Master study
Systems and Control Engineering
Department of Technology
Telemark University College
DDiR, October 26, 2006

SCE1106 Control Theory

Exercise 6

Task 1

a) The poles of the open loop system is found as the roots of the nominator polynomial of the transfer function $h_p(s)$. I.e. from the roots of the characteristic equation

$$s^2 + 3s + 2 = 0. \tag{1}$$

This gives the poles $s_1 = -1$ and $s_2 = -2$. The process model can then be written as

$$\begin{aligned} h_p(s) &= \frac{e^{-2s}}{s^2+3s+2} = \frac{e^{-2s}}{(s+1)(s+2)} = \frac{1}{2}\frac{e^{-2s}}{(s+1)(\frac{1}{2}s+1)} \\ &= k\frac{e^{-\tau s}}{(1+T_1 s)(1+T_2 s)},) \end{aligned} \tag{2}$$

where the steady state gain of the open loop system is given by

$$k = \frac{1}{2} \tag{3}$$

and the time constant for the open loop system

$$T_1 = 1, \tag{4}$$

$$T_2 = \frac{1}{2}. \tag{5}$$

b) We use the half rule in order to reduce the process model (2) to a 1st order model approximation with inverse response of the form

$$h_p(s) = k\frac{1-\tau s}{(1+Ts)},) \tag{6}$$

where $k = \frac{1}{2}$ and

$$T = T_1 + \frac{1}{2}T_2 = 1 + \frac{1}{2}\frac{1}{2} = \frac{5}{4} = 1.25 \tag{7}$$

$$\tau := \tau + \frac{1}{2}T_2 = 2 + \frac{1}{2}\frac{1}{2} = \frac{9}{4} = 2.25 \tag{8}$$

Using the Skogestad method and the reduced model gives the PI controller parameters

$$K_p = \frac{1}{2k}\frac{T}{\tau} = \frac{5}{9} \approx 0.56 \tag{9}$$

$$T_i = T = \frac{5}{4} = 1.25 \tag{10}$$

The poles of the closed loop system is approximately given by $s = -\frac{1}{T_c}$ where $T_c = \tau = 2.25$. The starting point for the above PI controller parameters is that the closed loop set point response is specified to be

$$\frac{y}{r} = \frac{1-\tau s}{(1+T_c s)} = \frac{h_p h_c}{1+h_p h_c} = T(s) \tag{11}$$

with a time constant for the closed loop system equal to the time delay of the open loop system, i.e. $T_c = \tau$.

A more exact computation of the poles of the closed loop system is given by the roots of the characteristic equation

$$1 + h_p h_c = 0 \tag{12}$$

In order to obtain exact computation of the closed loop system poles we must use the process model (2) and not the approximation (6).

c) The solution of this sub problem is implemented in the MATLAB script **ov6_1c.m**. The script can be down loaded from the homepage of the course. Se also the enclosed copy of the script. The script needs the Control System Toolbox in order to work.

d) This sub problem is implemented in the MATLAB script **ov6_1d.m**. The script can be down loaded from the home page of the course. The script is Toolbox independent.

MATLAB-script: ov6_1d

Figure 1: Simulation of the closed loop system after a unit step response in the reference,r. Process $h_p(s) = \frac{e^{-2s}}{s^2+3s+2}$ with PI controller $h_c(s) = K_p \frac{Tis+1}{Tis}$ where $K_p = \frac{5}{9} \approx 0.56$ and $T_i = 1.25$. The PI controller parameters are found by the Skogestad method as in step 1b).

```
% ov6_1d.m
% Formaal: Simulering av system h_p(s)=exp(-2s)/(s^2+3s+2) med PI-regulator
% h_c(s)=Kp(1+Tis)/(Tis) der Ti=1.25 og Kp=5/9=0.56. PI-regulatorparametrene
% er funnet vha "Skogestads metode".
% Benyttede spesielle funksjoner og toolbox'er: ingen
% Skrevet av: David Di Ruscio, 18. oktober 2002.

num=[0,0,1];                       % Definerer teller og nevner i h_p(s)
den=[1,3,2];                       % uten transportforsinkelsesleddet.
[A,B,D,E]=tf2ss(num,den);          % Tilstandsrommodell uten transportforsinkelse.

t0=0; t1=20; N=200;                % Lager en passende tidshorisont.
t=linspace(t0,t1,N);
dt=t(2)-t(1);                      % Samplingsintervall.

nt=floor(2/dt); yt=zeros(nt,1);    % Array for implementering av transport
                                   % forsinkelse.
r=1;                               % Referansesignal, sprang fra r=0 til r=1 ved t=
% PI-regulatorparametre
Kp=0.56; Ti=1.25;                  % PI-regulatorparametre fra oving 6, 1b.
Kp=0.6631; Ti=1;                   % PI-regulatorparametre fra oving 7.
Kp=1.4009; Ti=5.4366;              % Ziegler-Nichols PI-reg.parametre.

x0=[0;0];                          % Initialverdi for tilstandsvektoren, x.
z0=0;                              % Initialverdi for regulatortilstanden, z.

%%% Simuleringsloekke for det lukkede systemet.
x=x0; z=z0;
for i=1:N
   y=D*x;                          % Mling fr transportforsinkelseselementet.

   ym=yt(nt);                      % Implementering av transportforsinkelse.
   for k=nt:-1:2
      yt(k)=yt(k-1);
   end
   yt(1)=y;

   e=r-ym;                         % Beregner reguleringsavviket.

   u=Kp*e+z;                       % Implementering av PI-regulator.
   z=z+dt*Kp*e/Ti;                 % (integrerer med eksplisitt Euler.)

   Y(i,1:2)=[ym y];                % Lagrer systemvariable.
   R(i,1)=r;
   U(i,1)=u;

   fx=A*x+B*u;                     % Paatrykker paadraget paa prosessen.
```

```
   x=x+dt*fx;                        % (integrerer med eksplisitt Euler.)
end

%%% Plotter resultatene
subplot(211), plot(t,U), grid
title('Simulering av lukket system'), ylabel('u')
subplot(212), plot(t,[R Y(:,1)]), grid
xlabel('Tid [s]'), ylabel('y og r')

print -deps ov6_fig1
```

MATLAB-script: ov6_1c

```
% ov6_1c.m
% Formaal: Loesning til oving 6 oppgave 1c.
% Benyttede toolbox'er: control system toolbox for matlab.

%%% Definerer h_p(s)
num1=[0,0,1];                      % Definerer teller og nevner i h_p(s)
den1=[1,3,2];                      % uten transportforsinkelsesleddet.
hp1=tf(num1,den1);

tau=2;                             % Pade-approksimasjon til transportforsinkelsen
[num2,den2]=pade(tau,2);

hp2=tf(num2,den2);

hp=series(hp1,hp2);

%%% Definerer h_c(s)
Kp=5/9; Ti=1.25;

num_hc=[Kp*Ti,Kp];
den_hc=[Ti,1e-6];
hc=tf(num_hc,den_hc);

%%% definerer h_0(s)=h_p(s)*h_c(s)
h0=series(hp,hc);

%%% Definerer y/r=T(s)=h_p*h_c/(1+h_p*h_c)
T=feedback(h0,1);

%%% Bode-plot av T(s)
figure(1), bode(T)

%%% Tidsrespons i y etter enhetssprang i r.
figure(2)
step(T)
```