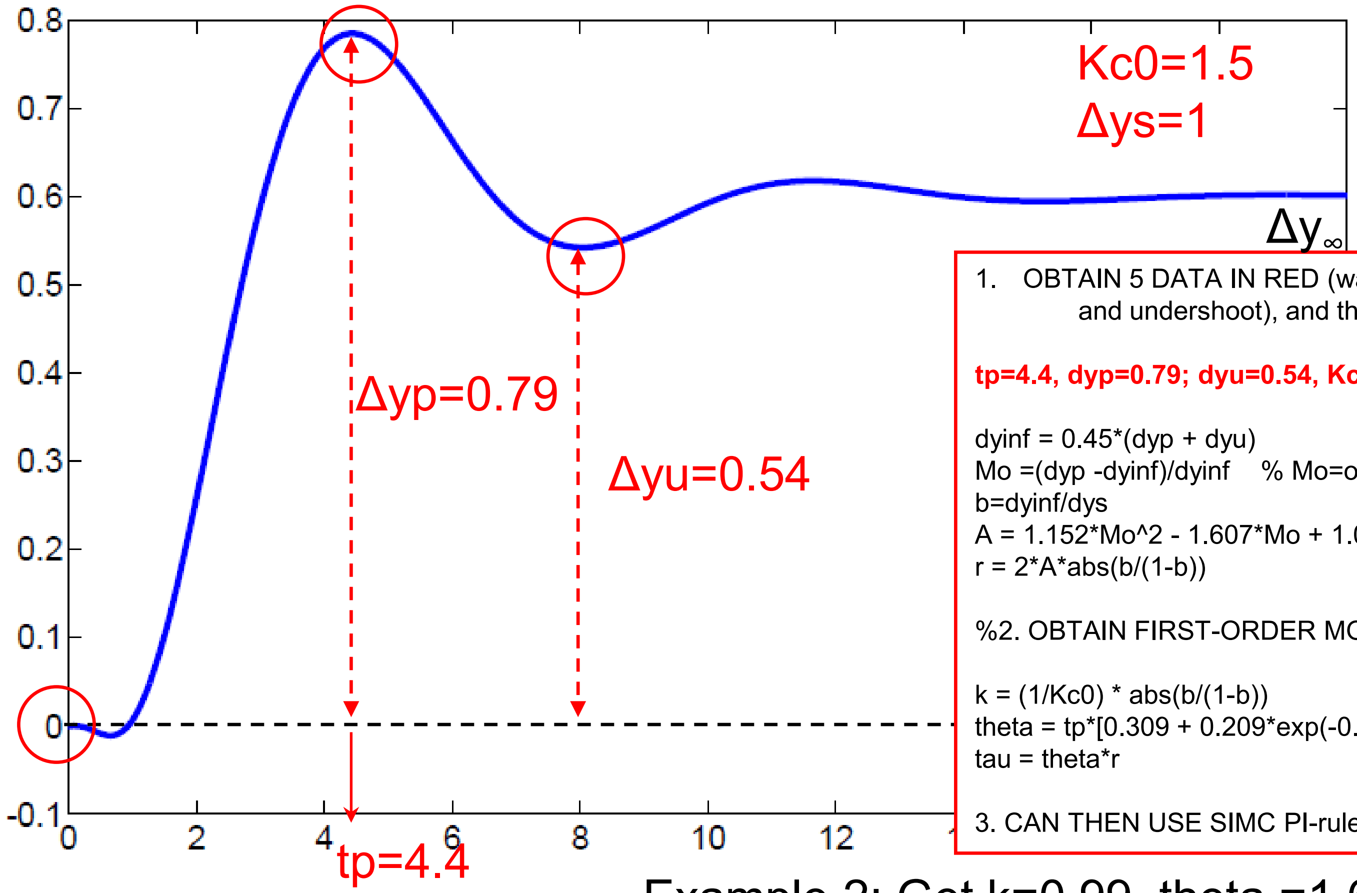


Imagine this as a 1st order with "infinite" τ_1:

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

1B. Closed-loop setpoint response

- Shams' method: P-controller with about 20-40% overshoot



1. OBTAIN 5 DATA IN RED (wait for first overshoot and undershoot), and then read off:

tp=4.4, dyp=0.79; dyu=0.54, Kc0=1.5, dys=1

```
dyinf = 0.45*(dyp + dyu)
Mo =(dyp -dyinf)/dyinf     % Mo=overshoot (about 0.3)
b=dyinf/dys
A = 1.152*Mo^2 - 1.607*Mo + 1.0
r = 2*A*abs(b/(1-b))

%2. OBTAIN FIRST-ORDER MODEL:

k = (1/Kc0) * abs(b/(1-b))
theta = tp*[0.309 + 0.209*exp(-0.61*r)]
tau = theta*r
```

3. CAN THEN USE SIMC PI-rule

Example 2: 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 PID-book 2012,
Sigurd Skogestad and Chriss Grimholt. "The SIMC Method for Smooth PID Controller Tuning" (2012))

2. Model reduction

- Start with complicated stable model on the form

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

- 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

NTNU

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).

Details:

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

In summary, let the original model be in the form

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

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

$$\tau_1 = \tau_{10} + \frac{\tau_{20}}{2}; \qquad \theta = \theta_0 + \frac{\tau_{20}}{2} + \sum_{i\geqslant 3}\tau_{i0} + \sum_j T_{j0}^{\text{inv}} + \frac{h}{2} \tag{10}$$

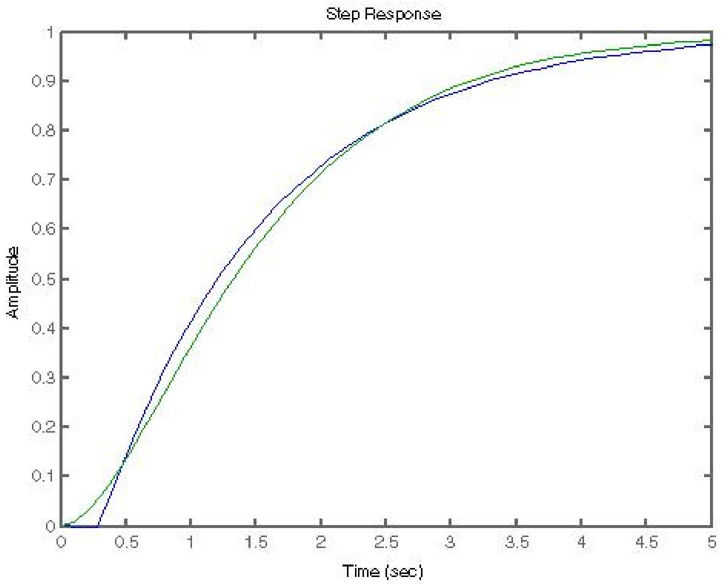
and, to obtain a second-order model (4), we use

$$\tau_1 = \tau_{10}; \qquad \tau_2 = \tau_{20} + \frac{\tau_{30}}{2};$$
$$\theta = \theta_0 + \frac{\tau_{30}}{2} + \sum_{i\geqslant 4}\tau_{i0} + \sum_j T_{j0}^{\text{inv}} + \frac{h}{2} \tag{11}$$

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

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

Example 1



The second-order process

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

is approximated as a first-order with delay process

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

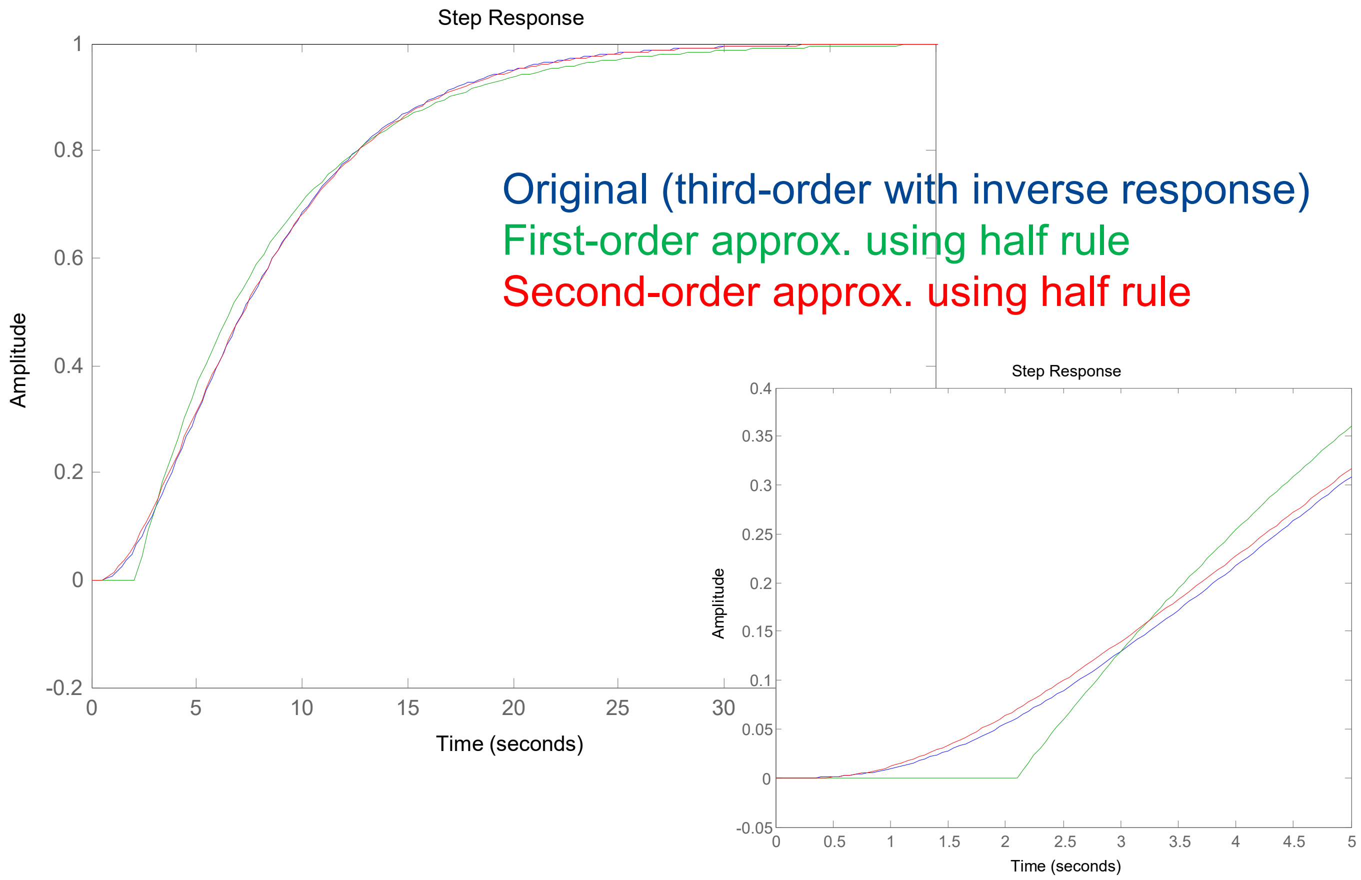
Half rule

with

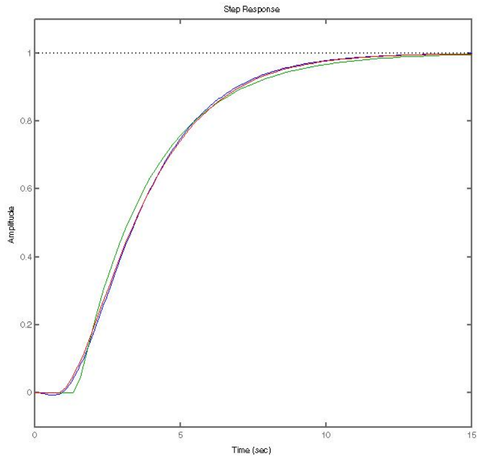
$$k = 1; \quad \tau_1 = 1 + 0.6/2 = 1.3; \quad \theta = 0.6/2 = 0.3;$$

Example 2

```
s=tf('s')
g=(-0.1*s+1)/[(5*s+1)*(3*s+1)*(0.5*s+1)]
g1 = exp(-2.1*s)/(6.5*s+1)
g2 = exp(-0.35*s)/[(5*s+1)*(3.25*s+1)]
step(g,g1,g2)
```



51



Example 3

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

half rule

is approximated as a first-order delay process with

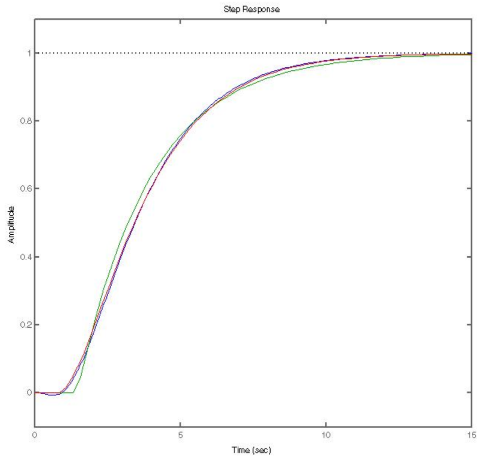
$$\tau_1 = 2 + 1/2 = 2.5$$
$$\theta = 1/2 + 0.4 + 0.2 + 3 \cdot 0.05 + 0.3 - 0.08 = 1.47$$

or as a second-order delay process with

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

Comment: The subtraction of T0=0.08 from the effective delay follows from the approximation (0.08s+1)/(0.2s+1) $\approx \frac{1}{(0.2-0.08)s+1}$ (rule T3). Alternatively, we could have used the approximation (0.08s+1)/(0.05s+1) ≈ 1 (rule T1b) which would reduce the effective delay by 0.05 (instead of 0.08). In any case, it only has a small effect om the effective delay, so it does not matter much for the final result.

MODEL, Approach 2



Example 3

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

half rule

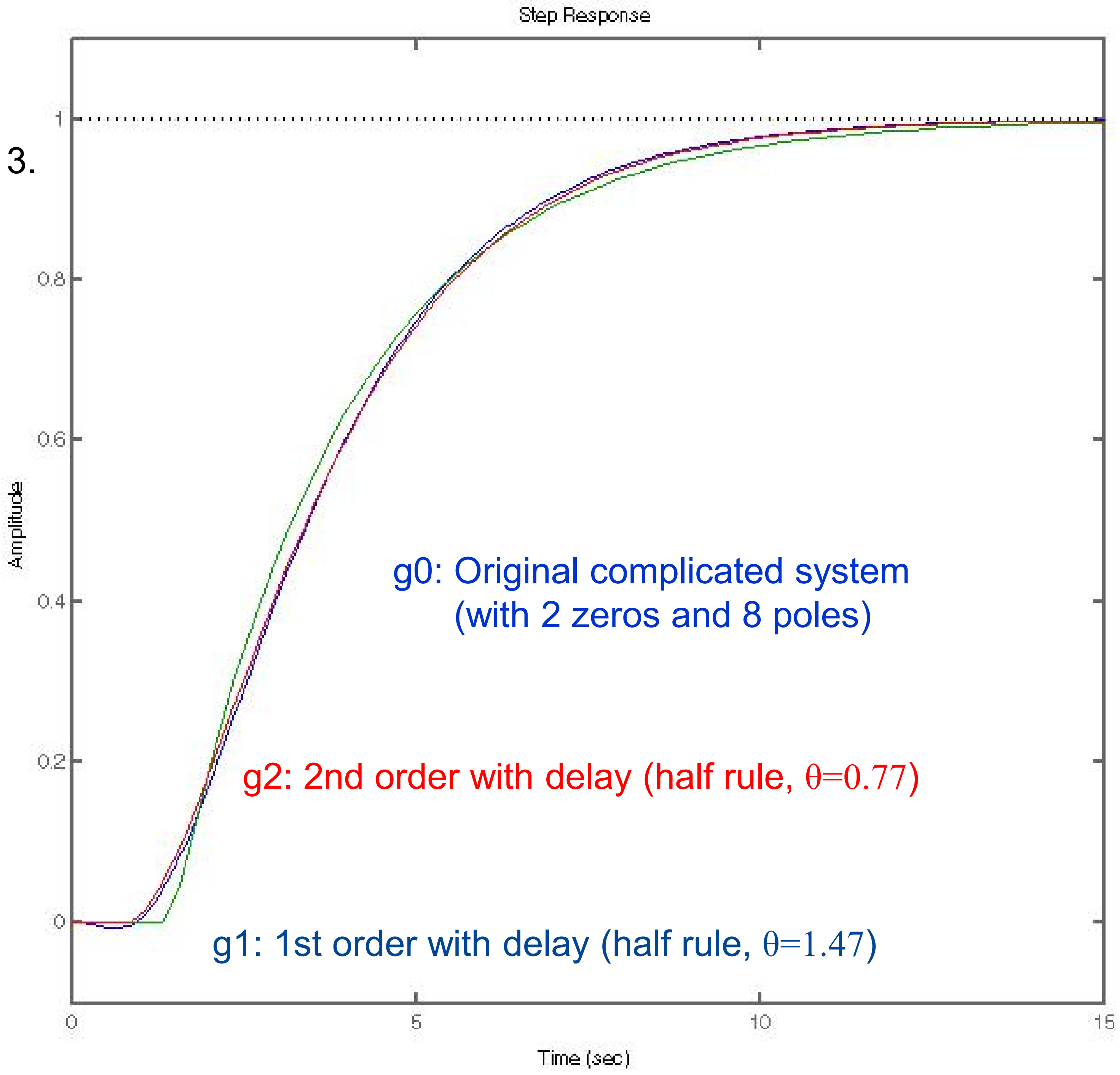
is approximated as a first-order delay process with

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

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 3.


Step Response
Amplitude
Time (sec)
0
0.2
0.4
0.6
0.8
1
0
5
10
15
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)

Example 4. Integrating process

$$g_0(s) = \frac{k'}{s(\tau_{20}s+1)}$$

Half rule gives

$$g(s) = \frac{k'e^{-\theta s}}{s} \text{ with } \theta = \frac{\tau_{20}}{2}$$

Proof:

Note that integrating process corresponds to an infinite time constant

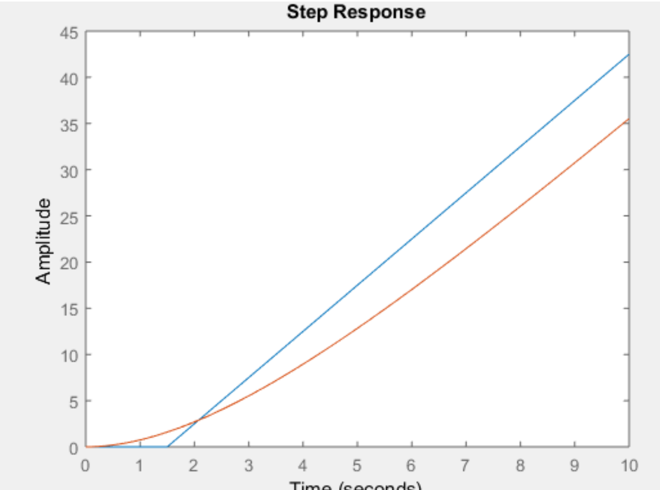
Write

$$g_0(s) = \frac{k'\tau_1}{\tau_1 s(\tau_{20}s+1)} = \frac{k'\tau_1}{(\tau_1 s+1)(\tau_{20}s+1)} \text{ where } \tau_1 \to \infty$$

and then apply half rule as normal, noting that $\tau_1 + \frac{\tau_{20}}{2} \approx \tau_1$:

$$g(s) \approx \frac{k'\tau_1 e^{-\frac{\tau_{20}}{2}s}}{(\tau_1+\frac{\tau_{20}}{2})s} = k'\frac{e^{-\frac{\tau_{20}}{2}s}}{s}$$

Example. g0 = 5/(s*(3*s+1)),
g = 5*exp(-1.5*s)/s,
step(g,g0,10)



Doesn't look so good
But it's OK

55

Approximation of LHP-zeros

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

τ_c = desired closed-loop time constant

We should approximate T_0 by a "close-by" τ_0.

- BUT: The goal is to use the model for control purposes, so we would like to keep (i.e., not approximate) the τ which is closest to the desired τ_c.

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

g0

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

we first introduce from Rule T2 the approximation

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

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

g1

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

or as a second-order time delay model with

g2

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

PID-controller (from 2nd order model) will give performance improvement because $\tau_2 > \theta$

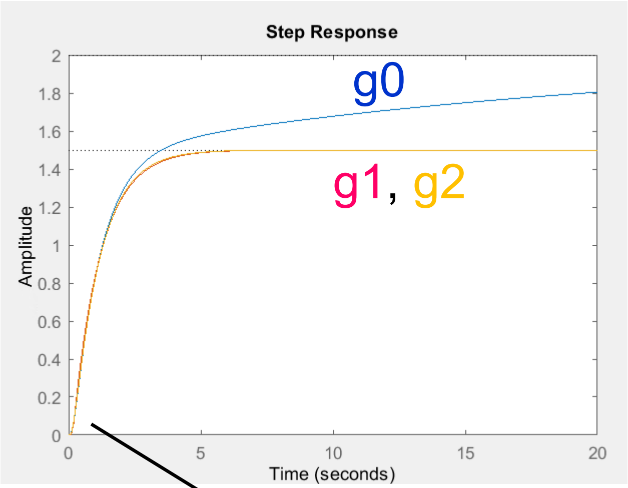
In Example E3, we have two possible values for τ_0, namely 20 and 1. Since T_0=15, it seems clear that we should select the closest τ_0 = 20 and use rule T2.

- But what if T_0=2, maybe selecting τ_0 = 1 is better (and using rule T1)?
- No, this is not clear. Since τ_c is between 0.05 (PID) and 0.15 (PI), we may want to keep τ =1 which is closest to τ_c, that is, also in this case select τ_0 = 20 (and use rule T2)
- This may seem surprising, but it turns out that it will not matter very much in the case for the PI/PID-tunings (try!), because k/tau1 (and thus Kc) will not change much and because tauI = min(tau,4(tauc+theta)).
- Of course, if T_0 gets much closer to 1, then we should select τ_0 = 1.

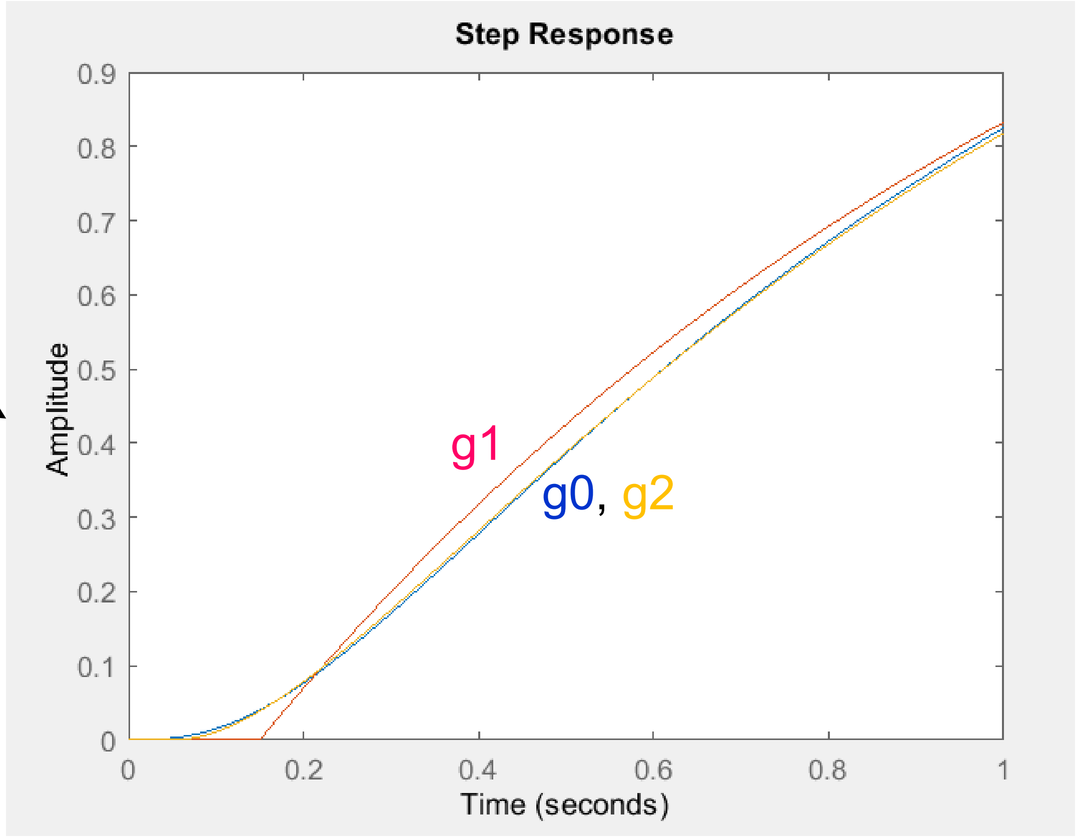
Generally, the LHP-zeros approximation rules results in acceptable (robust) PI/PID-settings, but not necessarily the "optimal" settings.

57

Step response (without control)

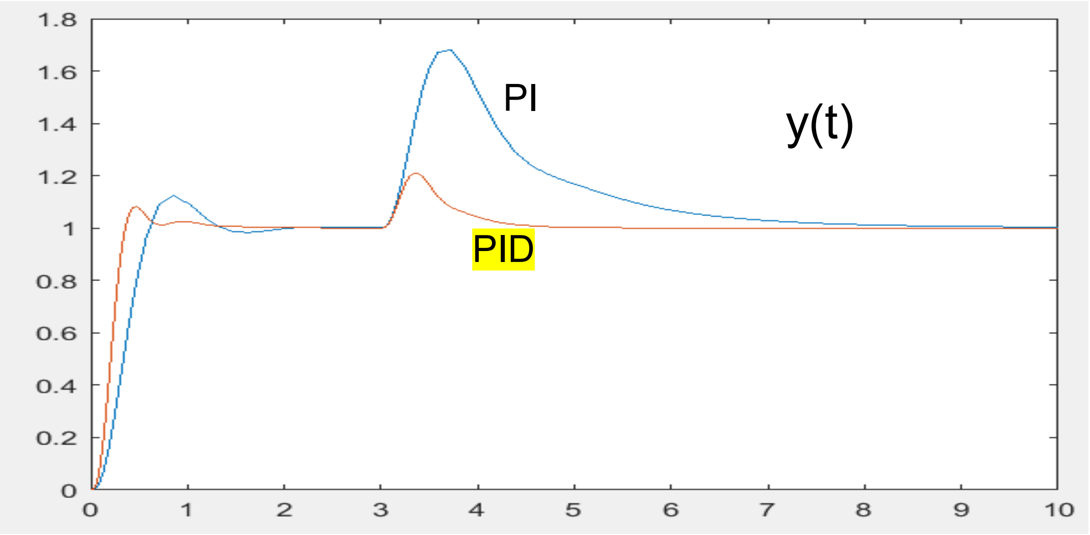


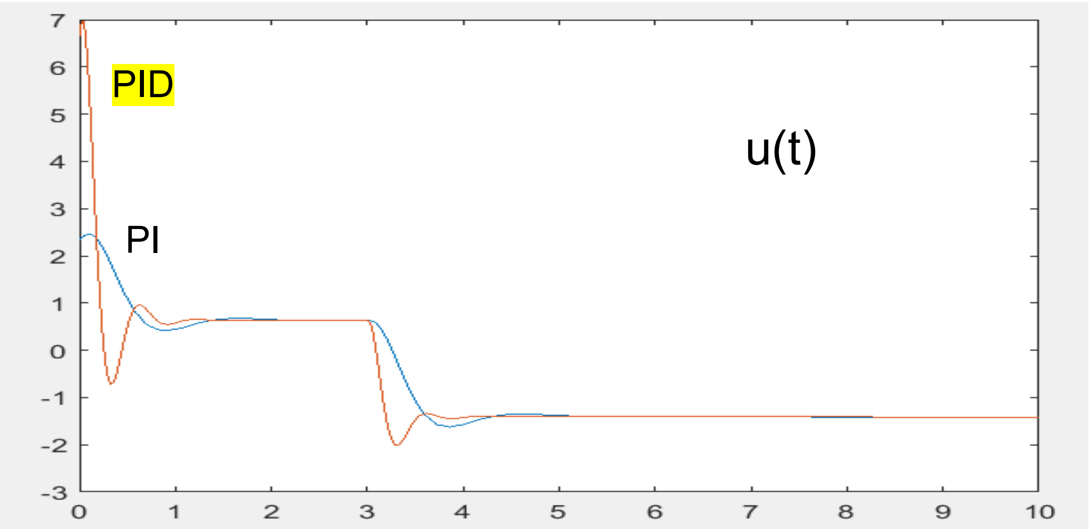
g0 = 2*(15*s+1)/((20*s+1)*(s+1)*(0.1*s+1)^2)
g1=1.5*exp(-0.15*s)/(1.05*s+1)
g2=1.5*exp(-0.05*s)/((s+1)*(0.15*s+1))
step(g0,g1,g2,1)



Note: It's the initial response that matters for feedback control (time from 0 to about 5*tauc)

Simulation with control is as expected: Better performance with controller based on g_2 (PID) than g_1 (PI) but more input usage





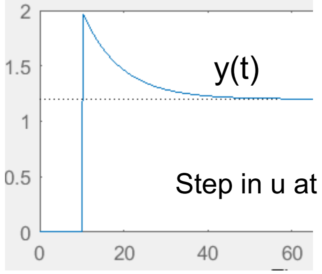
g = 2*(15*s+1)/((20*s+1)*(s+1)*(0.1*s+1)^2)
gd=g

Using g1:
PI. tauc=0.15
Kc=(1/1.5)*1.05/(2*0.15)=2.33
taui=min(1.05,8*tauc)=1.05
taud=0

Using g2
PID. tauc=0.05
Kc=(1/1.5)*1.00/(2*0.05)=6.66
taui=min(1.05,8*tauc)=0.4
taud=0.15, tauf=0.015

Extra slide

Example: Approximation of zero for flow control



– $G_0(s) = 1.2\left(\frac{15s+1}{9\,s+1}\right)$

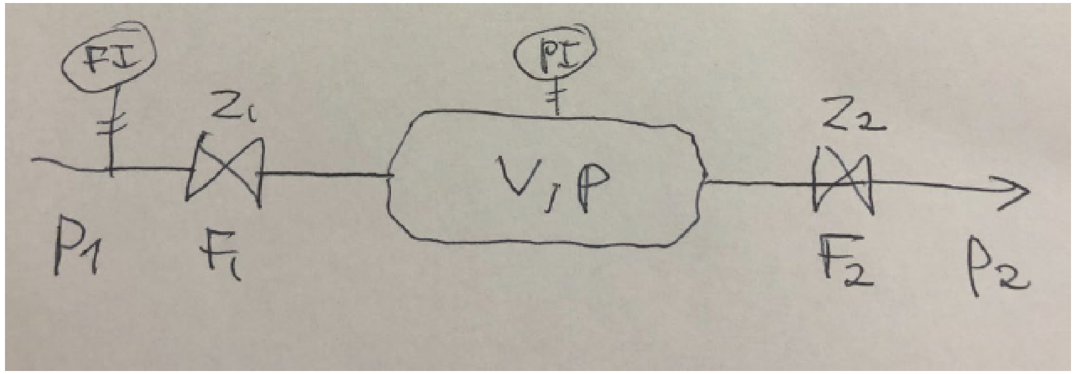
- Note that $T_0 = 15 > \tau_0 = 9$ so we get an overshoot in the step response
- How should we approximate this as a first-order with delay model?
It will depend on the value for tauc. If we apply the LHP-zero approximation rules then we get:
    1. Small tauc (tauc<9): (15s+1)/(9s+1) ≈ 15/9 (Rule T1) ⇒ G(s)= k = 1.2*15/9 = 2
    2. Intermediate tauc (9<taux<15): (15s+1)/(9s+1) ≈ 15/tauc (Rule T1a) ⇒ G(s)= k = 1.2*15/tauc=18/tauc
    3. Large tauc (tauc>15) . (15s+1)/(9s+1) ≈ 1 (Rule T1b) ⇒ G(s)= k = 1.2
- In all three cases we get $G(s) = k$ so we get $\tau_1 = 0$ and in all three cases the SIMC PI-controller becomes a pure I-controller C(s)=KI/s where KI = 1/(k*tauc). Here tauc is free to choose.
- **Flow controller,** The transfer function $G_0(s)$ is typical for a control valve where *u=z=valve position and y=F =flow.*
Consider a typical valve equation $F = Cz\sqrt{p_1 - p_2}$. Following a step change in z, F will immediately jump (to 1.2*15/9=2), but then it will drop down again (to 1.2) because of the reduction in the pressure drop $p_1 - p_2$ which for gases may take some time ($\tau_0 = 9$ in this case). (See Exam 2022, Problem 5 for how to derive G_0)
    – For liquids the dynamics are fast because of small compressibility and can be neglected. Thus, for liquids we always have case 3 (tule T1b). However, the short-term flow overshoot may result in the phenomena of "water hammering".
    – For gases, also cases 1 or 2 may happen if the valve is close to a large gas holdup (large tank or large pipeline).

For a flow controller, a typical value is tauc=10s.

*Some commercial controllers do not allow a pure I-controller. In this case, select taui as some small value (say taui=1s) and use Kc=KI*taui, that is, Kc=(1/k)*(taui/tauc) .*

*However, if the dynamics for changing z or measuring F are slow compared to the desired closed-loop response time tauc, then a better approximatiom of the valve may be G=k/(tau1*s+1). In this case a PI-controller with tauc=tu1 is recommened (SIMC-rule).*

Problem 5 (25%). Modelling and control of flow and pressure



Consider a gas pipeline with two valves. We have measurements of the inflow F_1 and the intermediate pressure p and these should be controlled. The volume of the pipeline can be represented as a tank with volume V as shown in the figure above.

Steady-state data: F_1=1 kg/s, z_1=z_2=0.5, p_1=2 bar, p=1.88 bar, p_2=1.8 bar, V=130 m^3, T=300 K, Parameters: R=8.31 J/K.mol, M_W=18e-3 kg/mol (so the gas is steam).

The following model equations are suggested to describe the system.

(1) $dm/dt = F_1 - F_2$
(2) $m = k_p p$ where $k_p = VM_w/(RT)$
(3) $F_1 = C_1 z_1 \sqrt{p_1 - p}$
(4) $F_2 = C_2 z_2 \sqrt{p - p_2}$

(a) (3%) Explain what the variables and equations represent. What assumptions have been made?
(b) (3%) Determine the parameters in the model (C_1, C_2, k_p). What is the steady-state value of m? What is the residence time of the gas, m/F_1?
(c) (12%) Linearize the model and find the 2x2 transfer function model from z_1 and z_2 (inputs) to F_1 and p (controlled variables). (Note: To simplify, you can assume p_1 and p_2 are constant)
(d) (4%) What pairings do you suggest for single-loop control (with u = [z_1 z_2], y= [F_1 p])? How could control be improved?
(e) (3%) (This can be answered without solving parts a-d). What control structure would you propose if we instead of p want to control the downstream pressure p_2? Thus, we have u=[z_1 z_2] and y = [F_1 p_2] .

Problem 5 (25%)

a) **Model equations and assumptions.**

(1) is the mass balance for the pipeline section [kg/s]

(2) is the ideal gas equation on mass basis with the temperature T is assumed constant.

(3) and (4) are the assumed valve equations. Note that we have assumed a linear valve characteristic.

Variables:

F_1: inlet flow
F_2: outlet flow
z_1: inlet valve opening
z_2: outlet valve opening
C_1: inlet valve constant
C_2: outlet valve constant
m: mass of gas in the pipeline
p: pressure of gas in the pipeline
p_1: pressure of gas at the inlet
p_2: pressure of gas at the outlet
V: volume of pipeline
T: temperature of the system
R: ideal gas constant
M_w: molar mass of gas

b) At steady state, $F_1 = F_2$, and therefore:

$$C_1 = \frac{F_1}{z_1\sqrt{p_1 - p}} = \frac{1}{0.5 \times \sqrt{2 - 1.88}} = 5.773\ kg/s \cdot bar^{1/2}$$

$$C_2 = \frac{F_2}{z_2\sqrt{p - p_2}} = \frac{1}{0.5 \times \sqrt{1.88 - 1.8}} = 7.071\ kg/s \cdot bar^{1/2}$$

$$k_p = \frac{VM_w}{RT} = \frac{130 \times 18 \times 10^{-3}}{8.31 \times 300} \frac{m^3 \times \frac{kg}{mol}}{\frac{J}{mol\,K}\,K} = 9.386 \times 10^{-4}\ kg/Pa = 93.86\ kg/bar$$

$$m = k_p\, p = 93.86 \times 1.88 = 176.457\ kg$$

Residence time: $t_R = m/F_1 = 176.457\ s$

c) Linearizing the model. First linearize the two static valve equations (3) and (4):

$$y_1 = \Delta F_1 = \left(C_1\sqrt{p_1 - p}\right)\Big|_* \Delta z_1 + \left(-\frac{C_1 z_1}{2\sqrt{p_1 - p}}\right)\Big|_* \Delta p = 2\,\Delta z_1 - 4.166\,\Delta p$$

$$\Delta F_2 = \left(C_2\sqrt{p - p_2}\right)\Big|_* \Delta z_2 + \left(\frac{C_1 z_2}{2\sqrt{p - p_2}}\right)\Big|_* \Delta p = 2\,\Delta z_2 + 6.250\,\Delta p$$

From (2) the mass balance (1) becomes $k_p\, dp/dt = F_1 - F_2$ which gives the linearized model for $y_2 = \Delta p$:

$$k_p \frac{d\Delta p}{dt} = \Delta F_1 - \Delta F_2 = 2\,\Delta z_1 - 2\,\Delta z_2 - 10.416\,\Delta p$$

$$\Rightarrow 93.86 \frac{d\Delta p}{dt} + 10.416\,\Delta p = 2\,\Delta z_1 - 2\,\Delta z_2$$

$$\Rightarrow 9.011 \frac{d\Delta p}{dt} + \Delta p = 0.192\,\Delta z_1 - 0.192\,\Delta z_2$$

Applying the Laplace transform to the last expression gives the transfer function for $y_2 = \Delta p$:

$$\Delta p(s) = \frac{0.1925}{9.011\,s + 1}\Delta z_1 - \frac{0.1925}{9.011\,s + 1}\Delta z_2$$

The expression for $y_1 = \Delta F_1$ then becomes

$$\Delta F_1 = \left(2 - 4.166 \times \frac{0.1925}{9.011\,s + 1}\right)\Delta z_1 - 4.166 \times \left(\frac{-0.1925}{9.011\,s + 1}\right)\Delta z_2$$

$$= \left(\frac{2 \times (9.011\,s + 1) - 4.166 \times 0.1925}{9.011\,s + 1}\right)\Delta z_1 + \frac{0.800}{9.011\,s + 1}\Delta z_2$$

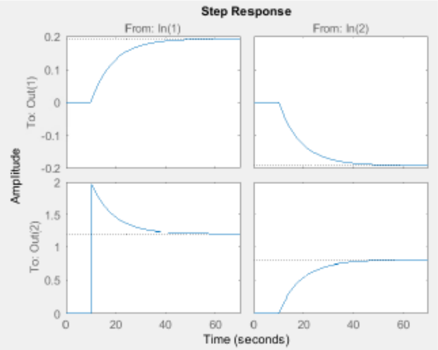
$$= \left(\frac{18.022\,s + 1.200}{9.011\,s + 1}\right)\Delta z_1 + \frac{0.8}{9.011\,s + 1}\Delta z_2 = 1.2\left(\frac{15.018\,s + 1}{9.011\,s + 1}\right)\Delta z_1 + \frac{0.8}{9.011\,s + 1}\Delta z_2$$

Conclusion

$$\begin{bmatrix}\Delta p \\ \Delta F_1\end{bmatrix} = G(s)\begin{bmatrix}\Delta z_1 \\ \Delta z_2\end{bmatrix}, \qquad G(s) = \begin{bmatrix} \frac{0.1925}{9.011\,s + 1} & \frac{-0.1925}{9.011\,s + 1} \\ 1.2\left(\frac{15.018\,s + 1}{9.011\,s + 1}\right) & \frac{0.8}{9.011\,s + 1} \end{bmatrix}$$

Note that the time constant of 9s is much smaller than the residence time of 176s. This is typical for gas systems. Also note that u1=z1 has a direct effect on y2=F1(as expected from physics:; see also element g21 in step response below which has an overshoot because of the zero).

```
s=tf('s')
g11=0.1925/(9*s+1); g12=-g11;
g21=1.2*(15*s+1)/(9*s+1);
g22=0.8/(9*s+1);
G=[g11 g12; g21 g22];
step(G*exp(-10*s)) % To make plot
clearer I put in a delay so that step is
at t=10
```

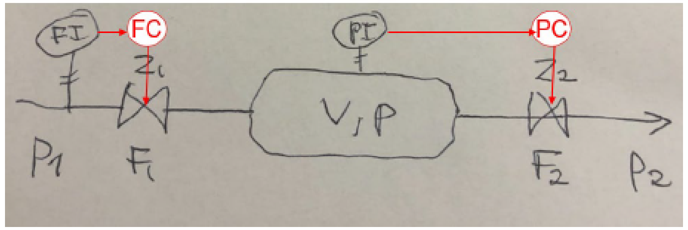


d)

Steady-state gain matrix: $G(0) = \begin{bmatrix}0.1925 & -0.1925 \\ 1.2 & 0.8\end{bmatrix} \rightarrow$

steady-state RGA matrix: $\Lambda = \begin{bmatrix}\lambda & 1-\lambda \\ 1-\lambda & \lambda\end{bmatrix}$ $where\ \lambda = \frac{0.1925 \times 0.8}{0.1925 \times 0.8 + 0.1925 \times 1.2} = 0.4.$

From the steady-state RGA, the recommended pairing is then the off-diagonal pairing, that is, $F_1 - z_1$ and $p - z_2$. This happens to coincide with the intuitive pairing ("pair-close rule") since z1 has a direct effect on F1. It also agrees with what we get from the RGA if we consider the initial response (high frequency).



However, high steady-state interaction is to be expected, since Λ is far from the ideal case (identity matrix). Possible solutions are the implementation of a decoupler (probably steady-state decoupler is OK), or separating the timescales of the two loops.

Since the flow control has a direct effect from z1 to F1, this should probably be the fast loop, and then the pressure loop can be about 5 times slower. But if both loops should be equally fast, a decoupler is preferred.

What about the tuning of the flow loop? What model should we use? We have that

$$G_0(s) = 1.2\left(\frac{15s + 1}{9\,s + 1}\right)$$

Note that T0=15 > tau0=9. How should we approximate this as a first-order with delay model? It will depend on the value for tauc. If we apply the LHP-zero approximation rules then we get.

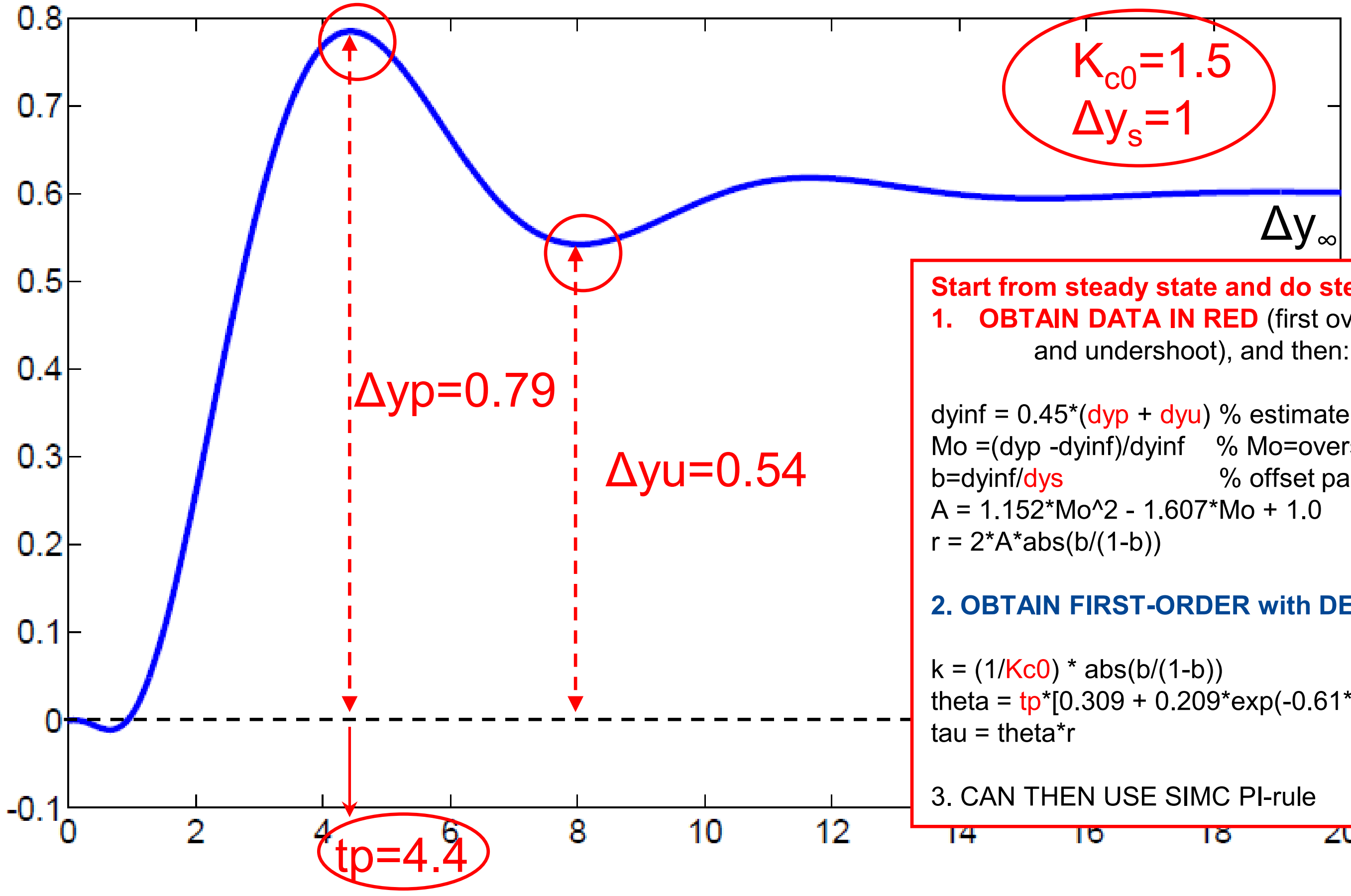
Small tauc (tauc<9): (15s+1)/(9s+1) ≈ 15/9 (Rule T1) ⇒ G(s)=1.2*15/9 = 2
Intermediate tauc (9<taux<15): (15s+1)/(9s+1) ≈ 15/tauc (Rule T1a) ⇒ G(s)=18/tauc
Large tauc (tauc>15) . (15s+1)/(9s+1) ≈ 1 (Rule T1b) ⇒ G(s)=1.2

In all these three case the SIMC PI-controller becomes a pure I-controller C(s)=KI/s with KI = 1/(k*tauc). Note that for the intermediate tauc we get KI=1/18 (independent of Kc).

e) This is a trick question, because it will not work. This control strategy would not be consistent, as we can see that that is does not follow the radiation rule. In general, the control of pressures that are external to the process is equivalent to a flow specification (TPM), which in this case would conflict with the specification of F_1.

Alternative to Ziegler-Nichols closed-loop experiment: Obtains more information and avoids cycling.

Shams' method: Closed-loop setpoint response with P-controller with about 20-40% overshoot



Start from steady state and do step P-response

1. **OBTAIN DATA IN RED** (first overshoot and undershoot), and then:

```
dinf = 0.45*(dyp + dyu) % estimate dyinf (so don't wait)
Mo =(dyp -dyinf)/dyinf    % Mo=overshoot (about 0.3)
b=dyinf/dys               % offset parameter
A = 1.152*Mo^2 - 1.607*Mo + 1.0
r = 2*A*abs(b/(1-b))
```

2. OBTAIN FIRST-ORDER with DELAY MODEL:

```
k = (1/Kc0) * abs(b/(1-b))
theta = tp*[0.309 + 0.209*exp(-0.61*r)]
tau = theta*r
```

3. CAN THEN USE SIMC PI-rule

Example 2: Get k=0.99, theta =1.67, tau=3.0

Ref: Shamssuzzoha and Skogestad (JPC, 2010) + modification by C. Grimholt (PID-book 2012)

Example E2 (Further continued) We want to derive PI- and PID-settings for the process

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

using the SIMC tuning rules with the "default" recommendation $\tau_c = \theta$. From the closed-loop setpoint response, we obtained in a previous example a first-order model with parameters $k = 0.994, \theta = 1.67, \tau_1 = 3.00$ (5.10). The resulting SIMC PI-settings with $\tau_c = \theta = 1.67$ are

$$\text{PI}_{cl}: \quad K_c = 0.904, \qquad \tau_I = 3.$$

From the full-order model $g_0(s)$ and the half rule, we obtained in a previous example a first-order model with parameters $k = 1, \theta = 1.47, \tau_1 = 2.5$. The resulting SIMC PI-settings with $\tau_c = \theta = 1.47$ are

$$\text{PI}_{\text{half-rule}}: \quad K_c = 0.850, \qquad \tau_I = 2.5.$$

From the full-order model $g_0(s)$ and the half rule, we obtained a second-order model with parameters $k = 1, \theta = 0.77, \tau_1 = 2, \tau_2 = 1.2$. The resulting SIMC PID-settings with $\tau_c = \theta = 0.77$ are

$$\text{Series PID}: \quad K_c = 1.299, \qquad \tau_I = 2, \qquad \tau_D = 1.2.$$

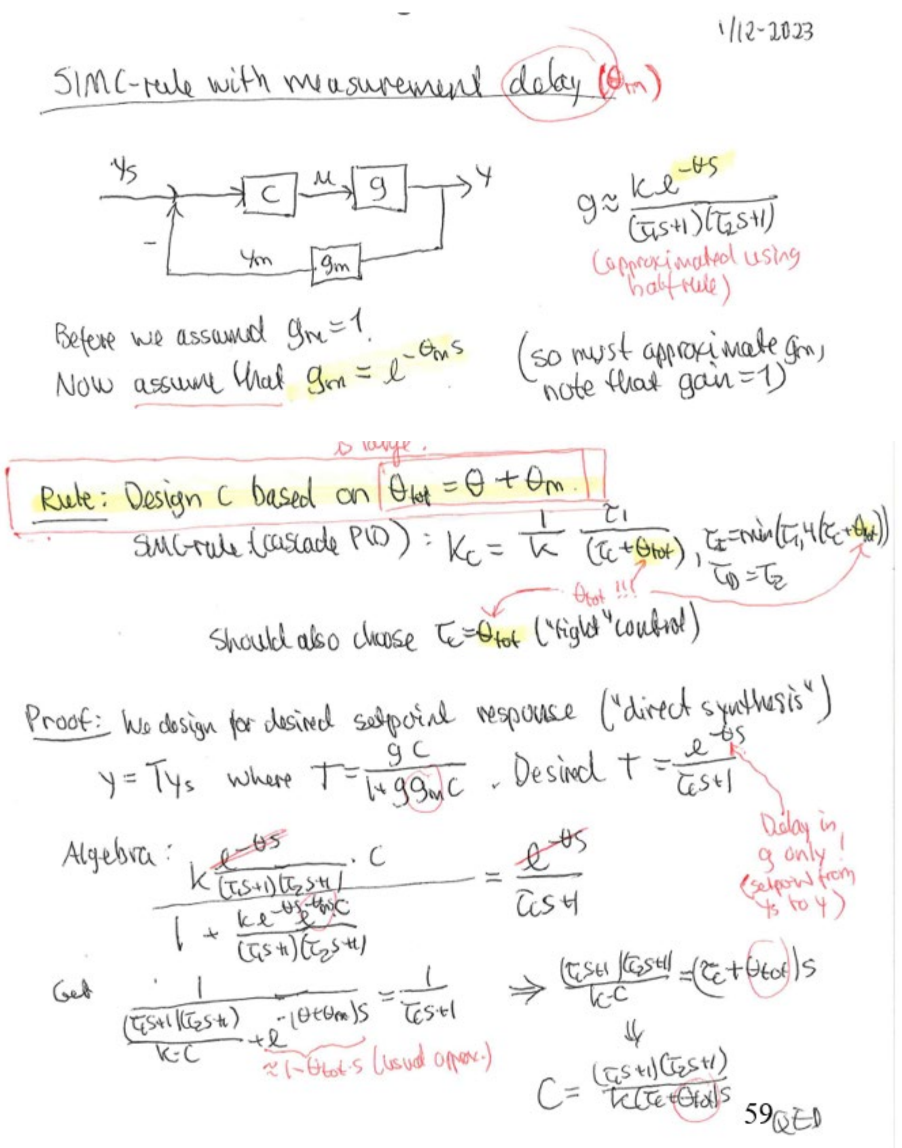
The corresponding settings with the more common ideal (parallel form) PID controller are obtained by computing $f = 1 + \tau_D/\tau_I = 1.60$, and we have

$$\text{Ideal PID}: \quad K_c' = K_c f = 1.69, \qquad \tau_I' = \tau_I f = 3.2, \qquad \tau_D' = \tau_D/f = 0.75. \tag{5.30}$$

SIMC-rule with measurement dynamics

This is simple: Combine the measurement dynamics $g_m(s)$ and the process model g(s) and apply the SIMC-rules on gg_m. This applies both to the model approximation (half rule) to get a 1st or 2nd model and to the PI- or PID-tuning, including the choice of τ_c.

See also the handwritten note for a «proof», for example, that the total delay also includes the delay in the measurement g_m.



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 initial part of step response

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

For cascade-form PID controller:

$$K_c = \frac{1}{k'} \cdot \frac{1}{(\theta+\tau_c)}$$
$$\tau_I = \min(\tau_1, 4(\tau_c+\theta))$$
$$\tau_D = \tau_2$$

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

Note: The delay θ includes any measurement delay

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"