Master study
Systems and Control Engineering
Department of Technology
Telemark University College
DDiR, September 22, 2009

SCE1106 Control Theory

Exercise 3

Design of PID controller for a chemical reactor

Exercise 1

we are going to design a PID controller in this exercise. The method which are going to be used is very general and leads to a simple and practical method for the design of PID controllers. The method is based on the fact that the process model may be approximated with a transfer function of the form, i.e. with inverse response

$$h_p(s) = k\frac{1-\tau s}{(1+T_1 s)(1+T_2 s)}, \tag{1}$$

or with delay

$$h_p(s) = k\frac{e^{-\tau s}}{(1+T_1 s)(1+T_2 s)}. \tag{2}$$

Here $T_1 \geq T_2 \geq 0$. The method can still be used if the second time constant is zero, i.e. $T_2 = 0$.
Note that if you have a more complicated model then model reduction techniques and system identification method may be used in order to obtain a model of the form (1) or (1). Here, τ, is the effective time delay or inverse response time. $T_1 > T_2$ is the dominant (largest) time constant. the model can also be simplified further to a 1st order delay model by the half-rule, i.e.,

$$h_p(s) = k\frac{1-\tau s}{1+T_1 s}, \tag{3}$$

where

$$T_1 \quad := T_1 + \tfrac{1}{2}T_2, \tag{4}$$
$$\tau \quad := \tau + \tfrac{1}{2}T_2. \tag{5}$$
$$\tag{6}$$

is obtained by the half-rule. We have here neglected the smallest time constant T_2 and distributed it evenly on the remaining time constant T_1 and the time delay τ.

a) Linearize the model around the steady state values and find a linearized model of the form

$$\begin{aligned} \dot{\delta x} &= A_c \delta x + B_c \delta u + C_c \delta v, & (7) \\ \delta y &= D \delta x, & (8) \end{aligned}$$

Find expressions for the system matrices A_c, B_c, C_c and D_c and the variables δx, δu and δv.

b) Find a transfer function model of the form

$$\delta y = h_p(s)\delta u, \quad h_p(s) = k\frac{1 - \tau s}{(1 + T_1 s)(1 + T_2 s)}. \tag{9}$$

Find numerical values for the parameters k, τ, T_1 and T_2. Find the zeroes of the system. What can be said about the system and its dynamics?

c) We shall in this task design a PID controller for the chemical reactor. Take the transfer function in task b) above as the starting point. The nominator polynomial is representing an inverse response which is approximately equivalent with a transport delay because

$$e^{-\tau s} \approx 1 - \tau s. \tag{10}$$

Let us specify the respnse from the reference signal, r, to the output δy by

$$\frac{\delta y}{r} = \frac{1 - \tau s}{1 + T_c s}. \tag{11}$$

Here, T_c is an specified time constant for the set-point response. This time constant may be chosen proportional with the time delay τ. This will be discussed furter. there is nothing to do with the time delay or inverse response in the set-pint respons so we let it be in the expression (11). We also have that

$$\frac{\delta y}{r} = \frac{h_p h_c}{1 + h_p h_c}, \tag{12}$$

where $h_c(s)$ is the transfer function for the PID controller. Putt the two transfer functions, i.e. (11) and (12). equal to each other and solve for the controller $h_c(s)$. You shall obtain the controller $h_c(s)$ of the form.

$$h_c(s) = \frac{1}{k}\frac{T_1}{T_c + \tau}\frac{1 + T_1 s}{T_1 s}(1 + T_2 s). \tag{13}$$

This is a cascade formulation of the PID controller.

$$h_c(s) = K_p \frac{1 + T_i s}{T_i s}(1 + T_d s), \tag{14}$$

where

$$\begin{aligned} K_p &= \frac{1}{k}\frac{T_1}{T_c + \tau}, & (15) \\ T_i &= T_1 & (16) \\ T_d &= T_2. & (17) \end{aligned}$$

d) It is reasonable to specify the time constant, T_c, of the set-pint response to be equal to or grather than the time delay, i.e.,

$$T_c \geq \tau. \tag{18}$$

Chose $T_c = \tau$ and show that

$$K_p^{\max} = \frac{1}{2k}\frac{T_1}{\tau}. \tag{19}$$

Put numerical values into the expressions.

e) You shoul know have found reasonable values for the PI controller parameters K_p and T_i. Simulate the system with these settings. Remember that the parameters is found for the continuous time system, and that you shoul take the step length into account when discretizing the system. The controller is in continuous time given by

$$e = r - y, \tag{20}$$
$$u = K_p e + z, \tag{21}$$
$$\dot{z} = \frac{K_p}{T_i} e. \tag{22}$$

Using Explicit Euler for discretizing the controller gives

$$e = r - y, \tag{23}$$
$$u = K_p e + z, \tag{24}$$
$$z = z + \frac{K_p}{\frac{T_i}{h}} e. \tag{25}$$

where h is the step length. Hence, we can view $T_i^d = \frac{T_i}{h}$ as a discrete integral time. It is important to take this into account when implementing the PI-controller.

Oppgave 2

The effective time delay (the length of the inverse response) can be found by simulating the step-response and plotting the response. From this we find $\tau = 24$ [sec].
In order to design a PI-controller for the process it can be useful to approximate the process model with a 1st order model with inverse response. An trial and error procedure shows that the model (9) can be approximated with the following model

$$h_p(s) = k\frac{1 - \tau s}{1 + T_1 s}. \tag{26}$$

where

$$k = \frac{2}{125}, \tag{27}$$
$$\tau = \frac{1}{120}, \tag{28}$$
$$T_1 = \frac{1}{75}. \tag{29}$$

Here, the half-rule can also with advantage be used. Do it!
We shall now use the model (26) for PID controller synthesis. Compare the step response of (26) with (9). It is important that the time constant and the effective delay for the approximate model is as similar to the real process as possibile. The model (26) gives an effective delay of approximately $\tau = 30$ [sec]. Which values for K_p and T_i gives this? Simulate the reactor with this PI controller settings and compare with the results in task 1.

Solution of some particular sub-problems

Solution task 1a) Numerical values for the state space model matrices are given by

$$A_c = \begin{bmatrix} -125 & 0 \\ 50 & -125 \end{bmatrix}, \; B_c = \begin{bmatrix} 7.5 \\ -1 \end{bmatrix}, \; C_c = \begin{bmatrix} 25 \\ 0 \end{bmatrix}, \; D = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad (30)$$

where sub-script $(\cdot)_c$ is used to denote continuous time.

Solution task 1b) The transfer function $h_p(s) = D(sI - A)^{-1}B$ gives after some calculations that

$$\delta y(s) = h_p(s)\delta u(s), \quad h_p(s) := \frac{-s + 250}{s^2 + 250s + 15625} = \frac{2}{125}\frac{1 - \frac{1}{250}s}{(1 + \frac{1}{125}s)^2}. \quad (31)$$

This means that

$$k = \frac{2}{125}, \quad T_1 = T_2 = \frac{1}{125}, \quad \tau = \frac{1}{250}. \quad (32)$$

Se figure 1 for a unit step-response in the control input variable u.

Solution task 1d) With numerical values we obtain the following settings for the PI-controller parameters

$$K_p = \frac{1}{2k}\frac{T_1}{\tau} = \frac{125}{4}\frac{\frac{1}{125}}{\frac{1}{250}} = 62.5. \quad (33)$$

$$T_i = T_1 = \frac{1}{125}. \quad (34)$$

Solution task 2)

The numerical values for k, τ og T_1 as given in task 2 gives the following settings for the PI-controller

$$K_p = \frac{1}{2k}\frac{T_1}{\tau} = \frac{125}{4}\frac{\frac{1}{75}}{\frac{1}{120}} = 50. \quad (35)$$

$$T_i = T_1 = \frac{1}{75}. \quad (36)$$

Using the half-rule gives a first order model with inverse response of the form

$$h_p(s) = k\frac{1 - \tau s}{1 + T_1 s}. \quad (37)$$

where

$$k = \frac{2}{125}, \quad (38)$$

$$\tau = \tau + \frac{1}{2}T_2 = \frac{1}{250} + \frac{1}{2}\frac{1}{125} = \frac{1}{125}, \quad (39)$$

$$T_1 = T_1 + \frac{1}{2}T_2 = \frac{1}{125} + \frac{1}{2}\frac{1}{125} = \frac{3}{250} \approx \frac{1}{83}. \quad (40)$$

Se figure 1 for a unit step-response in the control input u. Using the synthesis rules gives the following upper limit for K_p.

$$K_p = \frac{1}{2k}\frac{T_1}{\tau} = \frac{125}{4}\frac{3 \cdot 125}{250} = 46.9. \tag{41}$$

$$T_i = \frac{1}{83.3}. \tag{42}$$

Concluding remarks

We have found three PI-controller settings. Which one of the settings is the best?
The settings found in task 1b) can not be used. It gives an oscillating behavior. the reason is that one can not only skip the time constant T_2 without modifying the remaining time constant T_1 and the inverse response time (delay) τ.
The settings in task 2 found by using the half-rule seams to result in the best setting. The reason is because it results in the most conservative settings with the smallest K_p and T_i. The difference is however small.

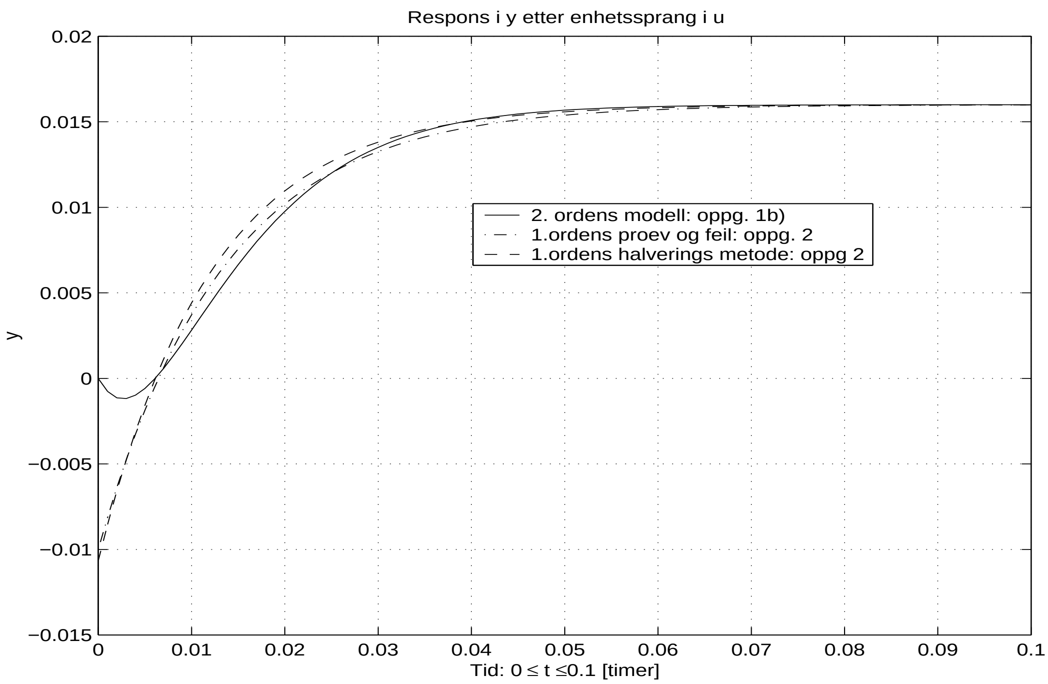

Figure 1: Unit step-response for the model in (9) and the approximate 1st order models in task 2 after unit step in u. The figure is generated by the MATLAB script file **demo_reac_pid.m**. Se the enclosed MATLAB file.

```
%% demo_reac_pid
% Simulering av sprangresponser for linearisert reaktor modell
% samt 1. ordens modellapproksimasjoner som skal benyttes til
% syntese av PI-regulatorer.

clear all

%%% Linearisert prosessmodell.
num=[0,-1,250];
den=[1,250,15625];
[a,b,d,e]=tf2ss(num,den);   % Tilstandsrommodell.

%%% Sprangrespons til den lineariserte modellen med to tidskonstanter.
h=0.001; t1=0.1;
t=0:h:t1;
y=step(num,den,t);

%%% Tunet 1.ordens modell (proev og feil ga denne)
num1=(2/125)*[-1/120,1]; den1=[1/75,1];
num1=(2/125)*[-1/120,1]; den1=[1/75,1];
y1=step(num1,den1,t);

%%% Halveringsregel
T1=1/125+0.5*(1/125);
tau=1/250+0.5*(1/125);
num2=(2/125)*[-tau,1]; den2=[T1,1];
y2=step(num2,den2,t);

%%% Plotter i samme figur.
figure(1), clf
plot(t,y,'r-',t,y1,'b-.',t,y2,'k--'), grid, ylabel('y')
legend('2. ordens modell: oppg. 1b)',...
   '1.ordens proev og feil: oppg. 2',...
   '1.ordens halverings metode: oppg 2')
title('Respons i y etter enhetssprang i u')
xlabel(strcat('Tid: 0 \leq t \leq ',num2str(t1),' [timer]'))

%%% Testing
num3=(2/125)*[-(1/250+1/125),1]; den3=[1/125,1];
y3=step(num3,den3,t);

%%% Plotter i samme figur.
figure(2), clf
plot(t,y,'r-',t,y1,'b-.',t,y2,'k--',t,y3,'r--'), grid, ylabel('y')
legend('2. ordens modell: oppg. 1b)',...
   '1.ordens proev og feil: oppg. 2',...
   '1.ordens halverings metode: oppg 2',...
```

```
   '1.ordens test')
title('Respons i y etter enhetssprang i u')
xlabel(strcat('Tid: 0 \leq t \leq ',num2str(t1),' [timer]'))

%%% Closed loop responses

T1=1/125; tau=1/250; k=2/125;

%% Standard tuning
Ti=T1;
Kp=40;
Kp=dread('Kp=',Kp);

kt=k*Kp/Ti;
num_c1=kt*[0,-tau,1];
den_c1=[T1,(1-kt*tau),kt];
yc4=step(num_c1,den_c1,t);
%%
Ti=1/83; Kp=46.9;
kt=k*Kp/Ti;
num_c1=kt*[0,-tau,1];
den_c1=[T1,(1-kt*tau),kt];
yc2=step(num_c1,den_c1,t);

%%% Plotter i samme figur.
figure(3), clf
plot(t,yc2,'r--',t,yc4,'k-'), grid, ylabel('y')
legend('1.ordens halverings metode: oppg 2','2. ordens')
title('Respons i y etter enhetssprang i u')
xlabel(strcat('Tid: 0 \leq t \leq ',num2str(t1),' [timer]'))
```