Master study
Systems and Control Engineering
Department of Technology
Telemark University College
DDiR, November 9, 2006

SCE1106 Control Theory

Solution Exercise 8

Task 1

a) The loop transfer function with the process model with a P controller is given by

$$h_0(s) = h_c(s)h_p(s) = K_p \frac{e^{-\tau s}}{(1+s)(2+s)} \tag{1}$$

In order to find ω_{180} and K_{cu} we first have to write the frequency response $h_0(j\omega)$ on polar form. We have

$$\begin{aligned} h_0(j\omega) &= K_p \frac{e^{-j\tau\omega}}{(1+j\omega)(2+j\omega)} \\ &= K_p \frac{e^{-j\tau\omega}}{\sqrt{1+\omega^2}e^{j\arctan(\omega)}\sqrt{2^2+\omega^2}e^{j\arctan(\frac{1}{2}\omega)}} \\ &= \frac{K_p}{\sqrt{1+\omega^2}\sqrt{4+\omega^2}} e^{j(-\tau\omega - \arctan(\omega) - \arctan(\frac{1}{2}\omega))} \\ &= |h_0(j\omega)|e^{j\angle h_0(j\omega)}. \end{aligned} \tag{2}$$

This means that the magnitude is given by

$$|h_0(j\omega)| = \frac{K_p}{\sqrt{1+\omega^2}\sqrt{4+\omega^2}} \tag{3}$$

and that the phase shift is given by

$$\angle h_0(j\omega) = -\tau\omega - \arctan(\omega) - \arctan(\frac{1}{2}\omega) \tag{4}$$

The phase crossover frequency, ω_{180}, is found as the frequency in which the phase shift is equal to $-\pi$ (i.e., $-180°$). This means that we have to solve the equation

$$\angle h_0(j\omega_{180}) = -\tau\omega_{180} - \arctan(\omega_{180}) - \arctan(\frac{1}{2}\omega_{180}) = -\pi. \tag{5}$$

This equation is simply solved by fiks point iterations. See the MATLAB script **losn_ov8.m**. The solution is

$$\omega_{180} \approx 0.9631 \tag{6}$$

Kritisk forsterkning, K_{cu}, er den verdi av K_p som medfører at amplitudeforholdet blir lik en samtidig som fasevinkelen er $-180°$. Dvs.

$$|h_0(j\omega_{180}| = \frac{K_p}{\sqrt{1+\omega_{180}^2}\sqrt{4+\omega_{180}^2}} = 1. \tag{7}$$

Dette betyr at systemet er på stabilitetsgrensen og det samme som at det lukkede systemet har en forsterkningsmargin $GM = 1$. Vi løser ut for K_p og finner at

$$K_{cu} \stackrel{def}{=} K_p = \frac{1}{h_p(j\omega_{180})} = \sqrt{1+\omega_{180}^2}\sqrt{4+\omega_{180}^2} \approx 3.0819 \tag{8}$$

b) Ziegler-Nichols PI-regulatorparametre finnes nå som en funksjon av kritisk forsterkning, K_{cu}, og fasekryssfrekvensen, ω_{180}. Vi har

$$K_p = \frac{K_{cu}}{2.2} \approx 1.4009, \tag{9}$$

$$P_u = \frac{2\pi}{\omega_{180}} \approx 6.5239, \tag{10}$$

$$T_i = \frac{P_u}{1.2} \approx 5.4366 \tag{11}$$

c) Formålet med dette oppgavepunktet er å undersøke hvilke marginer vi har i reguleringssystemet med PI-regulatoren funnet i punkt b) over. Sløyfetransferfunksjonen for prosess med PI-regulator er gitt ved

$$h_0(s) = h_c(s)h_p(s) = K_p\frac{1+T_i s}{T_i s}\frac{e^{-\tau s}}{(1+s)(2+s)}. \tag{12}$$

Vi skriver frekvensresponsen $h_0(j\omega)$ på polar form. Dvs.

$$h_0(j\omega) = K_p\frac{1+jT_i\omega}{jT_i\omega}\frac{e^{-j\tau\omega}}{(1+j\omega)(2+j\omega)} = |h_0(j\omega)|e^{j\angle h_0(j\omega)} \tag{13}$$

der

$$|h_0(j\omega)| = \frac{K_p\sqrt{1+(T_i\omega)^2}}{T_i\omega\sqrt{1+\omega^2}\sqrt{4+\omega^2}} \tag{14}$$

og

$$\angle h_0(j\omega) = -\tau\omega + \arctan(T_i\omega) - \frac{\pi}{2} - \arctan(\omega) - \arctan(\frac{1}{2}\omega). \tag{15}$$

Dette gir en fasekryssfrekvens (for system med PI-regulatoren fra punkt b)) lik

$$w_{180} = 0.8943 \tag{16}$$

og en forsterkningsmargin

$$GM = \frac{1}{|h_0(j\omega_{180})|} = 2.0551. \tag{17}$$

Videre finner vi en forsterkningskryssfrekvens, w_c, og fasemargin, PM, lik

$$w_c = 0.3736, \;\; PM = 1.3947 (= 79.9109^\circ). \tag{18}$$

Dette skulle tilsi (betyr) at vi har normalt gode marginer. Dette betyr ikke nødvendigvis at vi får god regulering som vi skal se i neste deloppgave. Se for øvrig MATLAB-scriptet **losn_ov8.m** for beregningene.

d) Systemet er simulert med de forskjellige PI-regulatorparametrene i fra øving 6, øving 7 og øving 8 i MATLAB-scriptet **ov678_pi.m** og resultatene er vist i Figur 1. Konklusjonen er at Skogestad PI-parametrene i fra øving 6 gir den beste settpunktsresponsen mens Ziegler-Nichols PI-parametrene gir meget dårlig settpunktsrespons i dette tilfellet.

Figure 1: Figuren viser pådraget u, og responsen, y, etter et enhetssprang i referansesignalet r for forskjellige valg av PI-regulatorparametre. Prosesen er gitt ved $h_p(s) = \frac{e^{-\tau s}}{(s+1)(s+2)}$, og regulalatoren er en PI-regulator $h_c(s) = K_p \frac{1+T_i s}{T_i s}$. Vi har benyttet parametrene i fra øving 6, dvs. $K_p = 0.56$, $T_i = 1.25$, parametrene i fra øving 7, dvs. $K_p = 0.6631$, $T_i = 1$ og parametrene i fra øving 8 $K_p = 1.4009$, $T_i = 5.4366$. Figuren er generert av MATLAB scriptet **ov678_pi.m**.

MATLAB-scriptet losn_ov8.m

```
% Fag: prosessregulering
% Loesning til oeving 8
% DDiR, 4/11-2002

% a)
% Beregning av w_180 for prosess med P-regulator
w=1;
for i=1:100
   w=(pi-atan(w)-atan(w/2))/2;
end
w_180=w

% Kritisk forsterkning, lukket system med P-regulator.
a_h0=1/(sqrt(1+w_180^2)*sqrt(4+w_180^2));
Kcu=1/a_h0

% b)
% Ziegler-Nichols PI-regulatorparametre
Kp=Kcu/2.2
Pu=2*pi/w_180;
Ti=Pu/1.2

% c)
% Beregning av w_180 for prosess med PI-regulator og ZN-parametre.
w=1;
for i=1:100
   w=(pi+atan(Ti*w)-pi/2-atan(w)-atan(w/2))/2;
end
w_180_pi=w
% Beregning av forsterkningsmargin
a_h0=Kp*sqrt(1+(Ti*w)^2)/(Ti*w*sqrt(1+w^2)*sqrt(4+w^2))
Gm=1/a_h0

% Beregning av w_c for prosess med PI-regulator og ZN-parametre.
w=1
for i=1:100
   w=Kp*sqrt(1+(Ti*w)^2)/(Ti*sqrt(1+w^2)*sqrt(2+w^2));
end
wc=w

% Beregning av fasemargin
v_h0=-2*w+atan(Ti*w)-pi/2-atan(w)-atan(w/2);
PM=v_h0+pi
PM*180/pi
```

MATLAB-scriptet ov678_pi.m

```
% ov678_pi.m
% Formaal: Simulering av system h_p(s)=exp(-2s)/(s^2+3s+2) med PI-regulator
% h_c(s)=Kp(1+Tis)/(Tis) med forskjellige valg for Kp og Ti. PI-regulatorparametrene
% er funnet vha "Skogestads metode" i oving 6, parametrene i fra oving 7 og 8.
% Benyttede spesielle funksjoner og toolbox'er: ingen
% Skrevet av: David Di Ruscio, 5. november 2002.

clear all
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=t0.
% PI-regulatorparametre
tekst=['PI-regulatorparametre fra oving 6, 1b : (valg 6)'
 'PI-regulatorparametre fra oving 7.    : (valg 7)'
 'Ziegler-Nichols PI-reg.parametre.     : (valg 8)'];
disp(tekst);
pi_reg=6;
pi_reg=dread('PI-parameter valg nr:',pi_reg);
if pi_reg==6
   Kp=0.56; Ti=1.25;                   % PI-regulatorparametre fra oving 6, 1b.
elseif pi_reg==7
   Kp=0.6631; Ti=1;                    % PI-regulatorparametre fra oving 7.
elseif pi_reg==8
   Kp=1.4009; Ti=5.4366;               % Ziegler-Nichols PI-reg.parametre.
else
   disp('Advarsel: hvilke PI-reg. param. oensker du ?');
   Kp=0.56; Ti=1.25;
   disp('Skogestad parametre?');
end
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
```