Extensions of Skogestad's SIMC tuning rules to oscillatory and unstable processes

Henrik Manum

December 19, 2005

Abstract

The aim of this report is to derive PID tuning-rules for processes with either a pair of complex poles in the left-half-plane or a single real pole in the right-half-plane. For the case of stable processes with complex poles and a first order filter, a simple model-reduction scheme has been proposed, that enables use of the existing SIMC-rules [6] to get PID-controller settings. For pure oscillatory stable processes and the unstable case with a singe real pole a review of previous work by professor Skogestad, NTNU, has been conducted. A method for perfomance-comparison of the resulting PID-controller with a IAE- and TV-optimal controller has been propsed. There is still work to be done on this topic.

KEYWORDS: PID SIMC UNSTABLE OSCILLATORY OPTIMAL

Contents

Chapter 1

Introduction

Skogestad [6] has published a very sucsessful set of tuning rules for PID controllers that work well for both reference tracking and load disturbances. Those analytically derived settings have the nice advantage of being simple and easy to memorize and work well on a broad range of processes. Basically, the tuning is done in two steps:

Step 1. Reduce to model at hand to a first or second-order plus delay model.

Step 2. Derive model-based controller settings

Although very practical and effective, the rules of Skogestad [6] do not directly tackle the problem of processes with complex poles and/or zeros and right-half plane (RHP) poles. In step 1, reduce the model at hand to a first or second-order plus delay model, one may use the half-rule in Skogestad [6]. In this project we aim to either reduce the model at to a first or second-orded plus delay model, so we can directly use the SIMC-settings of the reduced model, or we will derive new tuning rules directely. The new issue in model reduction will be to approximate a model with complex poles to a first or second order process plus delay with real poles[1].

1.1 Review of the SIMC PID settings

Assume we have a model on the form

$$g(s) = \frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)} e^{-\theta s} \tag{1.1}$$

or

$$g(s) = \frac{k}{(\tau_1 s + 1)} e^{-\theta s} \tag{1.2}$$

[1]This may seem a bit strange, but remeber that we have the delay as a degree of freedom in the approximation.

The SIMC PID controller settings are then [6, page 93] for the process model in (1.1)

$$K_c = \frac{1}{k}\frac{\tau_1}{\tau_c + \theta} \tag{1.3}$$

$$\tau_I = \min\{\tau_1, 4(\tau_c + \theta)\} \tag{1.4}$$

$$\tau_D = \tau_2 \tag{1.5}$$

For a process model on the form in equation (1.2) we get tuning rules by setting $\tau_D = 0$, and we get a PI controller. Note that to get a PID controller we start with a model on the form (1.1), whereas for a PI controller, we start with a model on the form (1.2). We note that the only tuning parameter is τ_c. A recommended value for τ_c for fast response and good robustness is $\tau_c = \theta$ [6].

The settings for a PID controller are given for a cascade controller:

$$c(s) = K_c\left(\frac{\tau_I s + 1}{\tau_I s}\right)(\tau_D s + 1) \tag{1.6}$$

1.2 A note on simulation tools used in the report

All simulations and optimization problems in this report were solved using Matlab and Simulink. The number of scripts and functions grew rather large, and as to reduce the length of the report I choose not to include all the files. However, I did include two files, that are shown in appendix D. I choose to include these two files because I think maybe future student might like to use them for their work. The main Simulink-models are attached in A.

Chapter 2

Stable processes with complex poles

2.1 Introduction

This part of the report will cover tuning processes of the kind

$$g(s) = \frac{1}{(\tau_1 s + 1)(\tau_0^2 s^2 + 2\zeta\tau_0 s + 1)} e^{-\theta s} \tag{2.1}$$

with PID controllers. Underdamped systems is the main focus. Ogata [3] shows that we will only get closed loop resonace if $\zeta < 0.707$[1], so only damping coefficients below this value will be considered.

As a first approach we will divide these processes into three categories as shown in table 2.1.

Category	Description
`Category A`	$\tau_0 >> \tau_1$. When this is the case we will observe a distinctive resonant peak in frequency domain. There processes may be threated as "pure" 2nd order underdamped systems, where we might include the neglected time-constant τ_1 in τ_0 and the delay θ_{eff}, as in the half-rule derived by Skogestad [6].
`Category B`	$\tau_0 < \tau_1$, but still a clear peak in the frequency domain, that is, the peak gain is the maximum gain for the process.
`Category C`	$\tau_0 < \tau_1$, where the resonant peak is less than the low-frequency gain for the process.

Table 2.1: Categories of systems

Figures 2.1 and 2.2 shows some typical examples of these kinds of processes. For the time-responses in figure 2.1 a unit-step in the input was conducted at $t = 0$. The letters indicate what kind of categories the different processes belong in.

,

[1]See also for example Seborg et al. [4, page 121] to understand why there is no peak when $\zeta > 0.707$

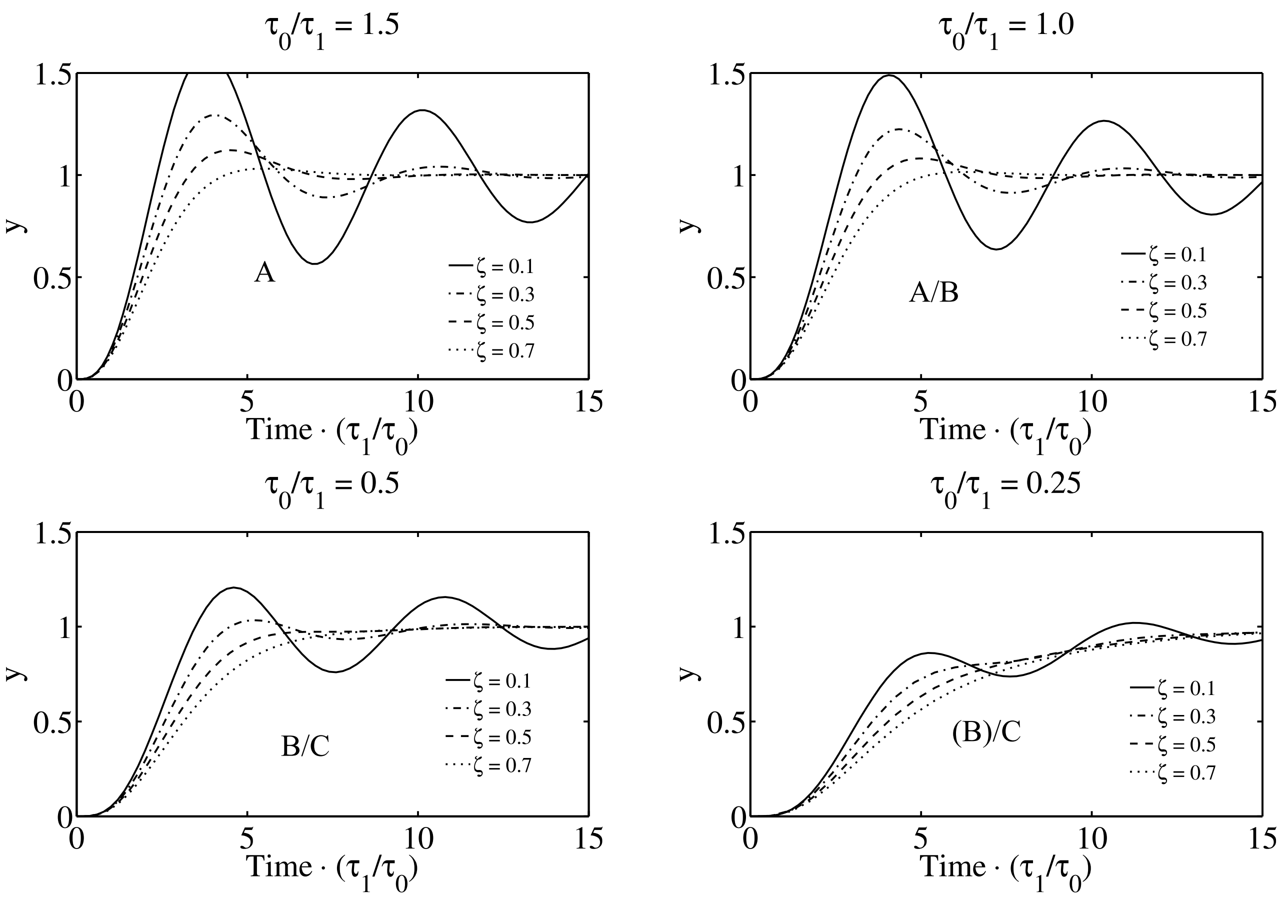

Figure 2.1: Time responses

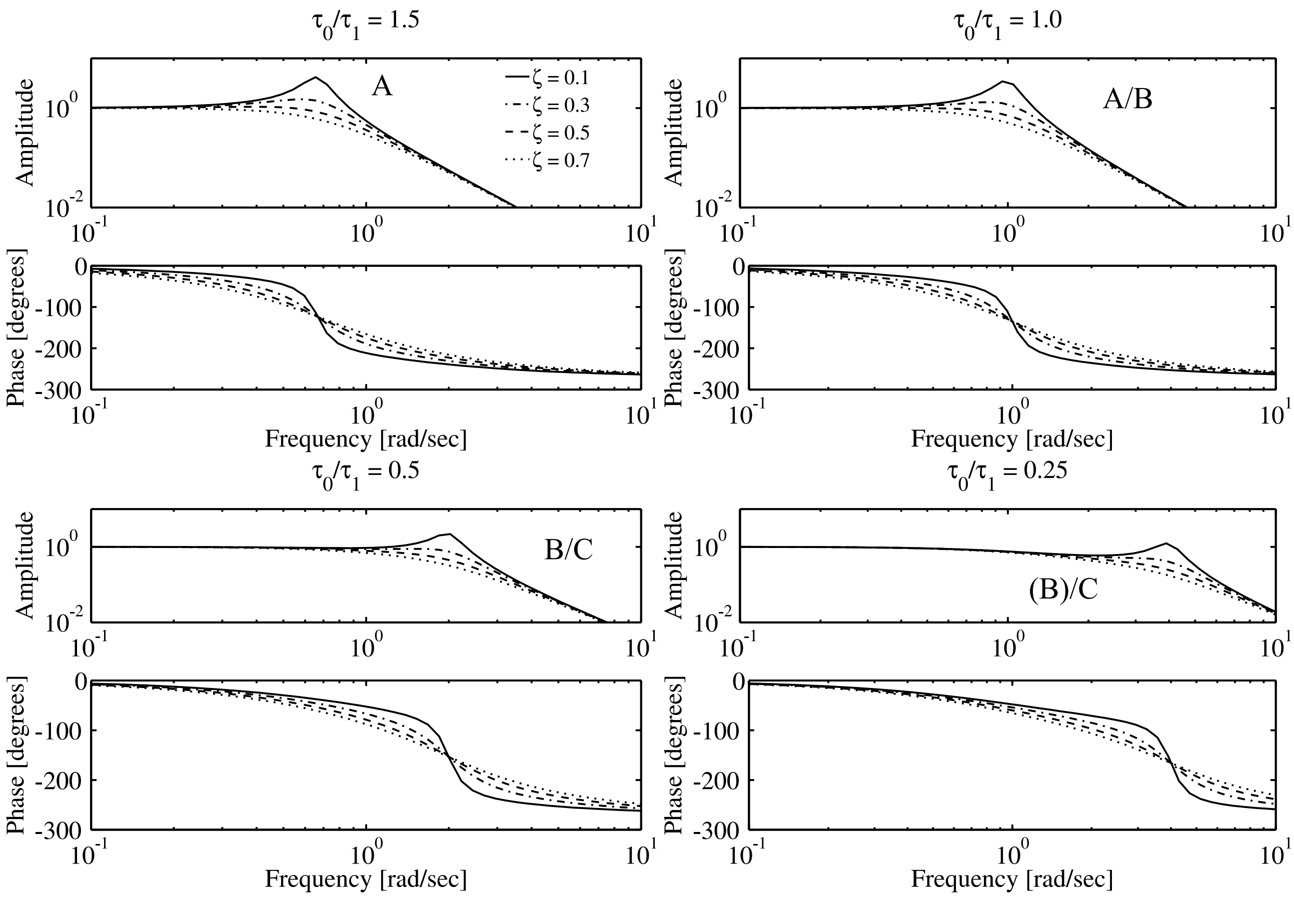

Figure 2.2: Frequency responses

2.2 Category B

2.2.1 Introduction

We have a process on the form

$$g(s) = \frac{k}{(\tau_1 s + 1)\left(\tau_0^2 s^2 + 2\tau_0 \zeta s + 1\right)} e^{-\theta s} \tag{2.2}$$

which we want to control with a PID controller. Ideally we would like to reduce this model to a first or second order plus delay model (see equations (1.2) and (1.1)), and thereafter use the SIMC-rules to derive PID controller settings.

2.2.2 Gain

From Seborg et al. [4, page 353] we have that the maximum amplitude ratio for a underdamped second order transfer is $(\mathrm{AR})_{\max} = 1/(2\zeta\sqrt{1-\zeta^2})$. AR is the amplitude ratio, defined as the ratio of output to input amplitude. We get a pure second-order underdamped process by setting $\tau_1 = 0$ in equation (2.2). In this part we assume that $\tau_1 > \tau_0$, so the dominating dynamics is the first order part. The dominating dynamic part will filter out the underdamped part and thereby reduce the oscillating behaviour of the process, so one may say that this is an filtered underdamped process. Remember that this category is the set of processes that has this filer-effect and also has a peak-gain at some frequency that is larger than the low-frequency (steady-state) process gain.

The first order part of the process model (equation (2.2)) will reduce the usual peak gain (for a pure second-order process with $\tau_1 = 0$) approximately linearly in the log-log space (See figure 2.3) if $\tau_0 \le \tau_1$. As we see from the figure the filter-effect ($\tau_1 \ge \tau_0$) will reduce the $\log_{10} k$ by $\log_{10}(1/\tau_0) - \log_{10}(1/\tau_1) = \log_{10}(\tau_1/\tau_0)$. This is true for asymptotic regions, i.e. when τ_0 and τ_1 is different from each other.

This means that the amplitude at the resonant peak is approximately:

$$\log_{10} \frac{k_{\max}}{k} = \begin{cases} \log_{10}\left(\frac{1}{2\zeta\sqrt{1-\zeta^2}}\right) - \log_{10}\left(\frac{\tau_1}{\tau_0}\right) & \tau_0 \le \tau_1 \\ \log_{10}\left(\frac{1}{2\zeta\sqrt{1-\zeta^2}}\right) & \tau_0 > \tau_1 \end{cases} \tag{2.3}$$

...or equivalently

$$\frac{k_{\max}}{k} = \begin{cases} \left(\frac{(\tau_0/\tau_1)}{2\zeta\sqrt{1-\zeta^2}}\right) & \tau_0 \le \tau_1 \\ \left(\frac{1}{2\zeta\sqrt{1-\zeta^2}}\right) & \tau_0 > \tau_1 \end{cases} \tag{2.4}$$

This gain-rule is always safe, but it is not good, as the gain is higher than the process gain at all frequencies.

Note that this gain should never be set to a value below 1. (Then we are no longer in this set of processes. See section 2.3).

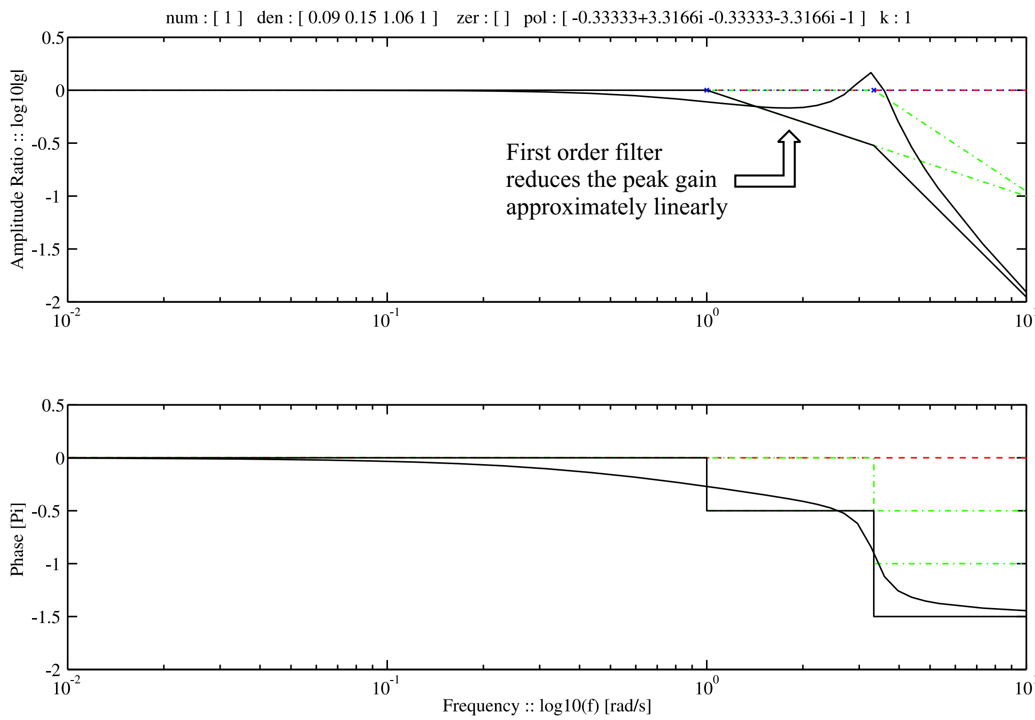

Figure 2.3: Approximate bode-plot for the function $g(s) = \frac{1}{(\tau_1 s+1)\left(\tau_0^2 s^2 + 2\tau_0\zeta s+1\right)}$, $\tau_1 > \tau_0$. $(\tau_1, \tau_0/\tau_1, \zeta) = (1, 0.3, 0.1)$

2.2.3 Phase, $\tau_0 \leq \tau_1$

The gain rule from section 2.2.2 is rather conservative for all frequencies, since it chooses the maximum gain for all frequencies. In a mathematical sense we may say that the gain-rule tries to find a gain $k_{\text{gain rule}} = \max_\omega |g(j\omega)|$. We want to make a model in order to afterwards use the SIMC-rules for tuning, so it is important that the model is as good as possible. The gain rule is of course safe for all frequencies, since it oversetimates the process gain at all frequencies. Since we want a fast and robust response for the resulting controller after applying the SIMC-rules, we should put some emphasis on getting a good model for the phase.

As a simple approach we could try to approximate the phase of the transferfunction (equation (2.2)) by a delay $e^{-\theta_{\text{approx}} s}$. The delay will only affect the phase, so however we choose this delay, our model for the gain will remain unchanged. We want to find θ_{approx} as a function of lag (τ_1, τ_0) and damping coefficient ζ.

Skogestad [6, pages 113-114] argues in his derivation of the half-rule that the important frequencies are below the bandwidth-frequency of the controller, and that this bandwidth is given by $\omega_{\text{bandwidth}} \approx 1/\theta_{\text{effective}}$. After some trial and error, we found that a possible approximation for the phase is given by equation (2.5).

$$\theta_{\text{approx}}(\zeta, \tau_1, (\tau_0/\tau_1)) = (1.5 + \zeta \cdot 5) \cdot (\tau_0/\tau_1) \cdot \tau_1 \cdot f \tag{2.5}$$

$$f = 0.6^{(\tau_0/\tau_1)^2} \tag{2.6}$$

To see why (2.5) is a possible approximation for the phase of the process, consider figures 2.4 and 2.5. By looking at (2.5) we observe that this phase-corretion consists of two parts, the term $(1.5 + \zeta \cdot 5)$ and

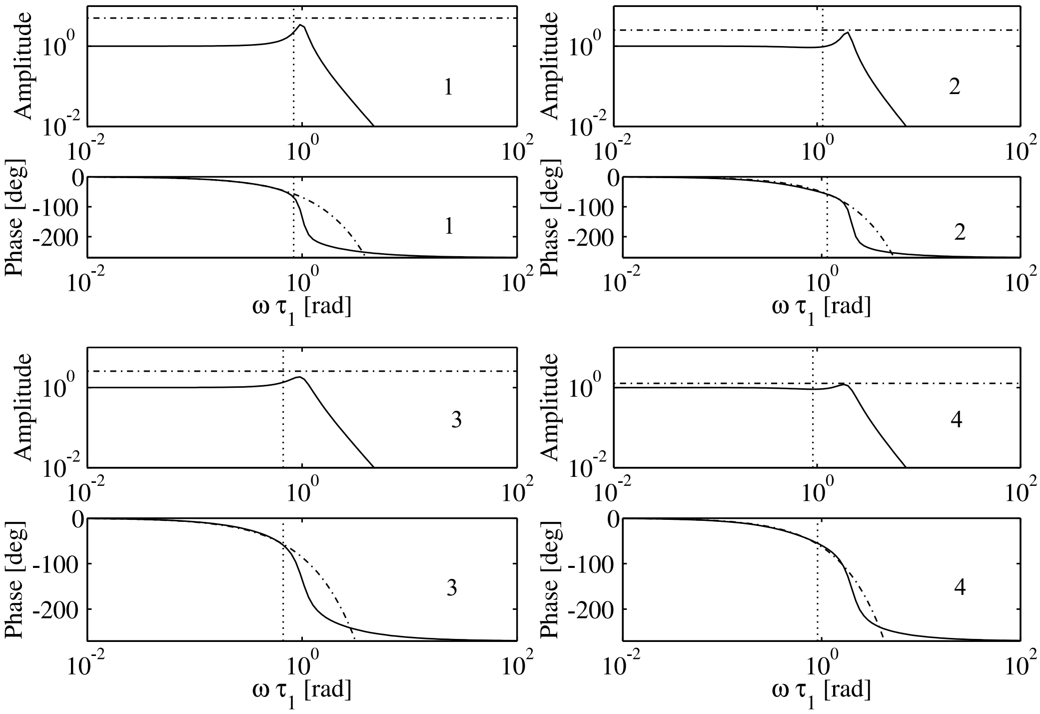

Figure 2.4: Comparing transferfunctions with the approximation given above, $g = k_{\max} e^{-\theta_{\text{eff}} s}$. 1: $(\zeta, \tau_0/\tau_1) = (0.1, 1)$, 2: $(\zeta, \tau_0/\tau_1) = (0.1, 0.5)$, 3: $(\zeta, \tau_0/\tau_1) = (0.2, 1)$, 4: $(\zeta, \tau_0/\tau_1) = (0.2, 0.5)$. The dotted vertical line is $1/\theta_{\text{eff}} \approx \omega_{\text{bandwidth}}$. The process (full line) is $g = \frac{k}{(\tau_1 s+1)(\tau_0^2 s^2+\tau_0\zeta s+1)}$. $k = 1$.

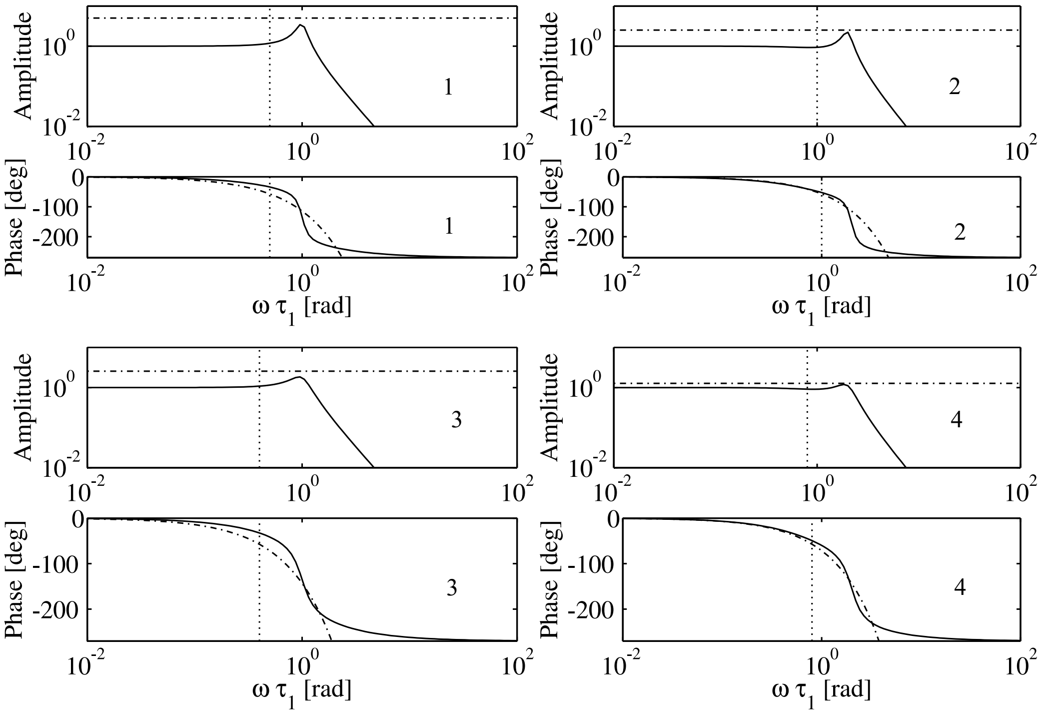

Figure 2.5: Comparing transferfunctions with the approximation given above, $g = k_{\max} e^{-\theta_{\text{eff}} s}$, but with $f = 1$. 1: $(\zeta, \tau_0/\tau_1) = (0.1, 1)$, 2: $(\zeta, \tau_0/\tau_1) = (0.1, 0.5)$, 3: $(\zeta, \tau_0/\tau_1) = (0.2, 1)$, 4: $(\zeta, \tau_0/\tau_1) = (0.2, 0.5)$. The dotted vertical line is $1/\theta_{\text{eff}} \approx \omega_{\text{bandwidth}}$. The process (full line) is $g = \frac{k}{(\tau_1 s+1)(\tau_0^2 s^2+\tau_0 \zeta s+1)}$. $k = 1$.

the term $f = 0.6^{(\tau_0/\tau_1)^2}$. The first term is meant to be a correction of the phase as a function of ζ when $\tau_0 >> \tau_1$. Then the phase-curve will have a certain form, and this term is meant to describe the phase when we have this situation. As τ_0 approches τ_1 (from above), the phase will drop quicker, as we get contributions from both dynamic modes at the same time[2]. Then we need a second correction, and the f-part is meant to take care of this.

By inspection of figures 2.4 and 2.5 we see that the f-correction gives a really good phase approximation, whereas setting $f = 1$ generally gives a phase-approximation with more phase-lag than the actual process. Of course it is more safe to assume that the approximated model has more phase-lag than the original process, since this will give more conservative controller settings in the resulting controller. (See equations (1.3) - (1.5), page 2). On the other hand, since our gain-approximation is (too) conservative at all frequencies except at the peak-frequency, it may be better to have a more optimistic approximation for the phase, and thereafter assume that the conservative gain and optimistic phase approximations cancel each other out, to give an overall good controller.

Let us now discuss the bandwidth assumption. Look at process 1 in figure 2.4, i.e. the upper-left process. The phase-approxiamtion is here good up to the assumed bandwidth frequency, but above the bandwidth it is not good and very unsafe, as the real process had significantly more phase-lag than our approximation. Fortunately the gain-approximation is very conservative in this region, but considering that we may have modelling-errors in the orginial model this may be a bit too optimistic approximation. However, this is a "limiting process", as $\tau_1 = \tau_0$ in this case, and we observe that the approxiamtion is better for processes with τ_1 strictly less than τ_0.

Another issuse worth mentioning is that we have assumed that we only need to have a model of the *phase* below the bandwidth, whereas we have tried to capture the *maximum gain* without considering the resulting bandwidth, and we observe that generally the peak actually lies outside the bandwidth. Since the processes we are looking at are generally more or less oscillatory, and we want to control and maybe remove / damp out some of the oscillations, it is most likely necessary to inculde the peak in the approximated model, even though it is outside the bandwidth.

2.2.4 Direct synthesis of a PID controller

The gain and phase rules from above approximates the process model from equation (2.2) by $g \approx k_{\max} e^{-\theta_{\text{eff}} s}$. By conducting a direct synthesis for setpoints, we have as in e.g. Skogestad [6] that

$$c(s) = g^{-1} \frac{1}{(\tau_c + \theta_{\text{eff}})s} \tag{2.7}$$

$$= (1/k_{\max}) \frac{1}{(\tau_c + \theta_{\text{eff}})s} \tag{2.8}$$

This is a pure I-controller. With the SIMC-rule $\tau_c = \theta$ we get a I-controller with no more degrees of freedom. (Thight control).

2.2.5 Simulation of controller in closed loop

We will now consider the process as in equation (2.2) with the process conditions $(\tau_1, \tau_0/\tau_1, \zeta, \theta/\tau_1) = (1, 0.5, 0.2, 0)$. An open-loop step-response of the process is displayed in figure 2.6. Figure 2.7 shows a

[2] i.e. the corner-frequencies of $1/(\tau_1 s + 1)$ and $1/(\tau_0^2 s^2 + 2\tau_0\zeta s + 1)$ approaches each other

closed loop plot with a step in the controller setpoint at $t = 0$ and disturbance at $t = 30$. The disturbance enters as showed in figure A.1, page 45. Both the input and output is plotted. We observe that the process output y is following it's setpoint rather good[3], and the disturbance rejection is acceptable. The input is not oscillatory. (This might be due to the fact that we have a pure integral controller, so the high frequencies are damped by the controller.)

The controller was implemented with the equation:

$$c(s) = (1/k_{\max})\frac{1}{(\tau_c + \theta_{\text{eff}})s} \tag{2.9}$$

$$k_{\max} \;:\; \text{given above. (Peak gain)} \tag{2.10}$$

$$\theta_{\text{eff}} = (1.5 + \zeta \cdot 5) \cdot (\tau_0/\tau_1) \cdot \tau_1 \cdot 0.6^{(\tau_0/\tau_1)^2} \tag{2.11}$$

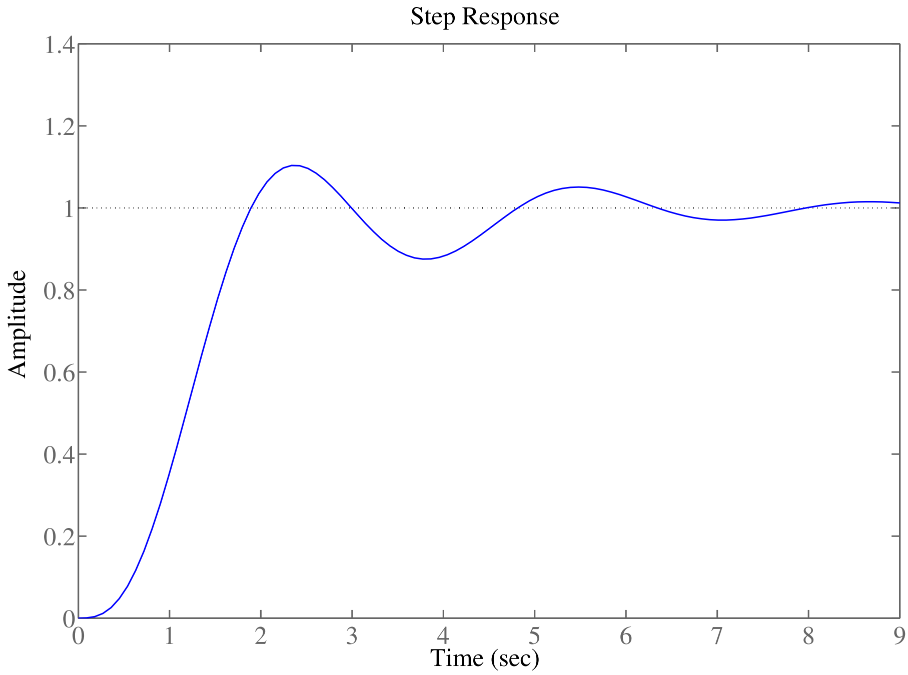

Figure 2.6: Step-response on the process $g(s) = \frac{k}{(\tau_1 s+1)(\tau_0^2 s^2+2\tau_0\zeta s+1)}e^{-\theta s}$ with conditions $(\tau_1, \tau_0/\tau_1, \zeta, \theta/\tau_1) = (1, 0.5, 0.2, 0)$

2.2.6 Performance of the pure I-controller compared to an optimal PI-controller

In order to assess the performance of the controller we compare it with an optimal PI-controller with respect to the measures IAE and TV. This is showed in figures 2.8 and 2.9 and will be discussed later.

[3]At least it seems good on the plot, but it might be a bit slow, keeping in mind that the dominant time constant $\tau_1 = 1$.

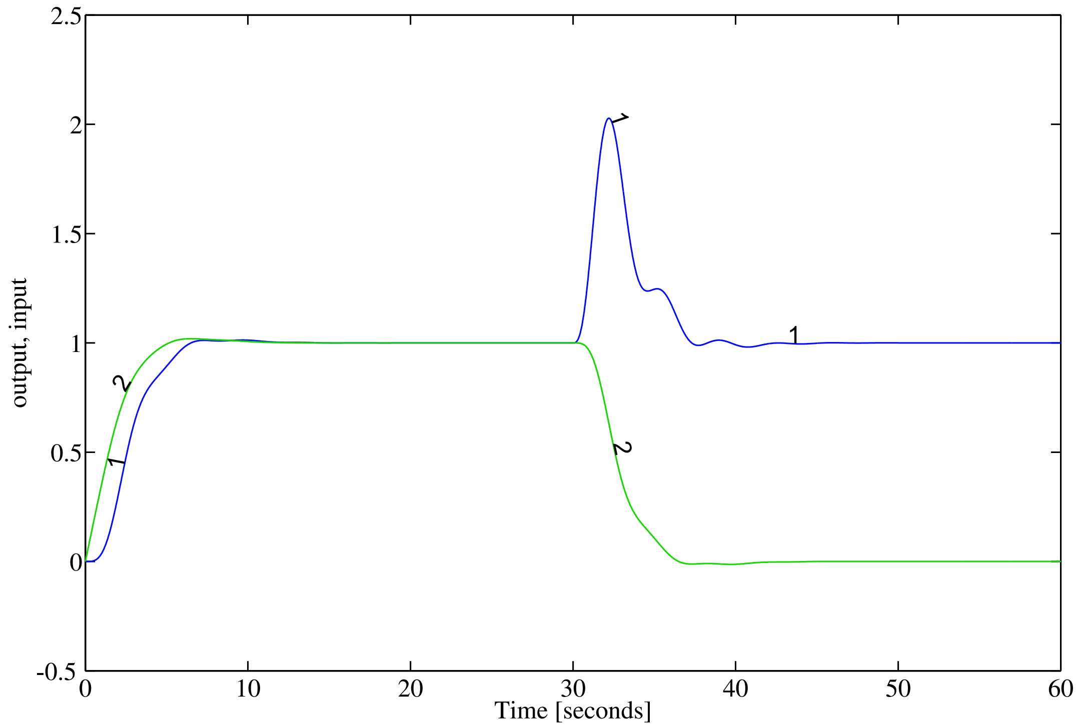

Figure 2.7: Closed loop response on the process $g(s) = \frac{k}{(\tau_1 s+1)\left(\tau_0^2 s^2+2\tau_0\zeta s+1\right)} e^{-\theta s}$ with conditions $(\tau_1, \tau_0/\tau_1, \zeta, \theta/\tau_1) = (1, 0.5, 0.2, 0)$ and controller $c(s) = (1/k_{\max}) \frac{1}{(\tau_c+\theta_{\text{eff}})s}$ with $\tau_c = \theta_{\text{eff}}$. Step in the controller setpoint at $t = 0$ and load disturbance at $t = 30$.
1: process output y. 2: controller output u.

IAE and TV are defined as follows:

$$\text{IAE} = \int_0^\infty |e(t)|\mathrm{d}t \tag{2.12}$$

$$\text{TV} = \sum_{i=1}^{\infty} |u_{i+1} - u_i| \tag{2.13}$$

Since one want to see how the controller is compared to both of these measures one may solve a multi-objective function problem:

$$\min \begin{cases} \text{IAE} \\ \text{TV} \end{cases}$$

subject to the constraints

The constraints in our problem are the process model with it's response to setpoint changes and disturbances, and of course technical issues like "only allow positive controller parameters". A naive way to solve this multi-objective problem is to follow these steps:

Step 1. Compute optimal PI-controller with respect to IAE only.

Step 2. Get TV from the solution, TV_0.

Step 3. Include TV in the constraints of the optimization, in the form $\text{TV} \leq \alpha^n \text{TV}_0$, where α is some factor $0 < \alpha < 1$, $n \in \mathbb{N}$ is some counter.

Step 4. Recompute the optimization to get a new point in the curve (See for example figure 2.8). Increase n by 1.

Step 5. Continue until we have enough points in the curve. (Go back to step 4).

The model we use (as a constraint) may vary. Look at figure 2.8. The line in the left-most figure (setpoint) was comupted by only doing a change in the setpoint, and let the process settle. In the middle figure (disturbance) we only performed a step in the disturbance, and let the process settle. In the right-most figure we performed steps in both setpoint and disturbance (and let the process settle inbetween and after). The dots show the same simulation perform on "our controller" as discussed in the previous section, with $\tau_c = \theta_{\text{eff}}$.

In figure 2.9 we have two other simulations also, when $\tau_c = 1.5\theta_{\text{eff}}$ and $\tau_c = 0.5\theta_{\text{eff}}$. Some performance/robustness measures were calculated both for the optimal curve and the I-controller, and are showed in table 2.2. Note that the following parameterization of the PI-controller was used in table 2.2:

$$c = K_c \left(1 + \frac{1}{\tau_I s}\right) = K_c + K_I \frac{1}{s} \tag{2.14}$$

i.e. $K_I = K_c/\tau_I$.

Let us look at figures 2.8 and 2.9. We observe that the setpoint-tracking for our controller is a large distance from the optimal PI-controller line, whereas for disturbance rejection it is close to the optimal line. Look at the $L = g(s)c(s)$ plots in figure 2.10. We see that the slope after the crossover-frequency is -1, then flat, then -2. This is close to what is desired in loopshaping [7, page 44], and the roll-off rate should imply good disturbance rejection, as we see from figure 2.8. The rather poor setpoint tracking may

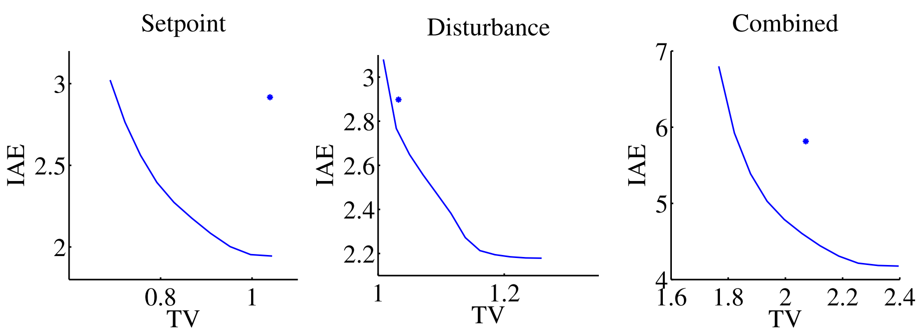

Figure 2.8: Process: $g(s) = \frac{k}{(\tau_1 s+1)(\tau_0^2 s^2+2\tau_0\zeta s+1)}$ with conditions $(\tau_0/\tau_1 = 0.5), (\zeta = 0.2), (k = 1)$. $\tau_c = \theta$.

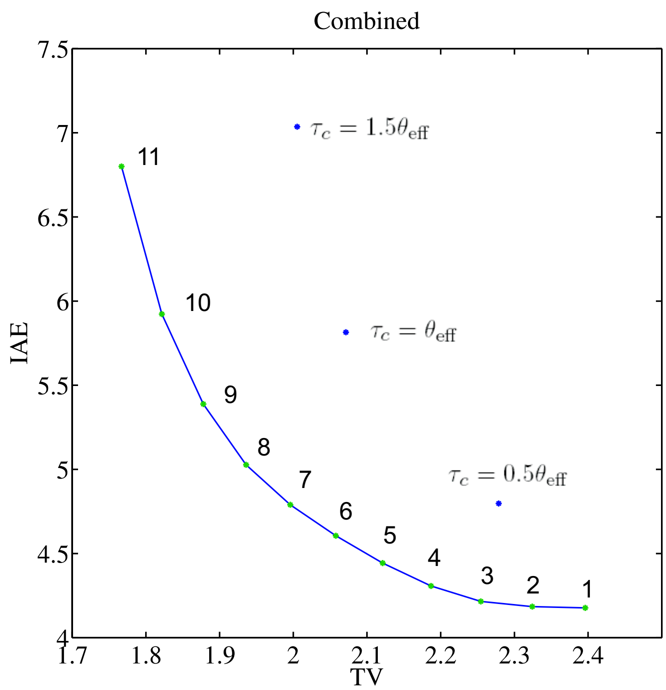

Figure 2.9: Process: $g(s) = \frac{k}{(\tau_1 s+1)(\tau_0^2 s^2+2\tau_0\zeta s+1)}$ with conditions $(\tau_0/\tau_1 = 0.5), (\zeta = 0.2), (k = 1)$ Robustness measures for the points in this figure are given in table 2.2.

Number	K_c	K_I	G_M	P_M	M_S	M_T
1	0.16	0.59	2.27	62.22	1.79	1.01
2	0.15	0.58	2.35	62.66	1.74	1.00
3	0.13	0.56	2.45	62.94	1.69	1.00
4	0.14	0.53	2.55	64.33	1.65	1.00
5	0.16	0.50	2.61	66.85	1.62	1.00
6	0.18	0.47	2.65	69.43	1.61	1.00
7	0.20	0.44	2.68	72.26	1.60	1.00
8	0.20	0.41	2.78	74.43	1.57	1.00
9	0.22	0.38	2.82	77.58	1.56	1.00
10	0.25	0.34	2.84	81.12	1.56	1.00
11	0.26	0.29	2.93	84.73	1.54	1.00
$\tau_c/\theta_{\text{eff}}$	K_c	K_I	G_M	P_M	M_S	M_T
1.5	0.00	0.28	5.01	71.10	1.28	1.00
1.0	0.00	0.36	4.01	66.81	1.36	1.00
0.5	0.00	0.47	3.01	60.12	1.51	1.02

Table 2.2: Robustness measures for points in figure 2.9

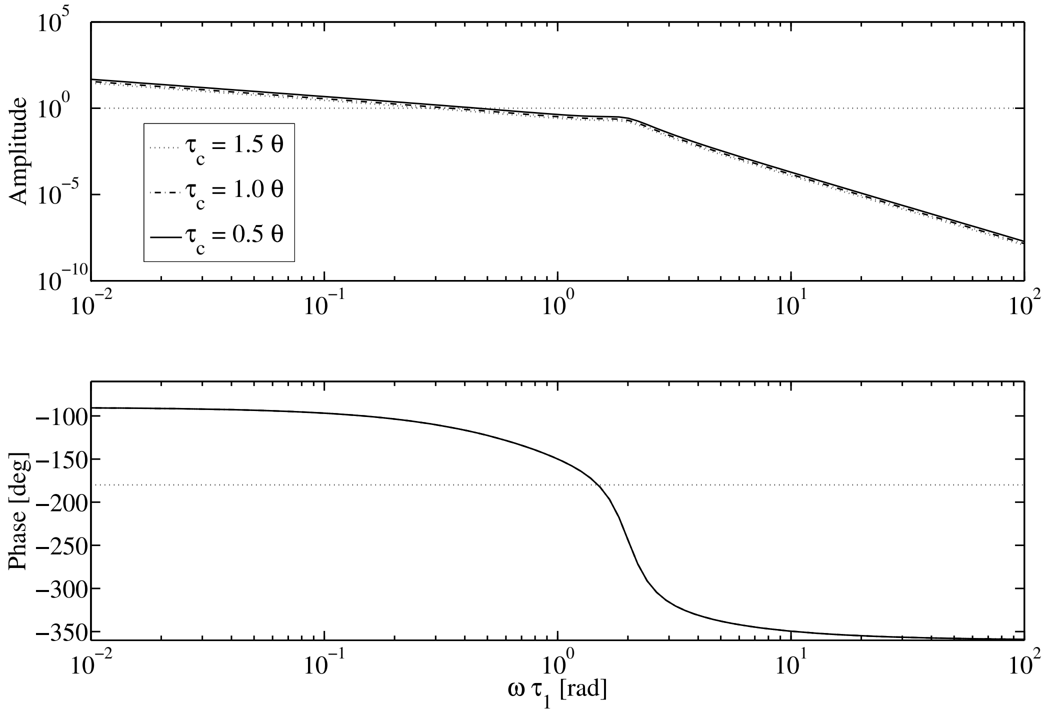

Figure 2.10: Bode-plots of $L = g(s)c(s)$.

be explained by the fact that the optimal controller has P-action, whereas our lack this feature. When we introduce a step in the setpoint the P-part of the controller will usually give the process a "kick" in the right directon, and then the I-part will "pull it" into it's correct operating point after some time.

Consider now figure 2.9 and table 2.2. In Seborg et al. [4, page 373] we have that:

> Guideline: In general, a well-tuned controller should have a gain margin (GM) between 1.7 and 4.0 and a phase margin (PM) between 30°and 45°.

Interestingly, none of the optimal controllers achieve a PM that is well-tuned. One of the differences between a PI controller and a PID is that the D-part in the PID-controller lifts the phase by about 90°[4]. Maybe the guideline from Seborg et al. [4] tells us that for saistfactory control of these processes we should include a D-part also. However, if we want to use PID-control with the SIMC-rules we need to have a second order (plus time-delay) model, see for example equation (1.1). There is quite a lot of work to modify/extend our current approximated model, which is a pure time delay, to such a model, and due to time-limitations this will not be considered in this report.

Other things we observe from figure 2.9 is that reducing the tuning parameter τ_c gives generally faster control (IAE goes down) but less roubstness (GM and PM down, M_S and M_T up). This is excatly as expected, remembering that τ_c is the desired closed-loop first-order reponse time constant. The same trend is also seen with the optimal controllers, but we also observers that optimal controller number 1 has more or less good GM, PM, and M_T. The M_S value of 1.79 is maybe a bit high, but it is in the range of the following guideline, also found in Seborg et al. [4, page 377]:

> Guideline: For a satisfactory control system, M_T should be in the range of 1.0-1.5 and M_S should be in the range of 1.2-2.0.

This observation hints that using a IAE-optimal PI-controller on this process will work OK, it is not necessary to consider optimization with respect to TV. Note that this obersvation is only valid for controllers with no D-action, as this surely would imply consideration of the TV measure also.

Furthermore, we see that the simple I-controller with the rule $\tau_c = \theta$ is not bad at all compared to an optimal PI-controller that is designed for both setpoint tracking and disturbance rejection. It is "best" for disturbance rejection, and if it should be improved we should focus on including some P-action to get the setpoint-tracking better.

As a last comment, consider the gain and phase margins in table 2.2. One observes that the optimal controller and our I-controller has quite similar phase margins, whereas the gain margin is considerably higher for the I-controller than the optimal one. This is expected, as we were very conservative with the gain approximation, while we tried to model the phase as good as possible in the expected closed loop bandwidth.

Some comments on practical issues with this method for tuning-evualtion

I had significat convergence problems with using this method, and this is the main (and very strong) reason why only one example is included in the report. One problem that often occured was that the simulation

[4] Depends of course on how one chooses to implement the usual D-filter

did not converge, because the gain was to high or the integral action to low, so we got an unstable process with the controller implemented. (SIMULINK-model, see A.3, page 47) When writing this comment I have not time to improve the rountine, but one simple solution I have thougth of is to include a saturation-block on the input, or possibly on the output. Then the objective function, which is to minimize IAE, will then just increase to a bound if the optimizer for some reason is going in this direction, but it would not diverge. Of course one should try to record if the optimization converged to this bound, because then we need to relax it until we converge to a solution that don't "need" input saturation, since the process we want to simulate does not have saturation inherent.

Another approach I would like to investigate is the method of Kristiansson and Lennartson [2]. This is a method based on frequency rather than the present method, which is based on time-domain simualtions. Some pros and cons of the current method and the one of Kristiansson and Lennartson [2] are discussed at the end of chapter 3, page 39.

2.3 Category C

During a project on tuning several of the low level PID control loops on the Snøhvit LNG plant one controller was encountered that was difficult to tune with the assumptions that was done at the time. I will use this process as an example loop for `Category C`-processes.

2.3.1 Look at step response ("empirical")

Let me explain what we did by first looking at a step response of the process g, as shown in figure 2.11. The solid line is approximately the response we got. We wanted to only use PI-control on this process,

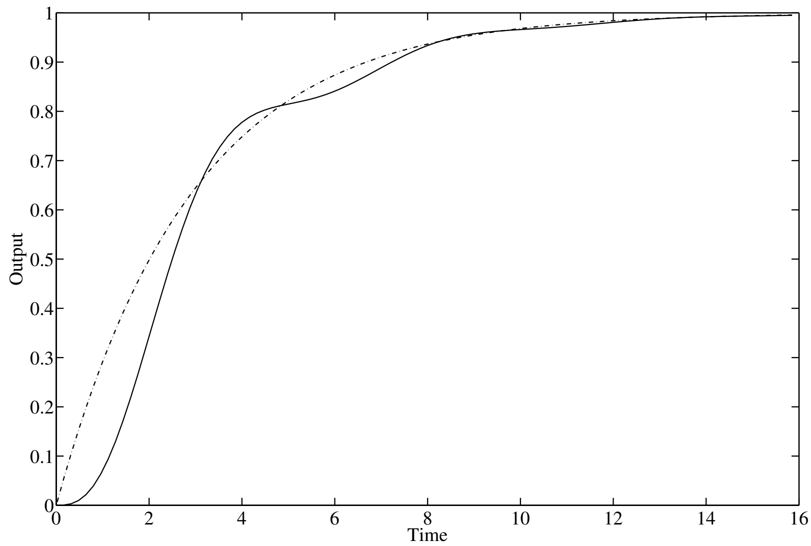

Figure 2.11: Solid line: $1/((\tau_1 s+1)(\tau_0^2 s^2+2\tau_0\zeta s+1))$. Dotted line: $1/(\tau_1 s+1)$. $\tau_1 = 2.9$, $\tau_0 = 0.75$ and $\zeta = 0.31$

and therefore we tried to fit the reponse by drawing a "1st order" line through the reponse signal. This is shown by the dotted line (only the dominat time constant). Intuitively this seems conservative, because the high-frequency gain is larger[5], but we are missing the oscillating peak. There was also some delay in this process that we have neglected for the current discussion (and in figure 2.11). The delay was about 0.5 time units. The tuning was done on a rigourous model of a large part of the plant, so the model we were tuning on was in nature non-linear. The solid line in figure 2.11 is a approximated response.

When tuning this controller we did not have the second order time τ_0 constant (This was found later by approximating the reponse in MATLAB), so we selected the delay to be the observed delay in the process and we used the SIMC rules directly with $\tau_c = \theta$. The result was an unstable loop.

2.3.2 Systematic design based on model: Naive use of half rule

Let us now look at this response in the frequency domain, shown in figure 2.12. The vertical dotted line in the figure shows $1/\theta_{\text{eff}}$. The effective delay θ_{eff} is in this case deduced from the half-rule, i.e. $g_{\text{approx}} = e^{-\theta_{\text{eff}} s}/(\tau s + 1)$, $\theta_{\text{eff}} = 0.5\tau_0$ and $\tau = \tau_1 + 0.5\tau_0$. That is, we use the half-rule [6] naively by simply ignoring the damping coefficient ζ and just distributing the largest neglected time constant τ_0 evenly between τ and θ_{eff}.[6] From figure 2.12 we see that the difference between g_{τ_1} (only the dominant

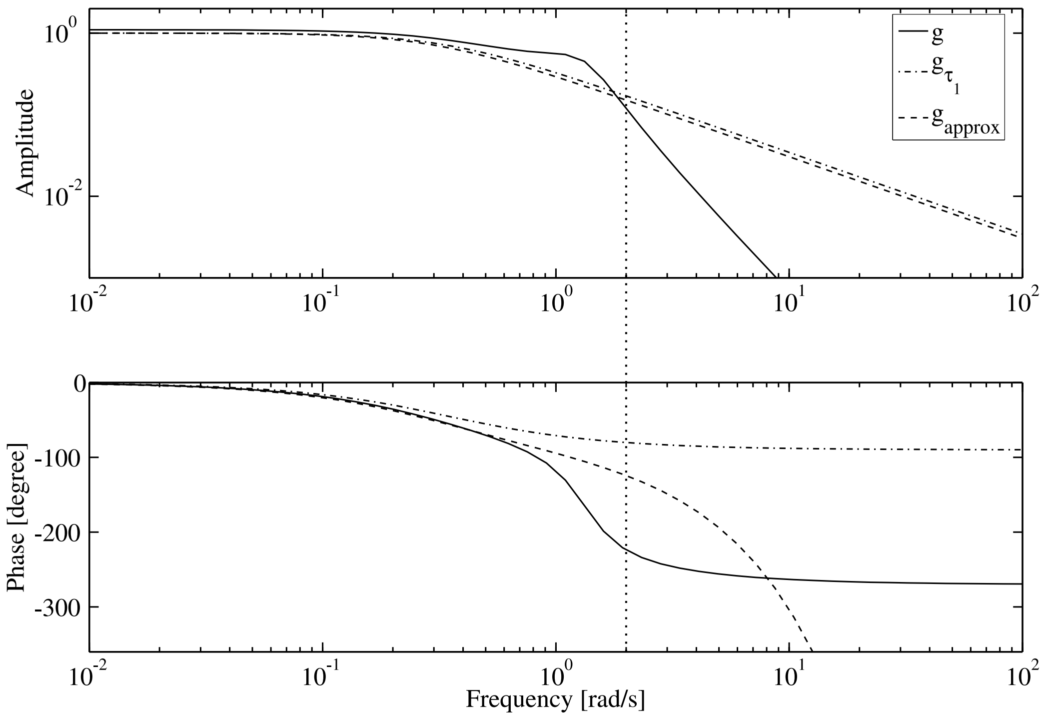

Figure 2.12: g: $1/((\tau_1 s + 1)(\tau_0^2 s^2 + 2\tau_0\zeta s + 1))$. g_{τ_1} : $1/(\tau_1 s + 1)$. $g_{\text{approx}} = e^{-0.5\tau_0 s}/((\tau_1 + 0.5\tau_0)s + 1)$. $\tau_1 = 2.9$, $\tau_0 = 0.75$ and $\zeta = 0.31$. Vertical dotted line is $1/(0.5\tau_0)$, i.e. the assumed closed-loop bandwidth.

time-constant) and g_{approx} is minor in amplitude, and both of them underestimates the gain of the process transferfunction g. In phase the approximation with delay is not so bad, and it follows g until about -80°in

[5]Steeper ascent at $t = 0$ means smaller time constant which leads to a larger corner-frequency 1τ.

[6]Perhaps one could argue that we should use the half rule this way: $\theta_{\text{eff}} = \theta_0 + \tau_0$, $\tau = \tau_1 + \tau_0$, since the $\tau_0^2 s^2 + 2\zeta\tau_0 s + 1$ term in the denominator acutally consists of *two* poles. This would yield more conservative PI-settings by using SIMC-rules, but due to project time-constraints this is yet to be investigated.

phase, but when the two curves differ the difference is rather significant. Actually it is very wrong in the assumed bandwidth-area, and this will surely yield problems when we later use the SIMC-rules.

In addition we have a problem with the gain, see figure 2.12, where we see that the approximation is optimistic. (The process gain is higher, especially around the peak frequnecy, and this error is inside the assumed band-width of the resulting controller).

The loop transfer function $L = gc$ with a SIMC-tuned PI controller for this process is shown in figure 2.13. We see that in closed loop this will be unstable. For the unstable loop the gain margin is 0.6 (absolute) and the phase margin is -30°. The instabillity is not surprising when approximations in both gain and phase are optimistic, and they are both wrong (see above) around the assumed bandwidth frequnecy $\omega_{\text{bandwith}} \approx 1/\theta_{\text{effective}}$, see Skogestad [6, page 114].

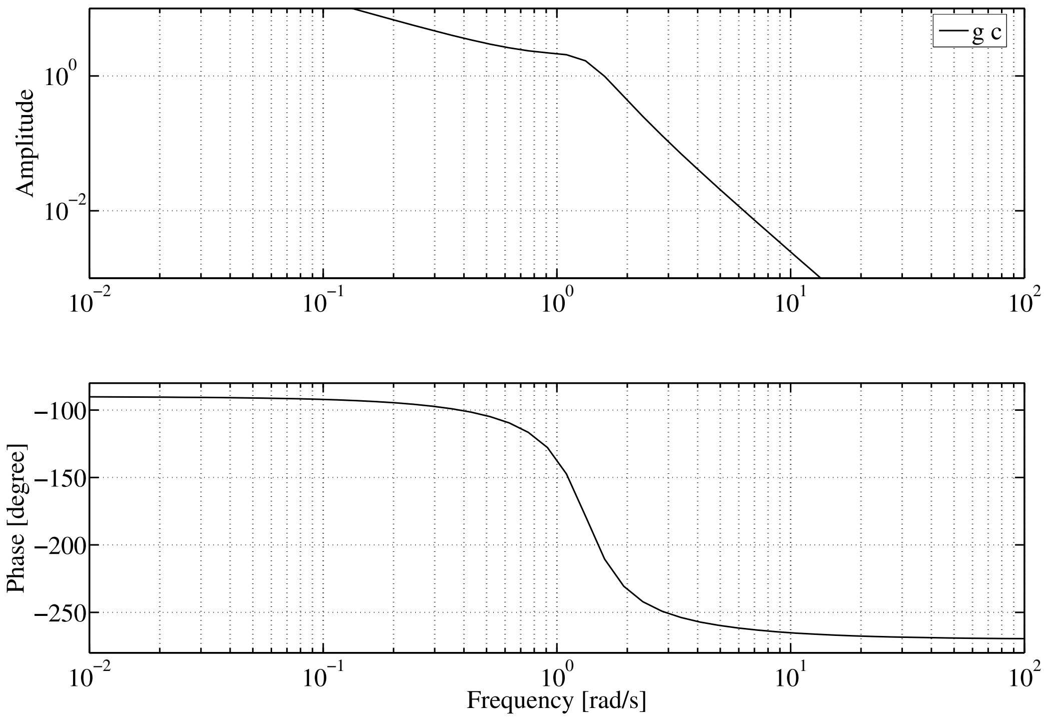

Figure 2.13: $L = g(s)c(s)$. g is the process *without any delay*, and $c(s)$ is a PI-controller with settings given by the SIMC-rules of the process model $g_{\text{approx}} = e^{-0.5\tau_0 s}/((\tau_1 + 0.5\tau_0)s + 1)$

2.3.3 Systematic designed based on model: Making an approximate model for SIMC-tuning purposes

As we see from figure 2.12 the gain at the peak is neglected when we estimate the process without considering the damping coefficient ζ. The phase is also rather poorly modelled in this case.

In Seborg et al. [4, page 353], we have that the resonant peak frequency is:

$$\omega_r = \frac{\sqrt{1 - 2\zeta^2}}{\tau_0} \tag{2.15}$$

Assuming we want to use a PI-controller, we should preferably model the process as a 1st order + delay process. In gain, we should be safe if we can maintain a high gain in the model so that the resonant peak is "well covered". See figure 2.14. If the amplitude ratio at the peak is less than the low-frequnecy gain for the process transfer function we could model the gain by a first order transferfunction[7]:

$$g_{\text{gain}} = \frac{1}{\tau_{\text{gain}} s + 1} \tag{2.16}$$

$$\tau_{\text{gain}} = \frac{\tau_0}{\sqrt{1 - 2\zeta^2}} \tag{2.17}$$

This approxiamtion will be safe in gain for all frequencies, and it will be good for low frequencies. What we did was to place the corner-frequency "above" the process gain, so we should stay conservative with respect to gain for all frequencies ω.

For the phase we might use the same rule as for `Category B`, which is that we adjust the phase by a delay until the phase "touches" the process delay. This rule is un-safe (but good at low frequencies) in phase, but again, as for `Category B`, we have a safe approximation in gain, and overall the approximation should be OK.

These two approximations is conducted and displayed as the dotted line in figure 2.14. Note that we had to modifty the rule (equation (2.5)) slightly. This implies that the rule is not fully developed yet. The necassary modification was to change the "offset parameter" from 1.5 to 0.5.[8]

Since we have a 1st order + time-delay model we can use the SIMC settings to find nice PI-controller paramteres. This was done, and a closed-loop simulation is showed in figure 2.15. The closed loop response is rather nice, with acceptable input usage and good setpoint-tracking and disturbance rejection. In figure 2.16 the loop transfer function $L = g(s)c(s)$ is plotted and the usual roboustness-measures is documented.

2.3.4 Conclusion

By looking at the reponses in time-domain (figure 2.15), frequency-domain (figure 2.16) and noting that (GM, PM, M_S, M_T) = (3.90, 44.05, 1.77, 1.34) we can conclude that as a simple approach of approximating a model as conduced in this chapter, we get a well tuned controller with respect to the guidelines from Seborg et al. [4], see page 16. Only the M_S-value is a bit over it's recommended value. (All theses measures are defined in appendix B). This shows that it is possible to get satisfactory control by first reducing the model to a first order + time-delay, and thereafter use this model as a basis for implementing a SIMC-tuned PI controller.

2.3.5 Further work

Only one process has been tested, and we need to perform more testing to verify the results of this proposed model-reduction. It would also be worthwhile to perform comparisons with other familiar tuning methods.

[7]Note that this approximation should be OK when there is no peak above the low-frequency gain. This should be true for all processes in this category.

[8]The very first number one sees when looking at equation (2.5)

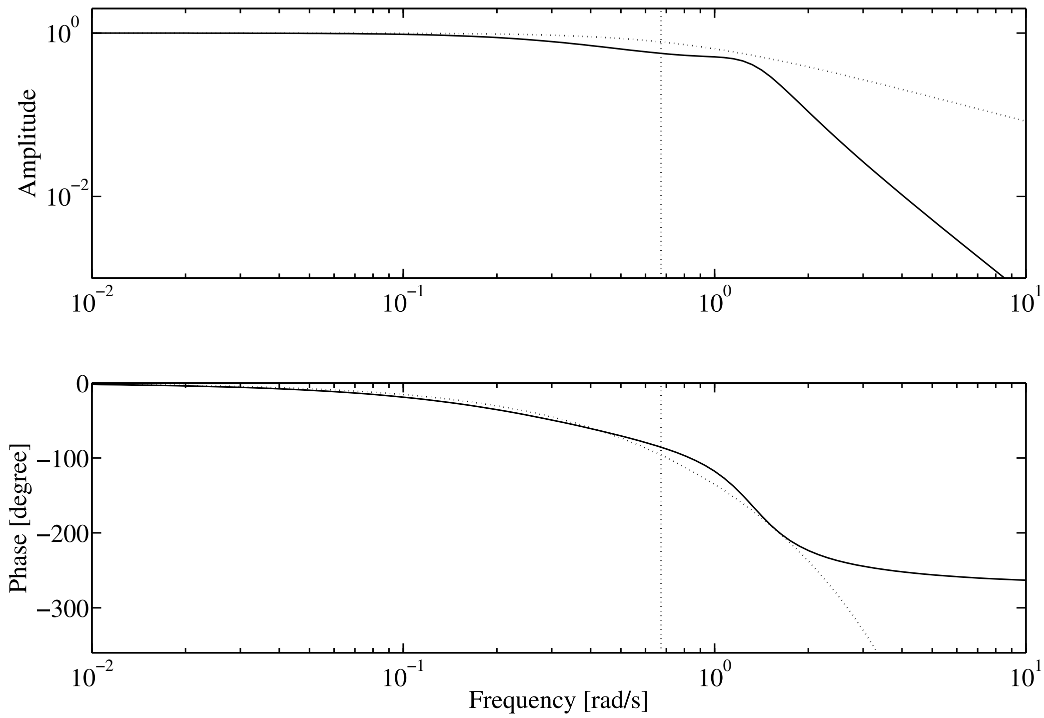

Figure 2.14: Solid line: $g(s)$ *without delay.* Dotted line: $g_{\text{approx}} = \frac{1}{\tau_{\text{gain}}s+1}e^{-\theta_{\text{eff}}s}$, $\theta_{\text{eff}} = (0.5+\zeta\cdot 5)\cdot 0.6^{(\tau_0/\tau_1)^2}$. Vertical dotted line: $1/\theta_{\text{eff}}$.

2.4 Category A

In this case the oscillating mode of the process is the dominating one, so we may regrad this as a pure oscillating second order process[9]. Some work has been conduced on this subject by prof. Skogestad[10], and this section will be based upon his work. The `.txt` file (see footnote) is the basis for the following discussion.

2.4.1 Extending the SIMC rules to cover $0 < \zeta < 1$ for a seond order plus delay transferfunction

We have a process model on the form

$$g(s) = k\frac{e^{-\theta s}}{\tau_0^2 s^2 + 2\tau_0\zeta s + 1} \tag{2.18}$$

with $0 < \zeta < 1$. By direct synthesis we can derive an IMC controller for this process that would yield good setpoint tracking, but it is not necessarily good for disturbance rejection. We can show that the resulting controller for a double integrating process, i.e. $\tau_0 = \infty, \tau_0\zeta = 0$, the resulting IMC-controller is a pure D-controller, which would certainly be poor for disturbance rejection. Skogestad [6, page 93] have another

[9] $\tau_0/\tau_1 \to \infty$

[10] `http://www.nt.ntnu.no/users/skoge/publications/2003/tuningPID/more/extensions/oscillating.txt`

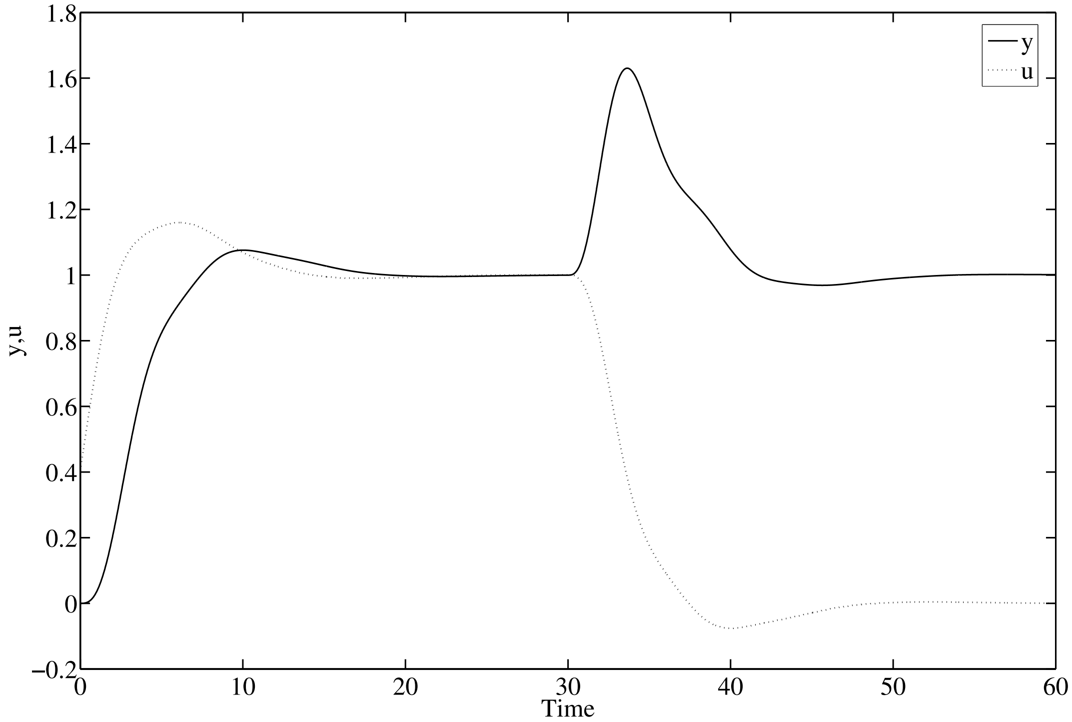

Figure 2.15: Simulation of the closed loop system $gc/(1+gc)$, with g process model *without delay* and c given from SIMC settings on the model $g_{\text{approx}} = \frac{1}{\tau_{\text{gain}} s+1} e^{-\theta_{\text{eff}} s}$, $\theta_{\text{eff}} = (0.5 + \zeta \cdot 5) \cdot 0.6^{(\tau_0/\tau_1)^2}$. Setpointchange from 0 to 1 at $t = 0$ and unit step in disturbance at $t = 30$. y is the process output and u is the input to the process from the controller. $\text{IAE}_{\text{setpoint}} = 3.88$, $\text{TV}_{\text{setpoint}} = 0.94$, $\text{IAE}_{\text{disturbance}} = 3.37$, $\text{TV}_{\text{disturbance}} = 1.16$

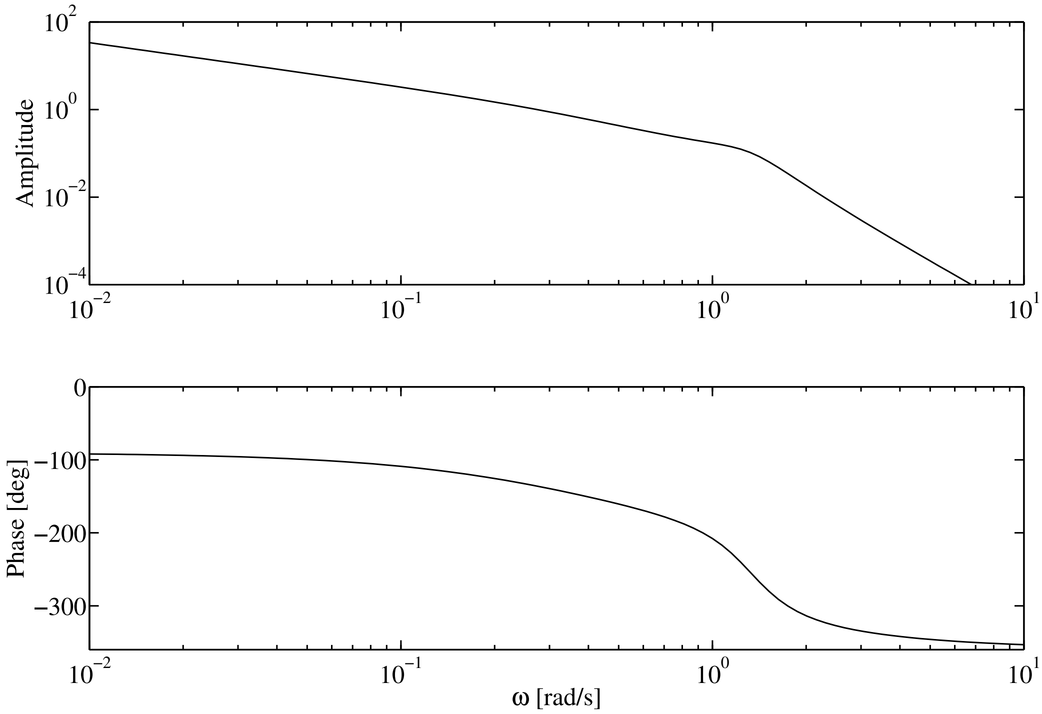

Figure 2.16: Loop transferfunction $L = g \cdot c$ with improved model and a SIMC-tuned PI controller. (GM, PM, M_S, M_T) = (3.90, 44.05, 1.77, 1.34).

recommendation for the double integrating process, which includes both P- and I-action. The question is: Can we find a simple "correction" for integrating processes that also includes oscillating processes?

Appendix C gives a derivation of a set of PID-tuning rules based upon comparing the following:

- IMC setpoint settings (for a process on the form 2.18) (A)
- SIMC-correction for integrating process with $\zeta > 1$ (B)
- SIMC-correction for double-integrating process with $\zeta = 0$ (C)

The derivation concludes that a possible set of tuning rules is found by interpolating between the SIMC-correction for integrating process with $\zeta > 1$ and the SIMC-correction for double-integrating processess with $\zeta = 0$ in the following way:

Interpolation rule

Let $c(s)$ be a controller on the form

$$c(s) = K_c' + K_I'/s + K_D's \tag{2.19}$$

We then have the following tuning for an underdamped stable second order process:

- $K'_c = \max\{A, X\}$, where $X = B$ for $\zeta \geq 1$ and $X = \zeta B' + (1-\zeta)C$ for $\zeta < 1$
- $K'_I = \max\{A, X\}$, where $X = B$ for $\zeta \geq 1$ and $X = \zeta B' + (1-\zeta)C$ for $\zeta < 1$
- K'_D from either A, B or C

where A, B, B' and C are defined by table 2.3

	K'_c	K'_I	K'_D
A	$\frac{1}{k''}\frac{1}{\tau_c+\theta}\frac{2\zeta}{\tau_0}$	$\frac{1}{k''}\frac{1}{\tau_c+\theta}\frac{1}{\tau_0^2}$	$\frac{1}{k''}\frac{1}{\tau_c+\theta}$
B	$\frac{1}{k''}\frac{1}{(\tau_c+\theta)^2}\left(1+4(\tau_c+\theta)+\left(\zeta+\sqrt{\zeta^2-1}\right)\frac{1}{\tau_0}\right)$	$\frac{1}{k''}\frac{1}{(\tau_c+\theta)^2}\frac{1}{\tau_0}\left(\zeta+\sqrt{\zeta^2-1}\right)$	$\frac{1}{k''}\frac{1}{\tau_c+\theta}$
B'	$\frac{1}{k''}\frac{1}{(\tau_c+\theta)^2}\left(1+4(\tau_c+\theta)+\frac{\zeta}{\tau_0}\right)$	$\frac{1}{k''}\frac{1}{(\tau_c+\theta)^2}\frac{\zeta}{\tau_0}$	$\frac{1}{k''}\frac{1}{\tau_c+\theta}$
C	$\frac{1}{2k''}\frac{1}{(\tau_c+\theta)^2}$	$\frac{1}{16k''}\frac{1}{(\tau_c+\theta)^3}$	$\frac{1}{k''}\frac{1}{\tau_c+\theta}$

Table 2.3: Expressions for use in interpolation rule. $k'' = k/\tau_0^2$. Note that B' is obtained by setting $\sqrt{\zeta^2-1} = 0$ in B.

2.4.2 Discussion

Tables 2.4, 2.5 and 2.6 shows GM, PM, M_S and M_T for the interpolation rule and Ziegler-Nichols (ZN) and Tyreus-Luyben (TL) tuning[4]. Figures 2.17 and 2.18 shows time-domain respones for the interpolation rule compared with ZN and TL tuning on the process (2.18) with $(k, \tau_0, \theta) = (1, 1, 0.5)$ and $\zeta = 0.15$ and $\zeta = 0.85$ respectively. Note that the robustness margings in tables 2.4, 2.5 and 2.6 are calculated with a derivative filter equal to $1/(\tau_f \tau_D s + 1)$, $\tau_f = 0.000001 \approx 0$, while in the time-domain simulations a filter of $\tau_f = 0.01\tau_D$ is used, see appendix A.2 for SIMULINK-model. Tables 2.7, 2.8 and 2.9 indicates IAE and TV vaules for both tracking and disturbance rejection for the different tuning procedures.

From the tables and figures mentioned above one observes that

- For very low delay and small ζ, see table 2.4, the rule $\tau_c = \theta$ yields an unstable closed loop. An obvious solution to this would be to increase τ_c until we achieve a desired response, as $\tau_c = \theta$ may be too thight control in this case. [11]
- For processes with moderate time-delay, see table 2.5 the interpolating-rule tuning with $\tau_c = \theta$ yields a controller that is inbetween ZN and TL with respect to all measures for most of the ζ-values, that is, we observe:

$$\begin{aligned}
\mathrm{GM}^{\mathrm{TL}} \leq \mathrm{GM}^{\mathrm{interpolation}} &\leq \mathrm{GM}^{\mathrm{ZN}} \quad 0.55 \leq \zeta \leq 0.95 \\
\mathrm{PM}^{\mathrm{TL}} \leq \mathrm{PM}^{\mathrm{interpolation}} &\leq \mathrm{PM}^{\mathrm{ZN}} \quad 0.25 \leq \zeta \leq 0.95 \\
\mathrm{M}_{\mathrm{S}}^{\mathrm{TL}} \leq \mathrm{M}_{\mathrm{S}}^{\mathrm{interpolation}} &\leq \mathrm{M}_{\mathrm{S}}^{\mathrm{ZN}} \quad 0.15 \leq \zeta \leq 0.95 \\
\mathrm{M}_{\mathrm{T}}^{\mathrm{TL}} \leq \mathrm{M}_{\mathrm{T}}^{\mathrm{interpolation}} &\leq \mathrm{M}_{\mathrm{T}}^{\mathrm{ZN}} \quad 0.15 \leq \zeta \leq 0.95
\end{aligned}$$

[11] We have here that $\tau_0/\theta = 1/0.05 = 20$, so there is very little delay in the process, and by setting $\tau_c = \theta$ we try to do strict cancellation of the process with the controller. This should be feasible, but since we have $(\tau_c + \theta)$ in the denominator for the gain, we get very high gain when $\theta = \tau_c$ is small, so the controller will be "too thight."

- The same trend is observed when θ is large, see table 2.6.
- Consider the two time-domain simulations in figure 2.17 and 2.18 and tables 2.5 and 2.8. We obeserve that for small ζ, e.g. $\zeta = 0.15$, the interpolation rule with $\tau_c = \theta$ might be a bit to thight. We see this from the time-domain reponse and the relatively low GM and PM - vaules compared to the ZN- and TL-tuning, see second row in table 2.5. The IAE for the interpolation rule is low and the TV is high when $\zeta = 0.15$ compared to ZN- and TL- tuning, confirming that the tuning might be too thight. When $\zeta = 0.85$ we observe that the interpolation rule is superior to the other two methods with respect to IAE, and TV is moderate. As mentioned above, the GM, PM, M_S and M_T for interpolation rule is inbetween the corresponding vaules for ZN and TL. This seems promising, as the ZN-tuning rules tend to produce oscillatory reponses and large overshoots for set-point changes, but still are widely used as a benchmark for evalutaion of different tuning methods, while the TL-tuning settings should be generally be a more conservative set of tuning-rules[4, page 319].

Overall the interpolation-rule seems like a promsing start for deriving PID-tuningrules for underdamped second-order processes. Further investigation could be to look at how one should choose the tuning parameter τ_c.

	Interpolation rule				Ziegler-Nichols				Tyreus-Luyben			
ζ	GM	PM	M_s	M_T	GM	PM	M_s	M_T	GM	PM	M_s	M_T
0.05	–	–	–	–	55.41	19.72	2.9	3.1	105.85	40.58	1.6	1.5
0.15	–	–	–	–	27.23	18.05	3.2	3.4	52.36	38.33	1.6	1.6
0.25	2.85	38.72	1.8	1.6	20.11	17.87	3.2	3.5	38.88	38.67	1.6	1.6
0.35	2.88	40.86	1.8	1.5	16.38	18.25	3.2	3.4	31.83	39.59	1.6	1.5
0.45	2.91	42.91	1.8	1.4	14.16	18.48	3.1	3.4	27.63	40.59	1.5	1.5
0.55	2.94	44.88	1.7	1.4	12.56	18.82	3.1	3.3	24.61	41.61	1.5	1.5
0.65	2.96	46.75	1.7	1.3	11.33	19.23	3.0	3.2	22.29	42.62	1.5	1.4
0.75	2.99	48.52	1.7	1.3	10.40	19.53	3.0	3.2	20.54	43.57	1.5	1.4
0.85	3.01	50.20	1.7	1.2	9.65	19.82	2.9	3.1	19.14	44.48	1.5	1.4
0.95	3.03	51.77	1.7	1.2	8.98	20.23	2.8	3.0	17.86	45.36	1.4	1.4

Table 2.4: $\tau_0 = 1$, $k = 1$, $\theta = 0.05$, $\tau_c = \theta$. Interpolation rule with $\tau_c = \theta$ is unstable for $\zeta = \{0.05, 0.15\}$.

	Interpolation rule				Ziegler-Nichols				Tyreus-Luyben			
ζ	GM	PM	M_s	M_T	GM	PM	M_s	M_T	GM	PM	M_s	M_T
0.05	2.78	21.07	2.7	2.9	12.34	41.50	2.1	1.6	36.89	92.90	1.3	1.0
0.15	2.98	41.02	1.7	1.4	5.71	42.47	2.1	1.5	14.85	98.54	1.3	1.0
0.25	3.14	61.35	1.6	1.0	4.20	42.47	2.0	1.5	10.34	104.09	1.3	1.0
0.35	3.14	61.35	1.6	1.0	3.53	42.31	2.0	1.5	8.40	109.52	1.3	1.0
0.45	3.14	61.35	1.6	1.0	3.15	42.03	2.0	1.5	7.30	114.83	1.3	1.0
0.55	3.14	61.35	1.6	1.0	2.90	41.69	2.0	1.5	6.59	119.92	1.3	1.0
0.65	3.14	61.35	1.6	1.0	2.72	41.33	2.0	1.5	6.10	124.49	1.3	1.0
0.75	3.14	61.35	1.6	1.0	2.59	40.96	2.1	1.5	5.73	127.72	1.3	1.0
0.85	3.14	61.35	1.6	1.0	2.49	40.56	2.1	1.5	5.44	126.78	1.3	1.0
0.95	3.16	56.98	1.6	1.1	2.41	40.22	2.1	1.5	5.22	119.77	1.3	1.0

Table 2.5: $\tau_0 = 1$, $k = 1$, $\theta = 0.5$, $\tau_c = \theta$.

ζ	Interpolation rule				Ziegler-Nichols				Tyreus-Luyben			
	GM	PM	M_s	M_T	GM	PM	M_s	M_T	GM	PM	M_s	M_T
0.05	1.85	62.71	2.2	1.2	1.27	92.30	5.1	4.2	3.05	91.73	1.6	1.0
0.15	3.14	61.35	1.6	1.0	1.66	94.61	2.7	1.7	3.82	93.94	1.4	1.0
0.25	3.14	61.35	1.6	1.0	1.72	96.31	2.5	1.6	3.81	96.30	1.4	1.0
0.35	3.14	61.35	1.6	1.0	1.73	97.03	2.5	1.5	3.75	98.64	1.4	1.0
0.45	3.14	61.35	1.6	1.0	1.74	96.46	2.5	1.5	3.70	100.92	1.4	1.0
0.55	3.14	61.35	1.6	1.0	1.74	94.16	2.4	1.5	3.66	103.14	1.4	1.0
0.65	3.14	61.35	1.6	1.0	1.75	89.59	2.4	1.5	3.63	105.30	1.4	1.0
0.75	3.14	61.35	1.6	1.0	1.75	82.92	2.4	1.5	3.60	107.37	1.4	1.0
0.85	3.14	61.35	1.6	1.0	1.75	75.85	2.4	1.5	3.58	109.37	1.4	1.0
0.95	3.14	61.35	1.6	1.0	1.75	69.87	2.4	1.5	3.56	111.27	1.4	1.0

Table 2.6: $\tau_0 = 1$, $k = 1$, $\theta = 1.5$, $\tau_c = \theta$.

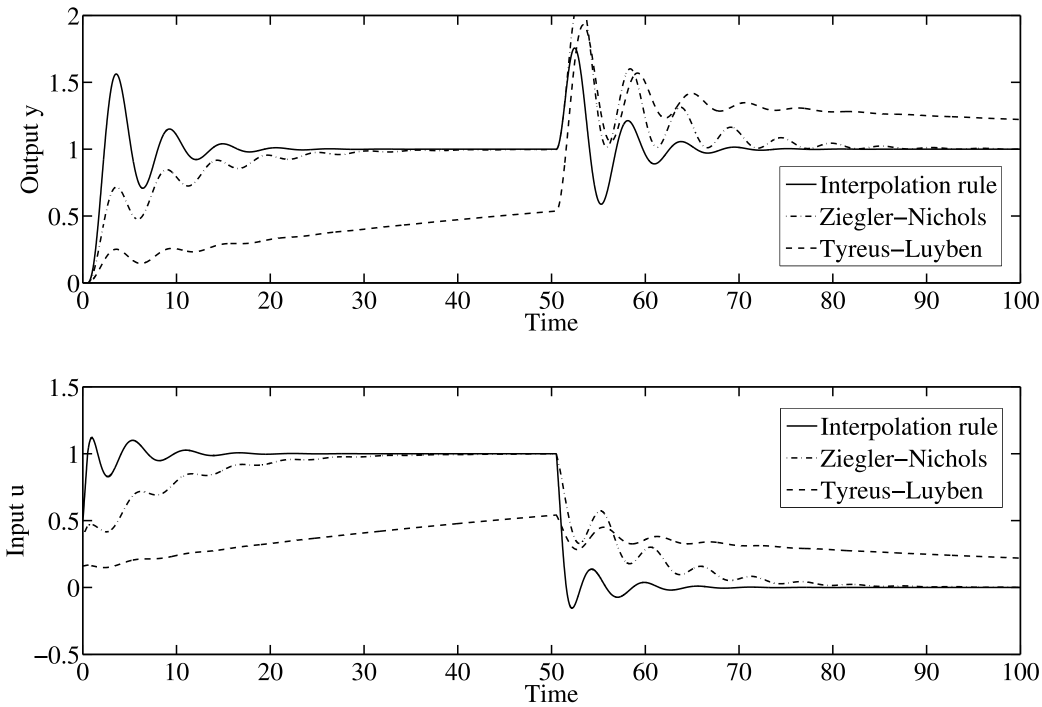

Figure 2.17: Time-domain simulations for the process $g(s) = \frac{k}{\tau_0^2 s^2 + 2\tau_0\zeta s + 1}$, with conditions $(k, \tau_0, \theta, \zeta) = (1, 1, 0.5, 0.15)$. Setpointchange at $t = 0$ and disturbance at $t = 50$ time units.

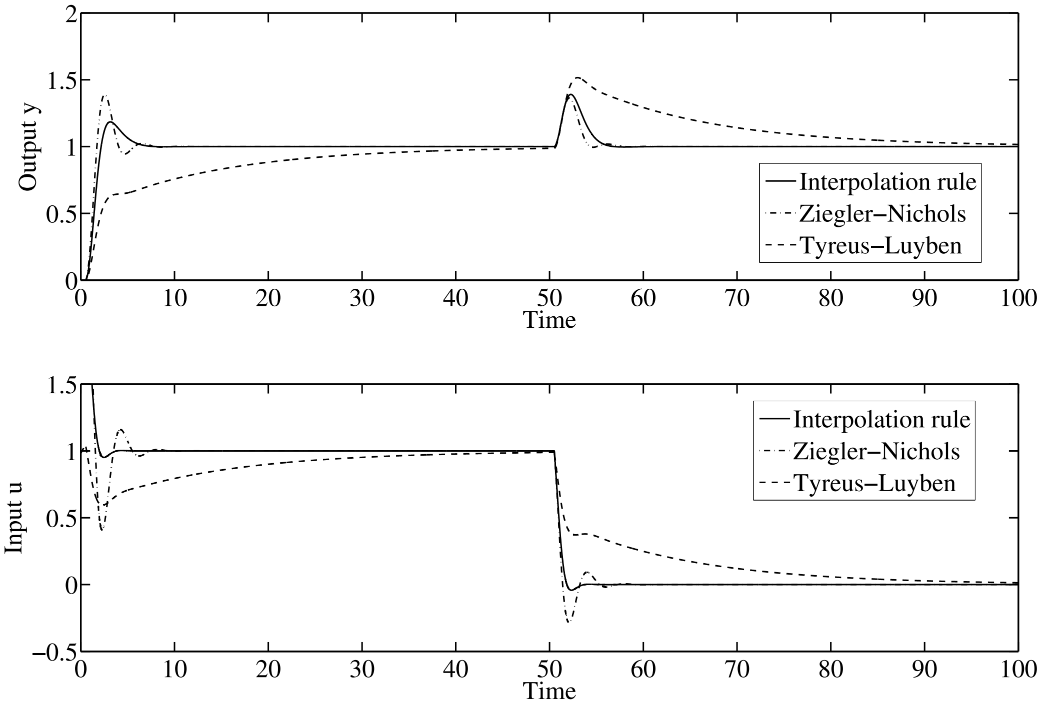

Figure 2.18: Time-domain simulations for the process $g(s) = \frac{k}{\tau_0^2 s^2 + 2\tau_0 \zeta s + 1}$, with conditions $(k, \tau_0, \theta, \zeta) = (1, 1, 0.5, 0.85)$. Setpointchange at $t = 0$ and disturbance at $t = 50$ time units.

	Interpolation rule				Ziegler-Nichols				Tyreus-Luyben			
	Setpoint		Disturbance		Setpoint		Disturbance		Setpoint		Disturbance	
ζ	IAE	TV	IAE	TV	IAE	TV	IAE	TV	IAE	TV	IAE	TV
0.05	–	–	–	–	3.12	4.54	2.51	3.96	15.77	1.19	15.77	2.13
0.15	–	–	–	–	2.33	15.63	0.65	4.41	3.48	2.89	3.48	2.07
0.25	0.41	87.04	0.02	2.03	1.94	26.91	0.32	4.46	1.85	4.66	1.67	2.00
0.35	0.42	79.09	0.02	1.94	1.65	37.11	0.20	4.41	1.31	6.37	1.02	1.95
0.45	0.44	72.08	0.02	1.85	1.47	46.90	0.13	4.31	1.05	7.93	0.71	1.89
0.55	0.45	66.97	0.02	1.74	1.32	56.01	0.10	4.23	0.90	9.61	0.53	1.85
0.65	0.47	60.91	0.02	1.70	1.21	65.32	0.08	4.11	0.80	10.98	0.41	1.77
0.75	0.49	59.83	0.03	1.63	1.11	73.47	0.06	4.03	0.73	12.44	0.33	1.74
0.85	0.50	54.57	0.03	1.58	1.04	82.42	0.05	3.97	0.70	13.71	0.28	1.69
0.95	0.51	51.36	0.03	1.53	0.97	91.39	0.04	3.96	0.67	15.14	0.23	1.65

Table 2.7: IAE and TV for setpoint change and disturbance. $\tau_0 = 1$, $k = 1$, $\theta = 0.05$, $\tau_c = \theta$. Interpolation rule with $\tau_c = \theta$ is unstable for $\zeta = \{0.05, 0.15\}$.

	Interpolation rule				Ziegler-Nichols				Tyreus-Luyben			
	Setpoint		Disturbance		Setpoint		Disturbance		Setpoint		Disturbance	
ζ	IAE	TV	IAE	TV	IAE	TV	IAE	TV	IAE	TV	IAE	TV
0.05	5.22	3.70	4.78	4.13	21.91	0.86	21.91	2.08	78.32	0.31	77.91	0.88
0.15	3.76	1.63	3.01	1.95	6.61	0.87	6.61	2.02	49.78	0.63	49.55	1.16
0.25	3.46	1.04	2.53	1.27	3.57	1.31	3.57	2.01	33.22	0.76	33.11	1.23
0.35	2.81	1.12	1.88	1.24	2.51	1.83	2.33	1.97	23.48	0.81	23.43	1.23
0.45	2.44	1.19	1.52	1.20	2.18	2.32	1.69	1.96	17.48	0.83	17.46	1.20
0.55	2.20	1.36	1.30	1.20	2.03	2.96	1.30	1.91	13.59	0.86	13.57	1.15
0.65	2.04	1.54	1.15	1.17	1.95	3.42	1.04	1.91	10.94	0.88	10.93	1.11
0.75	1.93	1.71	1.06	1.16	1.89	4.05	0.86	1.91	9.05	0.92	9.05	1.06
0.85	1.85	1.93	1.01	1.16	1.86	4.67	0.73	1.89	7.65	0.94	7.65	1.02
0.95	1.89	2.27	0.91	1.24	1.84	5.23	0.63	1.87	6.59	0.96	6.59	1.02

Table 2.8: IAE and TV for setpoint change and disturbance. $\tau_0 = 1$, $k = 1$, $\theta = 0.5$, $\tau_c = \theta$.

	Interpolation rule				Ziegler-Nichols				Tyreus-Luyben			
	Setpoint		Disturbance		Setpoint		Disturbance		Setpoint		Disturbance	
ζ	IAE	TV	IAE	TV	IAE	TV	IAE	TV	IAE	TV	IAE	TV
0.05	16.73	1.77	42.14	3.46	34.99	0.85	45.41	3.61	86.26	0.21	85.03	0.86
0.15	5.31	1.01	7.95	1.14	16.32	0.81	16.31	2.29	72.33	0.39	71.30	0.83
0.25	4.41	0.94	5.27	1.12	10.04	0.85	10.04	2.09	60.75	0.52	59.93	0.94
0.35	4.09	0.89	4.29	1.11	7.18	1.02	7.18	2.02	51.79	0.61	51.12	1.00
0.45	3.94	0.84	3.83	1.10	5.58	1.24	5.58	1.97	44.79	0.67	44.25	1.04
0.55	3.86	0.81	3.59	1.11	4.55	1.49	4.55	1.94	39.24	0.71	38.80	1.06
0.65	3.81	0.79	3.43	1.10	3.90	1.73	3.84	1.91	34.78	0.74	34.42	1.08
0.75	3.77	0.79	3.30	1.10	3.58	1.99	3.32	1.88	31.13	0.76	30.83	1.07
0.85	3.74	0.80	3.20	1.09	3.41	2.24	2.92	1.87	28.11	0.77	27.86	1.07
0.95	3.70	0.83	3.12	1.09	3.33	2.50	2.61	1.86	25.58	0.79	25.37	1.06

Table 2.9: IAE and TV for setpoint change and disturbance. $\tau_0 = 1$, $k = 1$, $\theta = 0.5$, $\tau_c = \theta$.

2.5 Conclusion

In this section tuning-rules has been proposed for processes on the form

$$g(s) = \frac{k}{(\tau_1 s + 1)(\tau_0^2 s^2 + 2\tau_0 \zeta s + 1)e^{-\theta s}}.$$

Three different sets of tuning-rules have been proposed for various combinations of the process parameters $(k, \tau_1, \tau_0, \zeta, \theta)$. When $\tau_0 < \tau_1$ we have two sets of rules, depending on the amplitude of the peak gain. When this peak is higher than the low-frequency gain we get a I-controller, and when the peak is below the low-frequency gain we get a PI-controller. The rules are to reduce the model to a pure time-delay and first order plus time-delay respectivly, and then use the SIMC-rules to tune the (P)I controller.

When $\tau_0 >> \tau_1$ we derived the "interpolation rule", which gives a PID controller. The rule is based upon direct synthesis.

Various methods for determining the performances of the controllers have been used in this section. They range from multi-objective optimization (page 14) to comparison with benchmark tuning methods such as Ziegler-Nichols and Thyreus-Luyben.

There is a gap in the tuningrules when $\tau_0 > \tau_1$, that is when τ_0 grows larger than τ_1 from below. This set of processes has not been considered, but a naive solution would be to use the half-rule [6] and increase the delay by $0.5\tau_1$ and the second-order part by $0.5\tau_1$.

2.6 Recommendations to further work

All of the tuningrules seemed to work nicely with respect to the performace indicators used. However, more work is needed, for instance:

- Introduce P-action in the pure I-controller for `Category B`-processes.
- Improve the phase-rule to cover a wider range of (τ_0, τ_1), so we get the same rule for all cases when $\tau_0 < \tau_1$.
- Determine how to choose τ_c when there is little delay for the pure underdamped process (`Category A`).
- Test the rules on a broader class of processes.

Chapter 3

Unstable processes

3.1 Unstable processes. Direct Synthesis

The following derivation is based upon Skogestad [5]

Consider the process

$$g(s) = \frac{g_0(s)}{s-a}, \quad a > 0 \tag{3.1}$$

Since the PID controller is unable to cancel a right-hand-side pole (RHP), we need

$$l(s) = g(s)c(s) = \frac{l_0(s)}{s-a}, \quad \text{i.e. } c(s) = l_0 \cdot g_0^{-1} \tag{3.2}$$

Closed loop set-point response will now be

$$\frac{y}{y_s} = \frac{gc}{1+gc} = \frac{l_0(s)}{s-a+l_0(s)} \tag{3.3}$$

We want to use integral action in the controller, so for a first attempt we may try

$$l_0 = \frac{K_0}{s} \tag{3.4}$$

We then get that

$$\frac{y}{y_s} = \frac{K_0}{s(s-a)+K} = \frac{K_0}{s^2 - as + K_0} \tag{3.5}$$

By inspection we see that the process transferfunction from y_s to y, equation (3.5), will always be unstable. We need something to lift the phase before the instability occurs. We therefore try

$$l_0 = \frac{K_0}{s}\left(Ts+1\right) \tag{3.6}$$

By substitution into (3.2) we get

$$\frac{y}{y_s} = \frac{K_0\left(Ts+1\right)}{s(s-a)+K_0\left(Ts+1\right)} = \frac{K_0\left(Ts+1\right)}{s^2+\left(K_0T-a\right)s+1} = \frac{Ts+1}{\frac{1}{K_0}s^2+\left(T-\frac{a}{K_0}\right)s+1} \tag{3.7}$$

Assume we want a denominator on the form $(\tau_{c1}s+1)(\tau_{c2}s+1) = \tau_{c1}\tau_{c2}s^2 + (\tau_{c1}+\tau_{c2})s + 1$. Solving for K_0 and T we get

$$K_0 = \frac{1}{\tau_{c1}\tau_{c2}}, \quad T = \tau_{c1} + \tau_{c2} + a\tau_{c1}\tau_{c2} \tag{3.8}$$

If the process was $g(s) = k'/(s-a)$ this would correspond to a PI-controller with

$$K_c = \frac{1}{k'}\frac{\tau_I}{\tau_{c1}\tau_{c2}} \tag{3.9}$$

$$\tau_I = T = \tau_{c1} + \tau_{c2} + a\tau_{c1}\tau_{c2} \tag{3.10}$$

Note that if we want very fast control, i.e. τ_{c1} and τ_{c2} small, $K_c \to \infty$ and $\tau_I \to 0$.

3.1.1 Unstable process with delay

Now let us consider the process

$$g(s) = k'\frac{e^{-\theta s}}{s-a} \tag{3.11}$$

As a first attempt we will consider pure P-control, i.e. $g(s) = K_c$. We approximate the delay by $e^{-\theta s} = 1 - \theta s$. The closed loop polynominal $1 + gc = 0$ becomes

$$(s-a) + k'K_c(1-\theta s) = 0 \tag{3.12}$$

$$\underbrace{\left(1 - k'K_c\theta\right)}_{>0} s + \underbrace{k'K_c - a}_{>0} = 0 \tag{3.13}$$

For stability we require that the coefficients in the polynomial are positive, i.e.

$$k'K_c < \frac{1}{\theta} \qquad \text{and} \qquad k'Kc > a \tag{3.14}$$

$$\text{i.e.} \quad K_c < \tfrac{1}{k'}\tfrac{1}{\theta} \tag{3.15}$$

$$\text{and} \quad K_c > \tfrac{a}{k'} \tag{3.16}$$

To fulfill both of these constraints simultaneously we need a process with the property that

$$a\theta < 1 \tag{3.17}$$

We may say that "the system should not go unstable before we have time to act on it". In practise, we need $a\theta < 0.25$ to get a reasonable good reponse.

What is a reasonable tuning?

- P-part Unstable processes needs to be counteracted with a high controller gain, so we set this value to the maximum value recommended by the SIMC rule for a stable process with time delay:

$$K_c = \frac{0.5}{k'}\frac{1}{\theta} \tag{3.18}$$

  We see that this is half the gain as given in (3.14), and it is the same value as a integrating process with delay

- I-part Let us compare two expressions:

$$\text{Pure integrating process } (a\theta \ll 1): \quad \tau_I = 8\theta$$
$$\text{Pure unstable process: } \quad \tau_I = \tau_{c1}\tau_{c2} + a\tau_{c1}\tau_{c2}$$

If we choose to set $\tau_{c1} = \tau_{c2}$ we get[1]

$$\tau_{c1} = \tau_{c2} = \frac{4\theta}{1-2a\theta}$$

Which gives an I-part in the controller as

$$\tau_I = \tau_c\left(2 + a\tau_c\right) = \frac{4\theta}{1-2a\theta}\left(2 + \frac{a\theta}{1-2a\theta}\right) \tag{3.19}$$

Observe that when $a \to 0$, $\tau_I \to 8\theta$.

As an alternative we may also express τ_I as

$$\tau_{Ia} = 8 \cdot \theta \cdot f \tag{3.20}$$

$$f = \frac{1 + \frac{a\theta/2}{1-2a\theta}}{1-2a\theta} = \frac{1 + \frac{a\theta}{2(1-2a\theta)}}{1-2a\theta} \tag{3.21}$$

$$f = \begin{cases} 1 & \text{for } a\theta = 0 \\ 1.33 & \text{for } a\theta = 0.1 \\ 1.95 & \text{for } a\theta = 0.2 \\ 2.5 & \text{for } a\theta = 0.25 \\ 10 & \text{for } a\theta = 0.4 \\ \infty & \text{for } a\theta = 0.5 \end{cases} \tag{3.22}$$

In the above derivation we made two significant assumptions, the first that the gain should be choosen to be the maximum gain allowed with time-delay[2]. The other assumption was to set $\tau_{c1} = \tau_{c2}$. The first assumption has already been discussed. Setting $\tau_{c1} = \tau_{c2}$ may be a bit optimistic, since we loose one degree of freedom by doing this. Given that unstable processes are difficult to control, we should maybe try to search a larger area of paramaters than only along the line $\tau_{c1} = \tau_{c2}$, so this is something that needs to be investigated later[3].

3.2 A method based on empirical fitting

Sree et al. [8] gives tuning rules for unstable systems based upon emperical fitting of simulations of unstable systems with PID controllers implemented. For PID control these rules have an "operating

[1] Proof: We have that $K_c^{\text{pure unstable}} = \frac{1}{k'}\frac{\tau_I}{\tau_c^2}$ and $K_c^{\text{max allowed with delay}} = \frac{0.5}{k'}\frac{1}{\theta}$. For the pure unstable proces we also found that $\tau_I = \tau_{c1} + \tau_{c2} + a\tau_{c1}\tau_{c2}$. Combining these expressions we observe that $K_c^{\text{pure unstable}} = K_c^{\text{max allowed with delay}} \Rightarrow \frac{1}{k'}\frac{\tau_{c1}+\tau_{c2}+a\tau_{c1}\tau_{c2}}{\tau_c^2} = \frac{0.5}{k'}\frac{1}{\theta}$. If we set $\tau_{c1} = \tau_{c2} = \tau_c$ we obtain $2\frac{1}{\tau_c} + a = \frac{1}{2\theta} \Rightarrow \tau_c = \frac{4\theta}{1-2a\theta}, \quad a\theta \leq \frac{1}{2}$.

[2] Same as setting $\tau_c = \theta$ in the original IMC-rule, where we have that $K_c = \frac{1}{k'}\frac{1}{\tau_c+\theta}$, see [6, equation (19), page 91].

[3] Actually we initially have three parameters, τ_c in the original IMC-rule, and τ_{c1} and τ_{c2}. But linear dependence needs to be checked to find out how many degrees of freedom we have.

area" of $0.01 \leq \epsilon \leq 0.9$, where $\epsilon = \tau_d/\tau = \tau_d a = a\theta$ for the process model $g = k_p \exp -\tau_d s/(\tau s - 1) = (k_p/\tau)\exp -\theta s/(s-a)$, $\tau_d = \theta$ and $a = 1/\tau$. For PI action only their rules "is operating" for $0.01 \leq \epsilon \leq 0.6$ This agrees rather well with the above derivation with respect to what kind of processes may be controlled with a PI(D) controller.

Huang and Chen [1] proposes a two-degree of freedom control structure that can handle $0 \leq a\theta \leq 2$ by using two PID-controllers, but as this report is on tuning and not structure, this will not be considered here.

3.3 Comparing the methods of Sree et al. and the one based unpon direct synthesis

Let us first review the notation used by Sree et al. [8, pages 2202 and 2208]. We have that

$$\begin{aligned} g(s) &= k_p \frac{e^{-\tau_d s}}{\tau s - 1} &(3.23)\\ &= \underbrace{\frac{k_p}{\tau}}_{k'} \frac{e^{-\theta s}}{s - a} &(3.24)\end{aligned}$$

so,

$$k_p = k' \frac{1}{a} \quad (3.25)$$

$$\tau_d = \theta \quad (3.26)$$

$$a = 1/\tau \quad (3.27)$$

for a FOPTD process. Their tuning rules for a PI-controller on an unstable FOPTD process is with $\epsilon = a\theta$[4]:

$$K_c = \frac{a}{k'} 0.8624(a\theta)^{-0.9744} \approx \frac{0.8624}{k'} \frac{1}{\theta} \text{ for } 0.01 \leq a\theta \leq 0.6 \quad (3.28)$$

$$\tau_I = \frac{1}{a}\left(143.34(a\theta)^3 - 73.912(a\theta)^2 + 19.039(a\theta) - 0.2276\right) \text{ for } 0.01 \leq a\theta \leq 0.6 \quad (3.29)$$

Note From equation (3.29) it might seem like $\tau_I \to \infty$ when $a \to 0$, but the expression is bounded by the restriction on $a\theta$, that is, $0.01 \leq a\theta \leq 0.6$. Another problem is that the expression inside the parenthesis is negative for $(a\theta) < 0.0126$[5]. As we will see in the next section, problems arise when $a\theta$ is below 0.015. Note also the close resemblance between the IMC-rule for the gain and the rule of Sree et al. [8], the latter beeing found by minimization. It is a little higher than the IMC-rule for stable processes, but not higher than the (derived) maximum gain allowed for stability, see equation (3.14).

3.3.1 Simulations

With reference to figure A.1 page 45 we may simulate the tuning rules on an unstable system. As shown above, these simulations only make sense when $a\theta \leq 0.5$. 4 simulations were conducted, and their process

[4]Equation (6), page 2202 in [8]: $\epsilon = \frac{\tau_d}{\tau} = \frac{1}{\tau}\theta = a\theta$.

[5]Found by using the MATLAB-function `fsolve` on the expression.

conditions is shown in table 3.1. The two lower vaules for $a\theta$ (={0.015,0.014}) were chosen because these were the lowest vaules we could implement and still achieve a stable closed loop with the method of Sree et al. [8]. I have adopted the notion of “servo problem” and “regulatory problem” from Sree et. al. Here the servo problem is simulated by conducting a unit step in the setpoint, while the regulatory problem is simulated by a unit step in the disturbance. See figure A.1.

Figure/Table	a	θ	$a\theta$
3.1/3.2	1	0.015	0.015
3.2/3.3	1	0.25	0.25
3.3/3.4	0.014	1	0.014
3.4/3.5	0.25	1	0.25

Table 3.1: Process conditions for simulations of the unstable system $g(s) = \frac{k}{s-a}e^{-\theta s}$ with references to corresponding figures and tables.

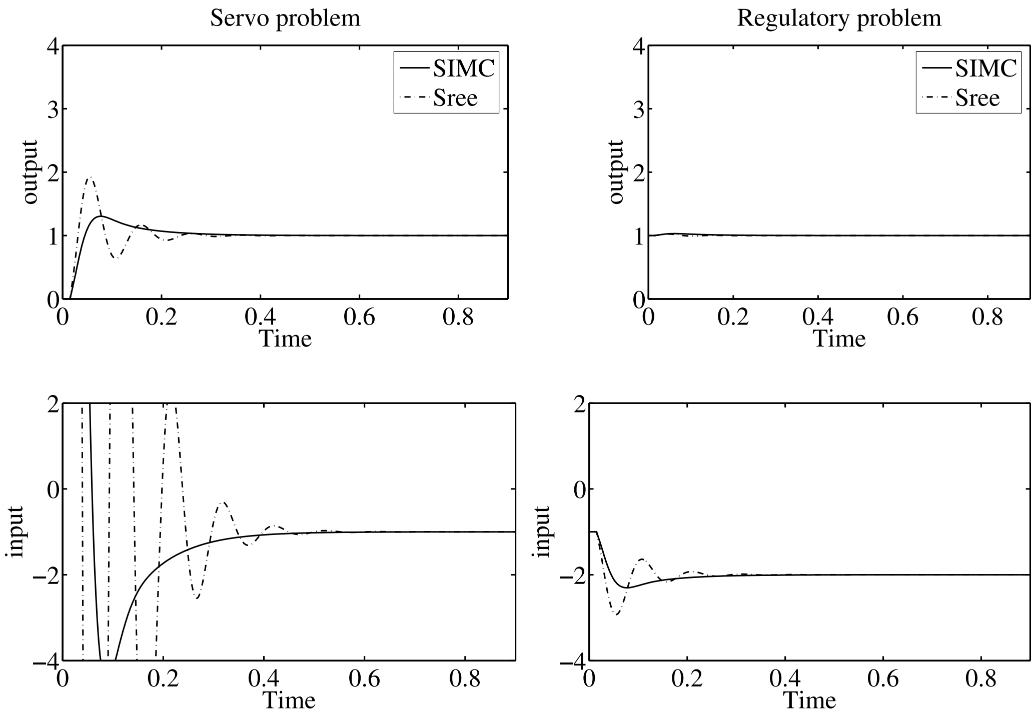

Figure 3.1: $g(s) = \frac{1}{s-1}e^{-0.015s}$

3.3.2 Discussion

First of all, note that the simulations were conducted at the boundaries for the area in which the tuning is sensible, that is, we have $0.01 \leq a\theta \leq 0.6$ from Sree et al. [8]. Further we know that in practise only systems with $a\theta \leq 0.25$ is likely to be controllable with a single PI controller. We therefore choose to perform the simulations at $(a, \theta) = \{(1, 0.015), (1, 0.25), (0.014, 1), (0.25, 1)\}$, which was the closest we could move in the lower $a\theta$ area and still have a stable closed loop with the method of Sree et. al.

Method	IAE, Setpoint	TV, Setpoint	IAE, disturbance	TV, disturbance
SIMC	0.06	49.31	0.0	1.6
Sree	0.08	230.37	0.0	4.2

Method	GM_L	GM	PM	M_s	M_T	K_c	τ_I
SIMC	0.03	2.95	45.72	1.7	1.3	33.3	0.1
Sree	0.03	1.60	17.68	3.9	3.4	51.6	0.0

Table 3.2: $g(s) = \frac{1}{s-1}e^{-0.015s}$, $c(s) = K_c\left(1 + \frac{1}{\tau_I s}\right)$.

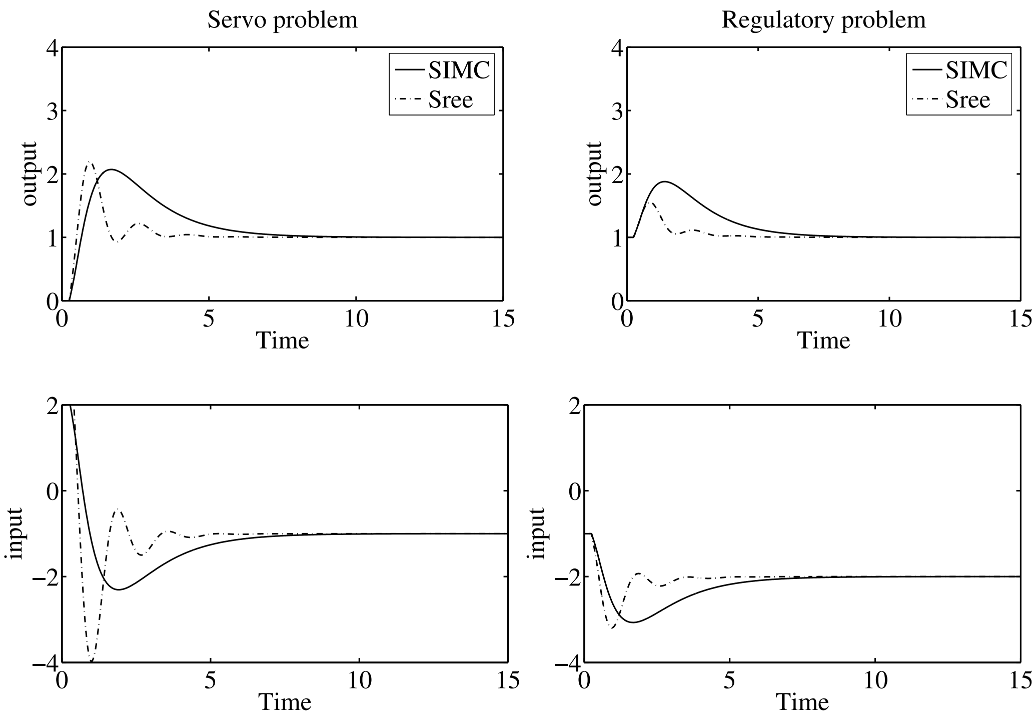

Figure 3.2: $g(s) = \frac{1}{s-1}e^{-0.25s}$

Method	IAE, Setpoint	TV, Setpoint	IAE, disturbance	TV, disturbance
SIMC	3.43	5.81	2.5	3.1
Sree	1.45	13.52	0.6	4.0

Method	GM_L	GM	PM	M_s	M_T	K_c	τ_I
SIMC	0.53	2.75	28.66	2.1	2.4	2.0	5.0
Sree	0.35	1.57	18.47	3.8	3.3	3.3	2.2

Table 3.3: $g(s) = \frac{1}{s-1}e^{-0.25s}$, $c(s) = K_c\left(1 + \frac{1}{\tau_I s}\right)$.

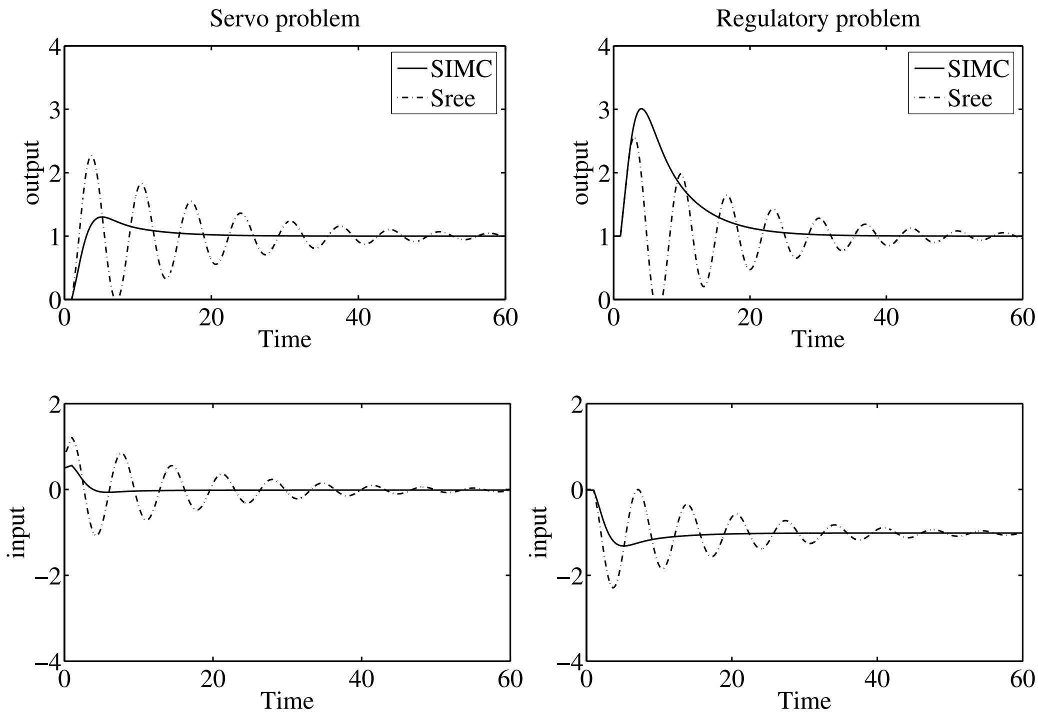

Figure 3.3: $g(s) = \frac{1}{s-0.014}e^{-s}$

Method	IAE, Setpoint	TV, Setpoint	IAE, disturbance	TV, disturbance
SIMC	4.14	0.74	16.6	1.6
Sree	15.60	12.65	17.1	14.0

Method	$\mathrm{GM_L}$	GM	PM	$\mathrm{M_s}$	$\mathrm{M_T}$	K_c	τ_I
SIMC	0.03	2.95	45.80	1.7	1.3	0.5	8.3
Sree	0.04	1.23	5.24	11.8	11.5	0.8	1.8

Table 3.4: $g(s) = \frac{1}{s-0.014}e^{-s}$, $c(s) = K_c\left(1 + \frac{1}{\tau_I s}\right)$.

Method	IAE, Setpoint	TV, Setpoint	IAE, disturbance	TV, disturbance
SIMC	13.72	1.45	40.0	3.1
Sree	5.79	3.38	10.3	4.0

Method	$\mathrm{GM_L}$	GM	PM	$\mathrm{M_s}$	$\mathrm{M_T}$	K_c	τ_I
SIMC	0.53	2.75	28.66	2.1	2.4	0.5	20.0
Sree	0.35	1.57	18.47	3.8	3.3	0.8	8.6

Table 3.5: $g(s) = \frac{1}{s-0.25}e^{-s}$, $c(s) = K_c\left(1 + \frac{1}{\tau_I s}\right)$.

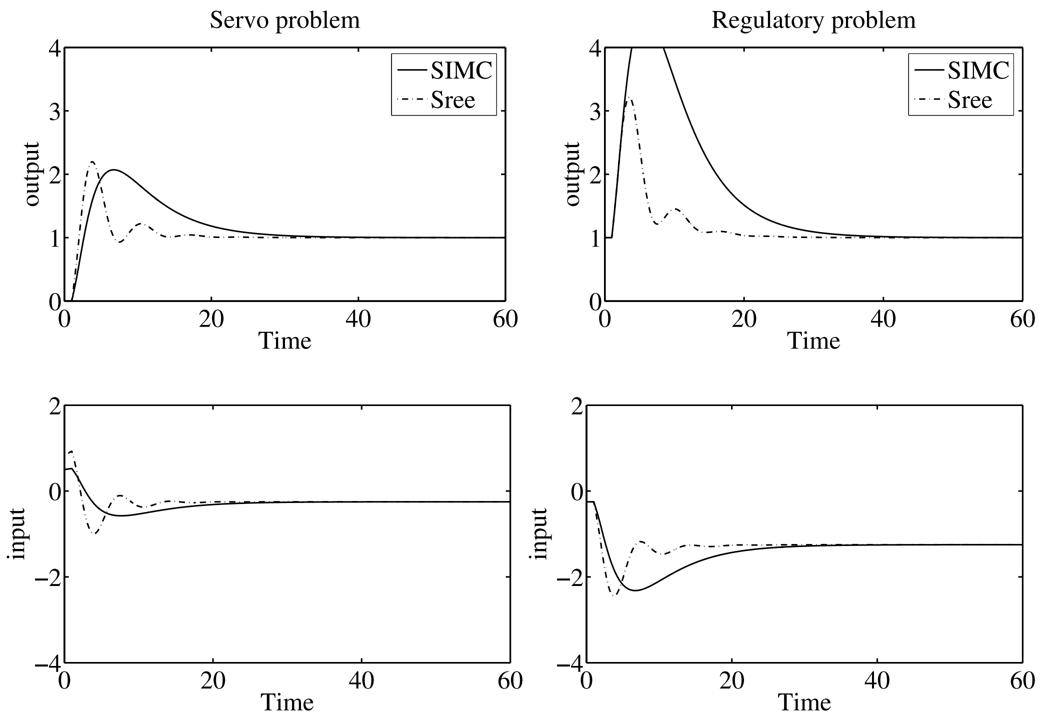

Figure 3.4: $g(s) = \frac{1}{s-0.25}e^{-s}$

One immediately observers that as either a or θ approaches zero, the metod of Sree et. al. is poor with respect to all measures, while the SIMC method[6] is good in both cases. When $a \to 0$ the SIMC-method approaches the SIMC-tuning for an integrating process with delay, while when $\theta \to 0$, it approaches the pure unstable tuning with $\tau_{c1} = \tau_{c2}$. At the two other edges of the valid tuning area the two methods are more comperable. As already commented, when $(a\theta)$ approaches 0.01, which is the lower validity-boundary for the Sree-tuning, their expression for τ_I goes to zero, and actually a bit under zero, so I suspect that if they had been more conservative with respect to the interval in which the tuning rules were valid, these problems would be avoided. I could not find exactly for which data set they had fitted their tuning-parameters to, but it seems at least that if they want to cover their present range for $a\theta$ they should put more emphasis (weight in a given objective function) on the boundary case when $a\theta \to 0.01$.

Consider now tables 3.3 and 3.4. I was a bit alarmed when I observed that the frequency-domain measures for the cases of $g(s) = e^{-0.25s}/(s-1)$ and $g(s) = e^{-s}/(s-0.25)$ seems to be almost equal. Look at the loop transferfunctions for the two cases, shown i figures 3.5 and 3.6, correspondingly. We observe that in the frequency domain the L's are shifted along the ω-axis, but all the margins (S and T not shown) are almost equal. The one measure we are missing to distinguish the two cases is the closed-loop bandwidth, $\omega_{\text{bandwidth}}$, which surely is different in the two cases. We observe that, as expected, the bandwidth for the process with more delay is the one with least bandwidth.

By looking at all the cases one obsereves that the (usual) gain margin (GM) and phase margin (PM) for the SIMC-tuned controllers are all higher that the corresponing values for the Sree-tuned ones. For the two comparable cases (tables 3.5 and 3.3) we see that there is a factor of about 2/3 between the GM and PM vaules. However, the lower gain margin is better for the method of Sree et. al., as we want this to be

[6] I choose to label the direct synthesis method SIMC because it is based upon a model (IMC) and we choose to use $\tau_c = \theta$ when we selected the controller gain.

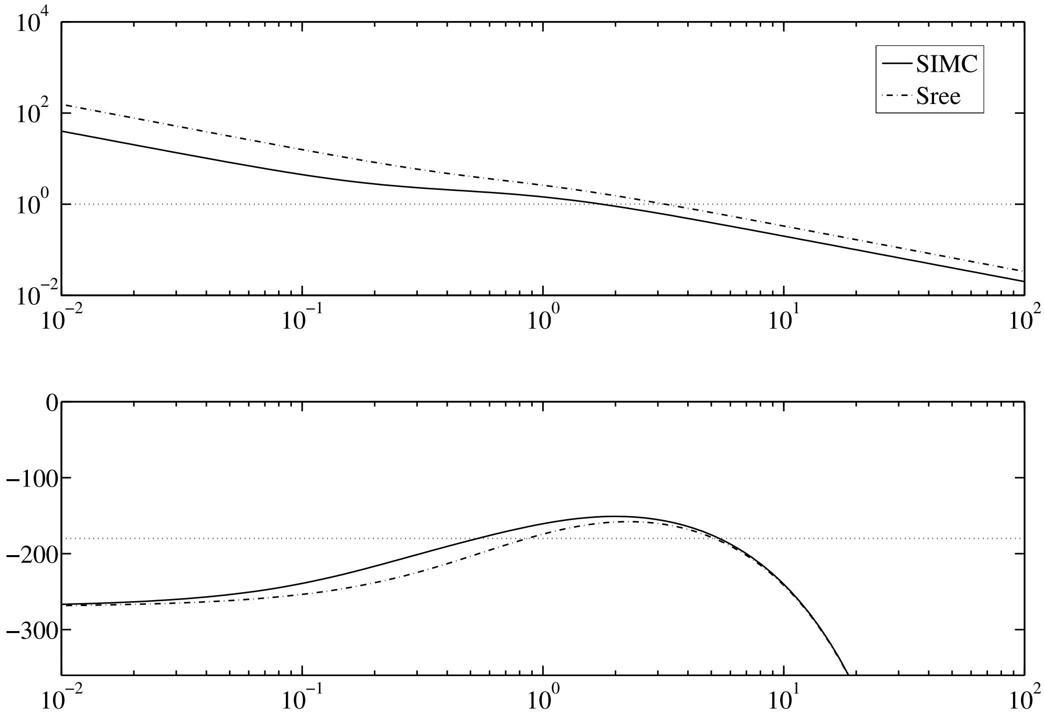

Figure 3.5: $L(s) = g(s)c(s)$ for $g(s) = \frac{1}{s-1}e^{-0.25s}$.

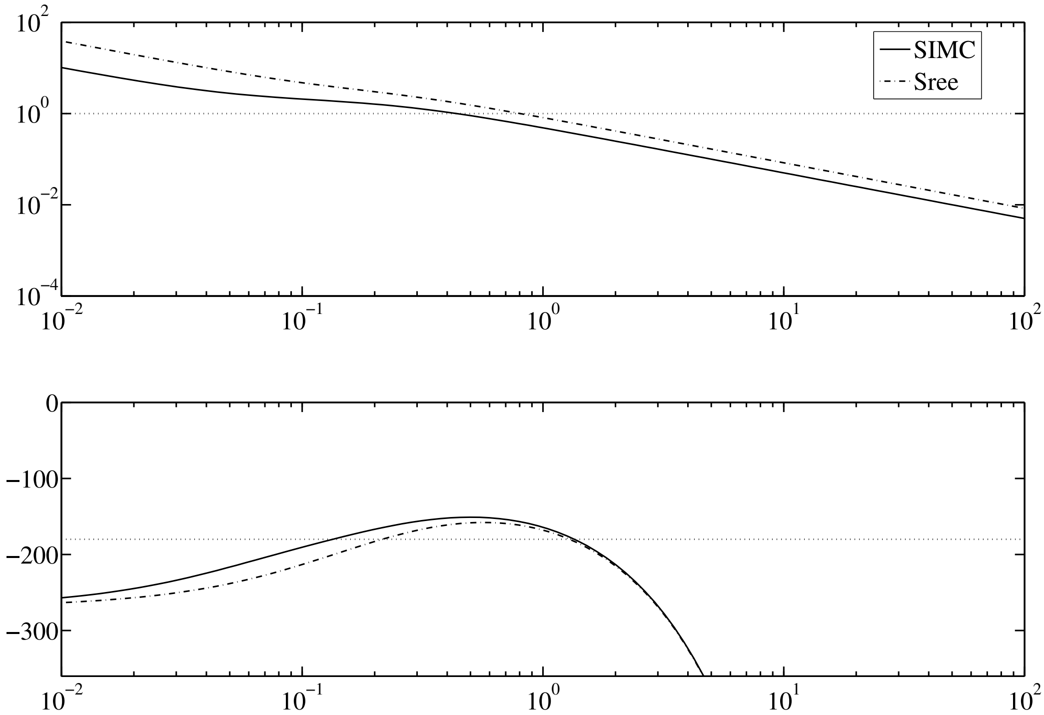

Figure 3.6: $L(s) = g(s)c(s)$ for $g(s) = \frac{1}{s-0.25}e^{-s}$.

as low as possible. (See appendix B for a review of the different performance and robustness measures). The lower gain margin of 0.53 for the SIMC-tuned controller is likely to be a bit high, as we would like to at least have it below 0.5[7].

By inspection of the time-domain plots in figures 3.4 and 3.2 one sees that the SIMC-tuned controller is slower and smoother than the corresponding Sree-controller. We observe that the Sree-controller is rather oscillatory in the transient response, and the input-usage is higher. This is of course expected as the GM and PM are lower and the M_S and M_T is higher for the Sree-controller. In this example one also observes the common "rule" that a high TV and low IAE usually corresponds to a less robust controller than a response with low TV and high IAE, so the TV measure is in fact a good measure for robustness. The practial problem is maybe that one has to compare it with another controller[8], since it will be relative to the other controller, whereas the usual frequency-domain measures such as PM and GM are more independent of comparison with a reference controller.

Summary

This section was meant to assess if the proposed tuning method is a good one or not. Unfortunaltely the present conculsion is inconclusive. We have tried to compare it to a (new) method found in litterature, but there is not much to say, besides that SIMC-tuning is better than the method for in litterature for the limiting cases as $a \to 0$ and $\theta \to 0$. When a and θ are mutually non-zero, the SIMC-tuning seemed to be more robust than the other method, but one needs to define more specifically what is desirable in the case of unstable processes. We also observed that when $a\theta = 0.25$ the lower gain margin of the present method possibly to high.

3.4 Recommendations to further work

Here I will list some ideas to further work on this subject:

- When deriving the tuning-rules we selected $\tau_{c1} = \tau_{c2}$. Since we have two degrees of freedom, we could maybe try to find a closed loop characteristic polynomial $l_{\mathrm{cl}}(s)$. With the two degrees of freedom we could specify two properties of this polynomial, for instance damping and time-constant. This would yield another expression for τ_I.
- We could investigate how an optimal controller with respect to some performance and robustness measures would select the tuning paramteres τ_{c1} and τ_{c2}, and possibly get some clues to how we can improve the model. By optimal controller I mean for instance to make a plot such as my naive method in chapter 2.2.6, or maybe better, to use the method proposed in Kristiansson and Lennartson [2]. They use three criteria, with two of them as inequality constraints in the objective funtion, while the last one is the objective function itself. As one example, they proposes to solve

[7] A rule-of-thumb is that the lower gain margin should be lower than 1/2 to 1/4.
[8] Or normalize it somehow such that is may be used as a general measure.

the following constrained optimization problem:

$$\begin{aligned}
&\min_{\rho} J_v(\rho) \\
&\text{subject to} \\
\mathrm{GM_s}(\rho) &\leq C_1 \\
J_u(\rho) &\leq C_2
\end{aligned}$$

where ρ is the vector of tuning-paramteres. Interesstingly, they note that it is worthwhile to include a filter constant for the D-filter in this vector. The other expressions are defined as:

$$\begin{aligned}
\mathrm{GM_s} &= \max\left(||S||_\infty, \alpha||T||_\infty\right) \\
J_u &= \left|\left|\frac{c(s)}{1+g(s)c(s)}\right|\right|_\infty \\
J_v &= \left|\left|\frac{1}{s}\frac{g(s)}{1+c(s)g(s)}\right|\right|_\infty \\
||\circ||_\infty &= \max_{\omega} |\circ(j\omega)|
\end{aligned}$$

where the last equation defines the norm $||\circ||_\infty$. I think α is a tuning parameter, in my opinion the paper was not clear on this point.

As one observes, the main difference between the simple method used in chapter 2.2.6 and the method of Kristiansson and Lennartson [2] is that the first one uses time-domain information in objective function and constraints, that is, we *simulate the system in time-domain*, whereas the latter method uses frequency-information. Intuitively, the latter method has two main advantages:

- Whereas the time-domain approach with unit steps in setpoint and disturbance only exites the low frequencies of the system[9], the method of Kristiansson and Lennartson [2] covers a broader range of frequencies.
- In the time-domain method one has to simulate the system with for instance SIMULINK. If the optimizer goes with large steps in for example D-action the simulation will quite easily fail to converge. In the frequency-approach this problem is eliminated, so the frequency-based approach seems easier to use in practise.

- Another obvious way to improve this discussion would be to compare the proposed method to more methods found in litterature. As mentioned above the comparison with the method of Sree et al. [8] was inconclusive.

[9] Unit step is $1/s$ in Laplace domain, which has high gain at low frequencies and low gain at high frequencies.

Chapter 4

Overall discussion, conclusion and recommendations

4.1 Discussion

This work that has been conducted from September to November 2005 turned more into a *indentification of research needs*, rather than pure research. I have discovered that in order to do work on a scientific field one has to have a lot of insight into what the problem really is. Further, one has to learn how to use and understand the various scientific tools. For me, this was for instance to get more insight into frequency-domain of linear systems. When I was thought this subject two years ago I found it quite easy to understand and did not identify much problems with it. However, to use a mathematical tool such as frequency-analysis, one needs to develop a gut feeling, or intuition, for how it's properties and what conculsion one can draw from using this tool. Furthermore, I assume that this is a peak one has to climb in any research related work before one can start to do some "serious buisness".

The conducted work has focused on tuning rules for systems with complex poles on the left-hand-side of the complex axis and systems with a single real pole on the right-hand-side of the complex axis. For the case with stable but complex poles I tried to divide the systems into three categories and develop tuning rules for these cases independently, but at the same time focusing on continuity between the rules. A first attempt on this was made, but still there is quite some work to be done before we have a good set of rules with the desired properties. For the case of an underdamped system with a first order filter a way to reduce the model to either a pure time-delay or a first-order plus time-delay model was proposed. We then investigated performance when using the existing SIMC-rules for stable processes. For the case of a pure underdamped system with no other apparent dynamics a review of work conducted by prof. Skogestad was performed. The resulting rules looked promising compared to the well-known Ziegler-Nichols and Tyreus-Luyben rules.

As for the case of a single real pole in the right half plane, a review of work conduced by professor Skogestad was done and a comaprison with a set of tuning-rules in a rather new (2004) paper in these kind of processes. However, the comparison was not conculsive, and we only manged to get an overview of the problem and discover how we need to work to develop the rules further.

Some various methods for evaluating the performance and robustness of the tuning-rules has been used in this report. They include

- Inspection of the loop transferfunction $L = gc$ and it's properties in the frequency domain. Issues such as roll-off, slope around bandwidth frequency and gain at low frequency were discussed.
- Time-domain evaulation by inspection of reponses after setpoint changes and disturbances.
- Comparison with well-known tuning-methos such as Ziegler-Nichols and Tyreus-Lubyen.
- Comparison with optimal controller with repsect to IAE and TV.
- Comparison with methods found in litterature (Sree et al. [8]).

4.2 Recommendations to further work

Throughout this report quite a lot of recommendations to further work has been made. I will now summarize the most important issues:

- Improve the rule for phase-approximation for the systems with $\tau_1 \geq \tau_0$ such that it covers all cases, for the process $g(s) = k/\left((\tau_1 s + 1)(\tau_0^2 s^2 + 2\tau_0\zeta s + 1)\right) e^{-\theta s}$.
- Try to get P-action in `Category B`-processes, that is the processes with $\tau_1 \geq \tau_0$ and a clear resonant peak.
- Investigate how to choose the tuning parameter τ_c for pure oscillatory processes, that is, $g(s) = k/(\tau_0^2 s^2 + 2\tau_0\zeta s + 1)e^{-\theta s}$, $0 \leq \zeta \leq 0.707$.
- Look at how to choose the tuning paramteres τ_{c1} and τ_{c2} for the case of a process with one real pole in the right half plane.
- Improve the comparison method with the optimal controller and do a closer comparison with the method of Kristiansson and Lennartson [2].

4.3 Conclusion

A start on the work of "Extensions of Skogestad's SIMC tuning rules to oscillatory and unstable processes" has been conducted. Processes with either a pair of complex poles in the left half plane, or a single real pole in the right half plane have been investigated. Tuning rules have been proposed based on either model reduction to a stable process or by direct synthesis. For all the covered processes the results seems promising, but at the same time they all share the need for further review.

A method for comparing the resulting PID-controller with an optimal controller with respect to integrated square error and total input variation has been prosed. We have observed that this method needs to be improved before it becomes practically applicable.

There is still work to be done before a complete set of satisfactory tuning-rules exists. This report may be used as a starting-point for further work on this topic.

Henrik Manum, Trondheim, December 19, 2005

Nomenclature

Here are some of the most used symbols in the report. See also table B.1, page 49 and the definitions on page 50.

Symbol	Description
τ	Time constant, [time]
θ	Delay, [time]
ω	Frequency, [rad/s]
$c(s)$	Controller equation
k	Gain
$g(s)$	Process equation
s	Laplace-domain variable [time^{-1}]

Bibliography

[1] H.-P. Huang and C.-C. Chen. Control-system synthesis for open-loop unstable process with time delay. *IEE Proc.-Control Theory Appl.*, 144(4), July 1997.

[2] Birgitta Kristiansson and Bengt Lennartson. Evaulation and simple tuning of pid controllers with high-frequency robustness. *Journal of Process Control*, 16:91–102, 2006.

[3] Katsuhiko Ogata. *Modern Control Engineering*. Prentice-Hall International Editions, 1990.

[4] Dale E. Seborg, Thomas F. Edgar, and Duncan A. Mellichamp. *Process Dynamics and Control*. John Wiley and Sons, second edition, 2003.

[5] Sigurd Skogestad. Tillegg til fag tkp4140: Prosessregulering. kopi av transparenter, øvinger etc. Institutt for Kjemsik Prosessteknologi, September 2000.

[6] Sigurd Skogestad. Simple analytic rules for model reduction and pid controller tuning. *Journal of Process Control*, 13:291–309, 2003.

[7] Sigurd Skogestad and Ian Postlethwaite. *Mulitivariable Feedback Control, Analysis and Design*. John Wiley and Sons, Ltd, second edition, 2005.

[8] R. Padma Sree, M. N. Srinivas, and M. Chidambaram. A simple method of tuning pid controllers for stable and unstable foptd systems. *Computers and Chemical Engineering*, 28:2201–2218, 2004.

Appendix A

Simulink models

For simulation in time-domain SIMULINK together with MATLAB was used to solve the simulation problem. Three different models were needed. First, a model for the cascade form PID-controller, showed in figure A.1. Further, a model for the ideal form PID-controller is showed in figure A.2. Last, we needed a model for the pure I-controller, which is displayed in figure A.3.

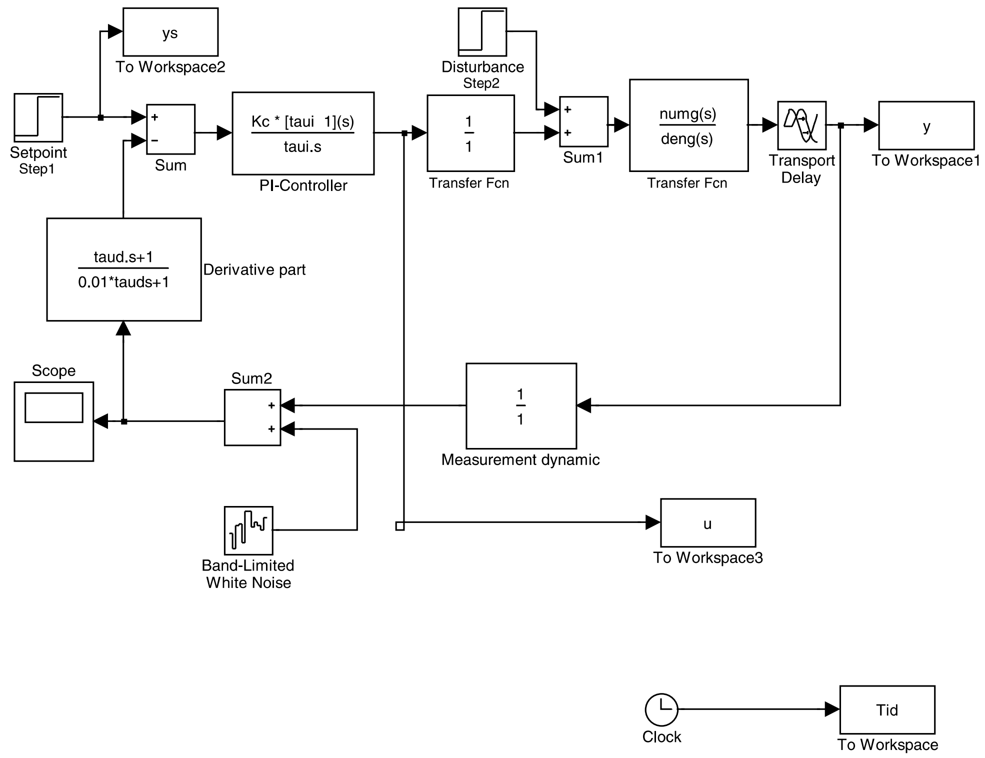

Figure A.1: SIMULINK-model for cascade form PID-controller

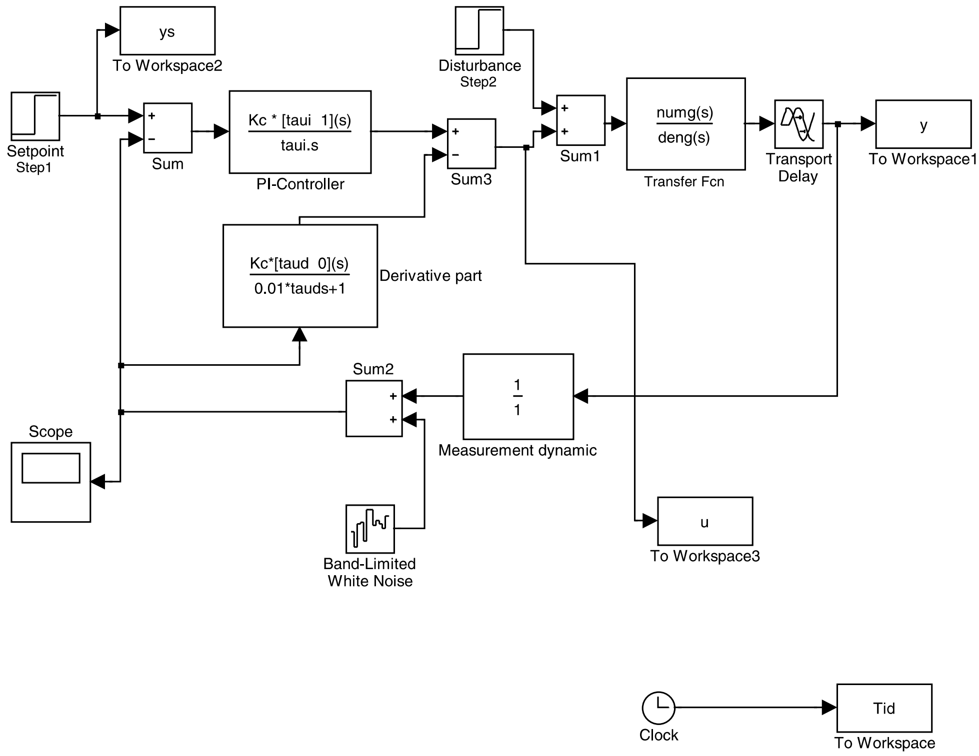

Figure A.2: SIMULINK-model for ideal form PID-controller

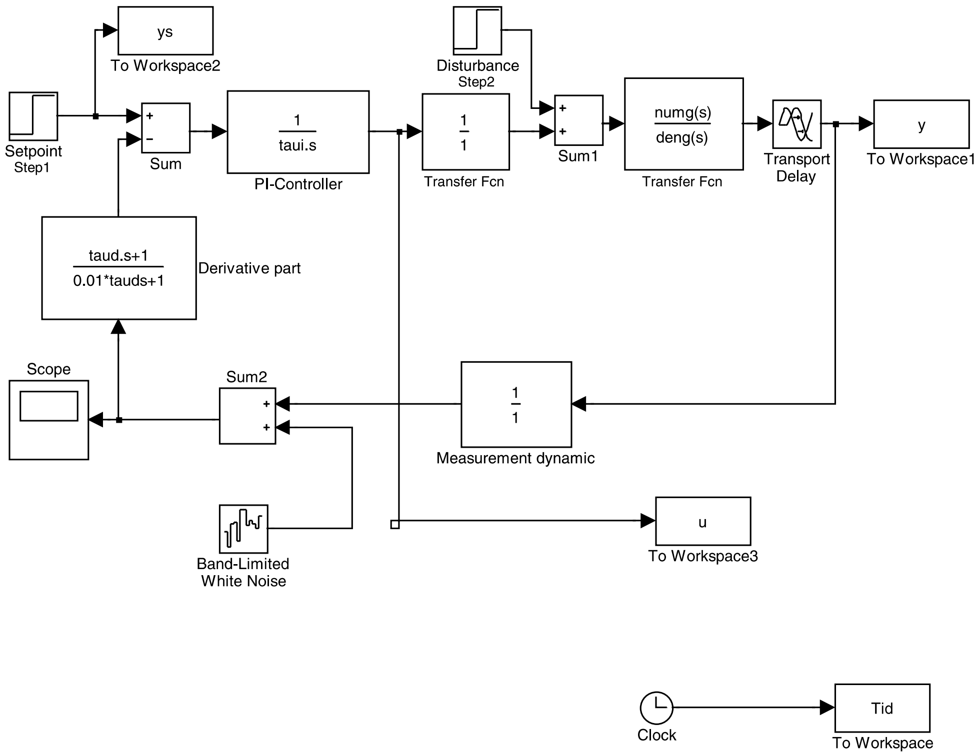

Figure A.3: SIMULINK-model a pure I-controller

Appendix B

Definitions

This chapter will cover a brief review of the basic measures used for evaluation of performance and robustness. All of this is taken from Skogestad and Postlethwaite [7].

B.1 Frequency domain performance

Consider the negative feedback scheme in figure B.1. We define the loop transfer function $L = gc$, where $G = g$ and $K = c$ in the single input single output (SISO) case. Further we define the sensitvity function $S = (I + L)^{-1} = 1/(1 + L)$ where the last equality holds for SISO-systems. The complementary sensitivity function is defined as $T = (I + L)^{-1}L = L/(L + 1)$ where again the last equality holds for SISO-systems, but generally not for multiple input multiple output (MIMO) systems. For performance and robustness

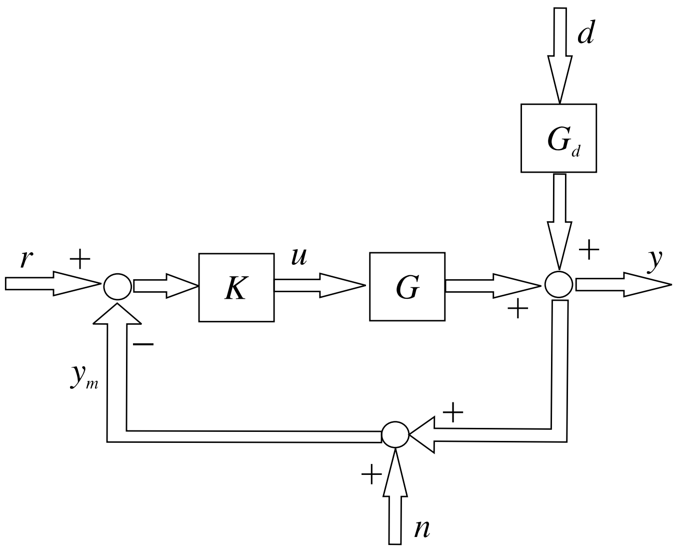

Figure B.1: Block diagram for negative feedback.

measures in the frequency domain may then be defined as in table B.1.

Measure	Definition	Description
GM	$1/\lvert L(j\omega_{180})\rvert$	**(Upper) gain margin.** The factor by which the loop gain $\lvert L(j\omega)\rvert$ may be increased before the closed-loop system becomes unstable. The GM is thus a direct safegard against steady-state gain uncertainty (error). The phase crossover frequency ω_{180} is where the Nyquist curve of $L(j\omega)$ crosses the negative real axis between -1 and 0.
$\mathrm{GM_L}$	$1/\lvert L(j\omega_{L180})\rvert$	**Lower gain margin.** The factor by which the loop gain $\lvert L(j\omega)\rvert$ may be *decreased* befor the closed-loop system becomes unstable. ω_{180L} is the frequency where the Nyquist curve of $L(j\omega)$ crosses the negative real axis between $-\infty$ and -1. If there is more than one such crossing, we take the closest crossing to -1.
PM	$\angle L(j\omega_c) + 180°$	**Phase margin.** The PM tells us how much more negative phase (phase lag) we can add to $L(s)$ at frequency ω_c before the phase at this frequency becomes $-180°$ which corresponds to closed-loop instability. ω_c is the frequency where $\lvert L(jw)\rvert$ crosses 1.
$\mathrm{M_S}$	$\max_\omega \lvert S(j\omega)\rvert$	**Maximum peak of the sensitivity function.** Measure for degraded control in the mid-frequnecy range and robustness measure. We want this to be close to 1.[a]
$\mathrm{M_T}$	$\max_\omega \lvert T(j\omega)\rvert$	**Maximum peak of the complementary sensitivity function.** A large value of $\mathrm{M_T}$ indicates poor performance as well as poor robustness, so we want this to be small.

[a] See the source, Skogestad and Postlethwaite [7], for more information about $\mathrm{M_S}$ and $\mathrm{M_T}$.

Table B.1: Frequency domain performance/robustness measures. Note: $j^2 = -1$ and ω [=] rad/sek.

B.2 Time domain performance

The two time domain measures used in this report are total variation (TV) and integrated absolute error (IAE). These are defined as:

$$\mathrm{TV}(v) = \sum_{i=1}^{\infty} |v_i| \tag{B.1}$$

$$\mathrm{IAE}(e) = \int_0^{\infty} |e(\tau)| \, \mathrm{d}\tau \tag{B.2}$$

Typically TV is used on the input signal u from the controller to the process, and $e(t)$ is usually $y_s(t) - y(t)$. IAE, which is defined for a contious signal above, may be computed for a discrete signal for instance with the trapeze method.

Generally a robust controller will have a low TV, as it moves the input less than a controller with a high TV. A controller with a low TV will in effect give a smooth, but possibly slow control, while a high-TV controller will give a fast but possibly less roubust control.

A low IAE value usually means that the controller is fast and has a good setpoint tracking and disturbance rejection, while a high IAE implies that the controller is acting slower. To achieve a low IAE one necessarily needs to use a lot of input, TV, so these so measures are conflicting.

Appendix C

Derivation of SIMC-rule for underdamped stable secondorder plus delay transferfunctions

This is taken entirely from Professor Skogestads work,
`http://www.nt.ntnu.no/users/skoge/publications/2003/tuningPID/more/extensions/oscillating.txt`.

> Note to the reader: This derivation is a bit tedious and it might be easy to "loose the tread". However, it's all more or less simple algebra, the point is just to keep track of the various expressions.

For a standard second order oscillation process (we neglect τ_1, assuming that only the oscillating part is dominant)

$$g(s) = k\frac{e^{-\theta s}}{\tau_0^2 s^2 + 2\tau_0\zeta s + 1} \tag{C.1}$$

direct synthesis (IMC) for setpoints gives the controller [6, page 90-91]

$$c(s) = (1/k)\frac{1}{(\tau_c + \theta)s}\left(\tau_0^2 s^2 + 2\tau_0\zeta s + 1\right) \tag{C.2}$$

Since this controller is derived for setpoints it would naturally give very good tracking, but we need to make sure that is it robust with respect to disturbance rejection.

Let us consider the limiting case of a double integrating process:

$$g(s) = \frac{1}{k''}\frac{e^{-\theta s}}{s^2}, \quad k'' = \frac{k}{\tau_0^2} \tag{C.3}$$

The double integrating process (C.3) is on the form (C.1) with

$$\tau_0 = \infty, \quad \tau_0\zeta = 0, \quad k'' = \frac{k}{\tau_0^2} \tag{C.4}$$

We observe that $k =$ "∞^2". For (C.3) the vcontroller (C.2) becomes

$$c(s) = (1/k'')\frac{1}{\tau_c + \theta}s \tag{C.5}$$

which is a pure D-controller with $K_c\tau_D = (1/k'')(\tau_c + \theta)$. This controller will *not* work well for an input (load) disturbance, s has as we know large gain at high frequencies, and these are the frequencies we want to reject by the controller.

In fact, for the double-integrating processes Skogestad [6, page 93, equations (26) and (27)] recommends the following tuning

$$K_c = \frac{1}{k''}\frac{1}{4(\tau_c + \theta)^2}, \quad \tau_D = 4(\tau_c + \theta), \quad \tau_I = 4(\tau_c + \theta) \tag{C.6}$$

where we observe the integral action has been added. (Note that these are for the cascade-form controller). Also note that the "D-term" for the controller in (C.6) is the same as just found, that is,

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

The main focus is now: Can we find a simple "correction" for integrating processes that includes also oscillating processes?

In the following we will work with a PID controller in the form:

$$c(s) = K_c' + K_I'/s + K_D's = K_c'(1 + 1/(\tau_I's) + \tau_D's) \tag{*}$$

C.1 IMC setpoint settings for general second-order processes

We are still looking at processes on the form (equation (C.1)):

$$g(s) = k\frac{e^{-\theta s}}{\tau_0^2 s^2 + 2\tau_0\zeta s + 1}$$

Controller (C.2) for process (C.1) corresponds to the following "ideal" settings:

$$\begin{aligned} K_c' &= \frac{1}{k}\frac{2\tau_0\zeta}{\tau_c + \theta} \\ \tau_I' &= 2\tau_0\zeta \\ \tau_D' &= \frac{1}{2}\frac{\tau_0}{\zeta} \end{aligned}$$

For the controller from in (*)

$$\begin{aligned} K_c' &= \frac{1}{k}\frac{2\tau_0\zeta}{\tau_c + \theta} \\ K_I' = K_c'/\tau_I' &= \frac{1}{k}\frac{1}{\tau_c + \theta} \\ K_D' = K_c'\tau_D' &= \frac{1}{k}\frac{\tau_0^2}{\tau_c + \theta} \end{aligned}$$

or in terms of k'':

$$K_c' = \frac{1}{k''}\frac{1}{\tau_c+\theta}\frac{2\zeta}{\tau_0} \tag{C.7a}$$

$$K_I' = \frac{1}{k''}\frac{1}{\tau_c+\theta}\frac{1}{\tau_0^2} \tag{C.7b}$$

$$K_D' = \frac{1}{k''}\frac{1}{\tau_c+\theta} \tag{C.7c}$$

Note: Double integrating processes where $\tau_0 = \infty$ and $\zeta = 0$ gives $K_c' = 0^2$ and $K_I' = 0^2$, i.e. both go to zero in second order.

C.2 SIMC-corrections for integrating overdamped processes ($\zeta > 1$)

Consider the process

$$g(s) = k\frac{e^{-\theta s}}{(\tau_1 s+1)(\tau_2 s+1)} \tag{C.8}$$

Comparing terms with (C.1) we see that (C.8) is on the form (C.1) with:

$$\tau_1\tau_2 = \tau_0^2$$
$$\tau_1+\tau_2 = 2\tau_0\zeta$$

Translation formulas to get τ_0 and ζ:

$$\tau_0 = \sqrt{\tau_1\tau_2}$$
$$\zeta = \frac{\tau_1+\tau_2}{2\sqrt{\tau_1\tau_2}}$$

Translation formulas to get to τ_1 and τ_2:

$$\tau_1 = \tau_0\left(\zeta+\sqrt{\zeta^2-1}\right)$$
$$\tau_2 = \frac{\tau_0}{\zeta+\sqrt{\zeta^2-1}}$$

From [6, Equations (23)-(25)] we have the following settings for a cascade PID controller (corrected for better disturbance rejection)

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

Translated to the ideal form using the translation formulas given in [6, Equation (36)], with $c = 1+\tau_D/\tau_I = 1+\tau_2/(4(\tau_c+\theta))$:

$$K_c' = K_c c, \quad \tau_I' = \tau_I, c \quad \tau_D' = \tau_D/c$$

For the controller form in *

$$K_c' = K_c c, \quad K_I' = K_c'/\tau_I' = K_c/\tau_I, \quad K_D' = K_c'\tau_D' = K_C\tau_D$$

so we get

$$K_c' = \frac{1}{k}\frac{\tau_1}{\tau_c+\theta}\left(1+\frac{\tau_2}{4(\tau_c+\theta)}\right)$$
$$K_I' = \frac{1}{4k}\frac{\tau_1}{(\tau_c+\theta)^2}$$
$$K_D' = \frac{1}{k}\frac{\tau_1\tau_2}{\tau_c+\theta}$$

or

$$K_c' = \frac{1}{4k}\frac{1}{(\tau_c+\theta)^2}\tau_1\left(\tau_2+4(\tau_c+\theta)\right)$$

Introducing τ_0 and ζ:

$$K_c' = \frac{1}{4k}\frac{1}{(\tau_c+\theta)^2}\tau_0\left(\tau_0+4(\tau_c+\theta)\left(\zeta+\sqrt{\zeta^2-1}\right)\right)$$
$$K_I' = \frac{1}{4k}\frac{1}{(\tau_c+\theta)^2}\tau_0\left(\zeta+\sqrt{\zeta^2-1}\right)$$
$$K_D' = \frac{1}{k}\frac{\tau_0^2}{\tau_c+\theta}$$

Or, in terms of k'' ($k = k''\tau_0^2$)

$$K_c' = \frac{1}{4k''}\frac{1}{(\tau_c+\theta)^2}\left(1+4(\tau_c+\theta)\left(\zeta+\sqrt{\zeta^2-1}\right)\frac{1}{\tau_0}\right) \tag{C.9a}$$
$$K_I' = \frac{1}{4k''}\frac{1}{(\tau_c+\theta)^2}\frac{1}{\tau_0}\left(\zeta+\sqrt{\zeta^2-1}\right) \tag{C.9b}$$
$$K_D' = \frac{1}{k''}\frac{1}{\tau_c+\theta} \tag{C.9c}$$

Comparing (C.9) with (C.7) we note that K_D' is unchanged.

C.3 Correction for oscillating processes, $0 < \zeta < 1$

Can we extend (C.9) to $\zeta < 1$? Not directly because this would give complex K_c' and K_I'.

To get some idea let us consider the double-integrating process which corresonds to

$$\tau_0 = \infty, \quad \zeta = 0$$

For this process we have already (C.6) which seems reasonable. These settings,

$$K_c = \frac{1}{k''}\frac{1}{4(\tau_c+\theta)^2}, \quad \tau_D = 4(\tau_c+\theta), \quad \tau_I = 4(\tau_c+\theta)$$

have $\tau_D/\tau_I = 1$. Thus $c = 1+\tau_D/\tau_I = 2$ and (C.6) corresponds to the following SIMC ideal-form tunings for double-integrating processes:

$$K_c' = K_c \cdot 2 = \frac{1}{2k''}\frac{1}{(\tau_c+\theta)^2}$$
$$\tau_I' = \tau_I \cdot 2 = 8(\tau_c+\theta)$$
$$\tau_D' = \tau_D/2 = 2(\tau_c+\theta)$$

or

$$K_c' = \frac{1}{2k''}\frac{1}{(\tau_c+\theta)^2} = \frac{\tau_0^2}{2k}\frac{1}{(\tau_c+\theta)^2} \tag{C.10a}$$

$$K_I' = K_c'/\tau_I' = \frac{1}{16k}\frac{\tau_0^2}{(\tau_c+\theta)^3} \tag{C.10b}$$

$$K_D' = K_c'\tau_D' = \frac{1}{k''}\frac{1}{\tau_c+\theta} = \frac{\tau_0^2}{k}\frac{1}{\tau_c+\theta} \tag{C.10c}$$

Want to find a correction that fits both (C.9) and (C.10), and which resembles (C.7) as much as possible. Let us recall the expressions:

(C.7)	IMC setpoint settings
(C.9)	SIMC-correction for integrating processes with $\zeta > 1$
(C.10)	SIMC-correction for double-integrating processes with $\zeta = 0$

Let us compare the expressions. Frist, the P-action: From (C.7) we have:

$$\begin{aligned} K_c' &= \frac{1}{k''}\frac{1}{\tau_c+\theta}\frac{2\zeta}{\tau_0} \\ &= \frac{1}{k}\frac{1}{\tau_c+\theta}2\zeta\tau_0 \\ &= \frac{1}{k}\frac{1}{\tau_c+\theta}(\tau_1+\tau_2) \end{aligned}$$

Corrections: From (C.9) we have:

$$\begin{aligned} K_c' &= \frac{1}{4k''}\frac{1}{(\tau_c+\theta)^2}\left(1+4(\tau_c+\theta)+\left(\zeta+\sqrt{\zeta^2-1}\right)\frac{1}{\tau_0}\right) \\ &= \frac{1}{4k''}\frac{1}{(\tau_c+\theta)^2}\left(1+\frac{4(\tau_c+\theta)}{\tau_2}\right) \\ &= \frac{1}{4k}\frac{1}{(\tau_c+\theta)^2}\tau_0^2\left(1+\frac{4(\tau_c+\theta)}{\tau_2}\right) \\ &= \frac{1}{4k}\frac{1}{(\tau_c+\theta)^2}\tau_1\left(\tau_2+4(\tau_c+\theta)\right) \end{aligned}$$

and from (C.10):

$$K_c' = \frac{1}{2k''}\frac{1}{(\tau_c+\theta)^2} = \frac{1}{2k}\frac{1}{(\tau_c+\theta)^2}\tau_0^2$$

Integral gain: (C.7) gives:

$$K_I' = \frac{1}{k''}\frac{1}{\tau_c+\theta}\frac{1}{\tau_0^2} = \frac{1}{k}\frac{1}{\tau_c+\theta}$$

Corrections: (C.9):

$$\begin{aligned} K_I' &= \frac{1}{k''}\frac{1}{(\tau_c+\theta)^2}\frac{1}{\tau_0}\left(\zeta+\sqrt{\zeta^2-1}\right) \\ &= \frac{1}{4k''}\frac{1}{(\tau_c+\theta)^2}\frac{1}{\tau_2} = \frac{1}{4k}\frac{1}{(\tau_c+\theta)^2}\tau_1 \end{aligned}$$

(C.10):

$$K_I' = \frac{1}{16k''}\frac{1}{(\tau_c+\theta)^3} = \frac{1}{16k}\frac{1}{(\tau_c+\theta)^3}\tau_0^2$$

Derivative gain:

$$\begin{aligned}(C.7): \quad K_D' &= \tfrac{1}{k''}\tfrac{1}{\tau_c+\theta} \\ (C.9): \quad K_D' &= \tfrac{1}{k''}\tfrac{1}{\tau_c+\theta} \\ (C.10): \quad K_D' &= \tfrac{1}{k''}\tfrac{1}{\tau_c+\theta}\end{aligned}$$

Note: $k'' = k/\tau_0^2 = k/(\tau_1\tau_2)$.

From this we have:

- K_D' is easy because it is unchanged
- K_c': Try absolute value $|\zeta^2 - 1|$ for (C.9). Gives factor 2 lower K_c' for (C.10), but this should be OK. But is seems like (C.9) is close to (C.10), so maybe it is even better just to take the maximum or even better: Simplify and always use expression (C.10) for K_c'. (Which is at most a factor 2 larger than B). By numerical inspection one can see that (C.10) > (C.9) in some cases, but at most a factor 2
- K_I': Can not use (C.9) directly due to division by τ_0. Better to take the maximum ((C.9) generally largest except when τ_0 is large. By looking at some examples one can show that (C.10) > (C.9) in some cases, and difference arbitary large for large values of τ_0

Since we ideally would like to retain the SIMC-rules for $\zeta > 1$,it seems easier to simply interpolate between (C.10) and (C.10).

C.4 Conclusion

Skogestad, still with reference to the `.txt` file as menitoned above, proposes the following conlusion to this derivation:

- $K_c' = \max\{A, X\}$, where $X = B$ for $\zeta \geq 1$ and $X = \zeta B' + (1-\zeta)C$ for $\zeta < 1$
- $K_I' = \max\{A, X\}$, where $X = B$ for $\zeta \geq 1$ and $X = \zeta B' + (1-\zeta)C$ for $\zeta < 1$
- K_D' as given from (C.7)
- A is the settings given in (C.7), B is the settings given in (C.9) and C is the settings given in (C.10)
- B' is obtained from B by setting $\sqrt{\zeta^2 - 1} = 0$ for $\zeta < 1$

This gives smooth transition and also makes the limiting case unchanged.

Appendix D

Matlab highlight

s

As the main simulation tools for this report MATLAB and SIMULINK were used. A lot of files were eventually made, and I will not attach all files here, since that would only imply a lot of pages of uninteresting code. Rather, I thought I should include some files that other studens (my younger colleagues) would perhaps use as a starting point for their work. Note that a very good startingpoint for PID-tuning related work is to use the files found on prof. Skogestad's homepage, `http://www.nt.ntnu.no/users/skoge/publications/2003/tuningPID/mfiles/`. If you are interested in other `Matlab`-functions that you suspect I have, please feel free to contact me.[1]

D.1 Lower gain margin

The routine `minbode.m` calculates a lot of performance and robustness criteria, but unfortunately not the lower gain margin. I therefore made a MATLAB-function called `GM_unst.m`, which calculates this. I also included the usual gain margin and phase margin, but this was taken from `minbode.m`. I just included it for simplicity when you use the function.

```
function [GM_L,GM,PM] = GM_unst(sys,w)
% function [GM_L,GM,PM] = GM_unst(sys,w)
%
% calculation of lower and upper gain margins and
% phase margin using matlab's bode.m function
%
% Henrik Manum, 10. Nov. 2005
[mag_sys,phase_sys] = bode(sys,w);
% get the mag and phase out of the 3D-matrix
for i = 1:length(w)
    mag(i) = mag_sys(1,1,i);
    phase(i) = phase_sys(1,1,i);
end
```

[1]Current email adress is `henriman@stud.ntnu.no`.

```
vA = phase - 360; A = mag;
%Beregning av w180 og gain margin
under = 1;
for m=1:length(vA)
  if (vA(m)< -180 )&(under == 1)
    pos = m;
  elseif vA(m) > -180
    under = 0;
  end

end
a = (vA(pos+1) - vA(pos))/(w(pos+1) - w(pos));
w_lower = (-180 - vA(pos))*(w(pos+1) - w(pos))/a;
a2 = (A(pos+1) - A(pos))/(w(pos+1)-w(pos));
A_w_lower = a2*w_lower/(w(pos + 1)- w(pos)) + A(pos);
GM_L = 1/A_w_lower;


if min(vA)>-180
  disp(['Ingen fase kryssover frekvens i det aktuelle frekvens' ...
' intervallet'])
    w180=[];
    GM=[];
else
  for m=pos:length(vA)
    if vA(m)>-180
      posisjon=m;
    end
  end
  %Secant
  w180=w(posisjon)-(vA(posisjon)+180)* ...
      (w(posisjon+1)-w(posisjon))/(vA(posisjon+1)-vA(posisjon));
  %Gain margin ved interpolasjon
  Lw180=A(posisjon)+(w180-w(posisjon))*(A(posisjon+1)-A(posisjon))...
      /(w(posisjon+1)-w(posisjon));
  GM=1/Lw180;
end

% PM
if max(A)<1
  disp('Maks amplitude mindre enn ein.')
  wc=[];
  vAwc=[];
  PM=[];
else
  for m=1:length(A)
    %Kryssing oven fra
    if A(m)>1
      posisjon=m;
```

```
    end
  end
  %Secant
  wc=w(posisjon)-(A(posisjon)-1)* ...
      (w(posisjon+1)-w(posisjon))/(A(posisjon+1)-A(posisjon));
  %Fasen ved
  vAwc=vA(posisjon)+(wc-w(posisjon))*(vA(posisjon+1)-vA(posisjon))...
      /(w(posisjon+1)-w(posisjon));
  PM=vAwc+180;
end
```

D.2 Ziegler-Nichols tuning

Ziegler-Nichols tuning and the modified version of Tyreus-Lyuben are often used in PID-tuning evaluation. I could not find that these methods had been implemented in the files on prof. Skogestads homepage, so I made the following file `ZN.m`. Note that I assume only one crossing of the -180° line here. I have not tested the file on unstable processes, but it surely needs modification if these cases should be covered. I guess, if there is only one phase crossing from above, we could use this one (that is, the upper gain margin) to derive the ZN tuning for unstable processes, so the modification does not need to be large.

```
function [Kc,taui,taud] = ZN(num,den,gain,delay,w,method)
% function [Kc,taui,taud] = ZN(num,den,gain,delay,w,method)
%  Input:
%  num      : er teller polynomet
%  den      : er nevner polynomet
%  gain     : er forsterkning
%  delay    : er dodtid
%  w        : er frekvensene (NB! omraadet maa vaere stort nok)
%  method   : 1 = ZN, P
%             2 = ZN, PI
%             3 = ZN, PID              [Ideal form]
%             4 = Tyreus-Luyben, PI
%             5 = Tyreus-Luyben, PID    [Ideal form]
%
% need: minbode.m
%
% Henrik Manum, 17. oct 2005
[A,vA,w180,wc,GM,PM,Ms,Mt]=minbode(num,den,gain,delay,w);
% søker 1/Kcu...
mindre = w < w180;
storre = w > w180;
lengde = length(w);
wl    = w(sum(mindre));
wu    = w(sum(mindre)+1);
a = (w180 - wl)/(wu-wl);
if (a < 0)|(a > 1)
```

```
    error('a not between 0 and 1! Please improve the routine');
end
A_wc = A(sum(mindre)) + a*(A(sum(mindre)+1) -  A(sum(mindre)));
Kcu = 1/A_wc;
Pu = 2*pi/w180;


% ref: s.318 i Process Dynamics and Control
if method == 1
    Kc = 0.5*Kcu;
    taui = 0;
    taud = 0;
elseif method == 2
    Kc = 0.45*Kcu;
    taui = Pu/1.2;
    taud = 0;
elseif method == 3
    Kc = 0.6*Kcu;
    taui = Pu/2;
    taud = Pu/8;   % paralell/ideal form !!!!
elseif method == 4
    Kc = 0.31*Kcu;
    taui = 2.2*Pu;
    taud = 0;
elseif method == 5
    Kc = 0.25*Kcu;
    taui = 2.2*Pu;
    taud = Pu/6.3;  % paralell/ideal form !!!!!
else
    error('Please choose a method')
end

subplot(2,2,[1 2]); loglog(w,A);
subplot(2,2,[3 4]); semilogx(w,vA); axis([w(1) w(end) -270 0])
```