Master study
Systems and Control Engineering
Department of Technology
Telemark University College
DDiR, November 17, 2009

SCE1106 Control Theory

Solution Exercise 10

Task 1

a) We have a model of the form

$$y = \frac{k}{(1+T_1 s)(1+T_2 s)} u \tag{1}$$

In this case it make sense to do the following definition

$$y = x_2 = \frac{1}{1+T_2 s} x_1 \tag{2}$$

where then

$$x_1 = \frac{k}{1+T_1 s} u. \tag{3}$$

Inverse Laplace transformation gives

$$\dot{x}_1 = -\frac{1}{T_1} x_2 + \frac{k}{T_1} u \tag{4}$$

and

$$\dot{x}_2 = -\frac{1}{T_2} x_2 + \frac{1}{T_2} x_1. \tag{5}$$

This can be written on state space form as follows

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \overbrace{\begin{bmatrix} -\frac{1}{T_1} & 0 \\ \frac{1}{T_2} & -\frac{1}{T_2} \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}^{x} + \overbrace{\begin{bmatrix} \frac{k}{T_1} \\ 0 \end{bmatrix}}^{B} u, \tag{6}$$

$$y = \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{D} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{7}$$

Hence, we have a continuous state space model of the form

$$\dot{x} = Ax + Bu, \tag{8}$$

$$y = Dx. \tag{9}$$

b)

The explicit euler method is obtained by using the approximation $\dot{x} \approx \frac{x_{k+1}-x_k}{\Delta t}$ at the left hand side and using the variables at discrete time k on the right hand side, i.e.

$$\frac{x_{k+1} - x_k}{\Delta t} \approx Ax_k + Bu_k, \tag{10}$$

which gives

$$x_{k+1} = (I + \Delta t A)x_k + \Delta t B u_k, \tag{11}$$

The discrete time measurements equation is given by

$$y_k = Dx_k. \tag{12}$$

c) Using the trapezoid method for discretization of a continuous system

$$\dot{x} = f(x, u) \tag{13}$$

then it make sense to put

$$\frac{x_{k+1} - x_k}{\Delta t} \approx \frac{1}{2}(f(x_k, u_k) + f(x_{k+1}, u_k)) \tag{14}$$

Note that we have used $f(x_{k+1}, u_k)$ as an approximation to the function value at time $k+1$, i.e., $f(x_{k+1}, u_{k+1})$, in order not for the equation to be implicit in the control. The reason is that when using the model for control purposes we do not know the control at the next time instant.

From the linear state space model we obtain

$$x_{k+1} = (I - \frac{\Delta t}{2}A)^{-1}(I + \frac{\Delta t}{2}A)x_k + \Delta t(I - \frac{\Delta t}{2}A)^{-1}Bu_k. \tag{15}$$

d) Some advantages with the trapezoid method:

- The step length parameter Δt can in principle be chosen infinitely large, i.e., $0 \leq \Delta t \leq \infty$. However one may have oscillations and inaccurate solutions when using large step length parameters so one should in practice use a "small" step length parameter. The problem of choosing Δt is a trade of between accuracy of the solution and the speed of computations.
- The trapezoid method have 2nd order accuracy. On the other hand the explicit Euler method have 1st order accuracy. hence, the trapezoid method is more accurate than the explicit euler method.

Advantages with the explicit Euler method:

- The method is simple. Need less computing time than using the trapezoid method, in particular for non-linear functions in which one have to solve an implicit non-linear equation at each step.

Task 2

a) The PID controller may be written

$$u = K_p e + z + K_p T_d s e \tag{16}$$

where

$$z = \frac{K_p}{T_d s} e, \quad \Rightarrow sz = \frac{K_p}{T_i} e. \tag{17}$$

Inverse Laplace-transformation gives

$$u = K_p e + z + K_p T_d \dot{e}, \tag{18}$$

$$\dot{z} = \frac{K_p}{T_i} e. \tag{19}$$

It make sense to put $e(t_0) = r - y = 0$ and $\dot{e}(t_0) = 0$ at startup. We then have that the initial value for the controller state may be chosen as

$$z(t_0) = u_0, \tag{20}$$

where u_0 is a nominal control or working point for the process when turning on the controller system in automatic mode . The nominal control may be found by analyzing the steady state behavior of the process. We have

$$y_0 = k u_0 = r, \tag{21}$$

where k is the gain. this gives the following initial value.

$$z(t_0) = \frac{r(t_0)}{k}. \tag{22}$$

b) We have two possibilities for implementing the derivation $\dot{e}$. The first possibility is to use

$$\dot{e} \approx \frac{e_k - e_{k-1}}{\Delta t}. \tag{23}$$

the second possibility and the most common choice is to not take the derivative of the reference signal, i.e. using

$$\dot{e} - \dot{y} \approx -\frac{y_k - y_{k-1}}{\Delta t}. \tag{24}$$

using the last choice and the explicit euler method for discretization of the controller state space model (19) gives

$$u_k = K_p e_k + z_k - \frac{K_p T_d}{\Delta t}(y_k - y_{k-1}), \tag{25}$$

$$z_{k+1} = z_k + \Delta t \frac{K_p}{T_i} e_k. \tag{26}$$

with initial value $z_0 = u_0 = \frac{r_0}{k}$.

this discrete state space model for the PID controller may be used directly. However, a formulation on deviation form may be derived as follows (compute the deviation $\Delta u_k = u_k - u_{k-1}$). This gives

$$\begin{aligned} u_k - u_{k-1} &= K_p e_k + z_k - \frac{K_p T_d}{\Delta t}(y_k - y_{k-1}) \\ &- (K_p e_{k-1} + z_{k-1} - \frac{K_p T_d}{\Delta t}(y_{k-1} - y_{k-2})) \end{aligned} \tag{27}$$

which may be written as

$$u_k - u_{k-1} = K_p e_k + z_k - z_{k-1} - K_p e_{k-1} - \frac{K_p T_d}{\Delta t}(y_k - 2y_{k-1} + y_{k-2}) \tag{28}$$

Using that

$$z_k - z_{k-1} = \Delta t \frac{K_p}{T_d} e_{k-1}. \tag{29}$$

This gives

$$u_k = u_{k-1} + K_p e_k - K_p(1 - \frac{\Delta t}{T_i})e_{k-1} - \frac{K_p T_d}{\Delta t}(y_k - 2y_{k-1} + y_{k-2}) \tag{30}$$

Hence, this is of the form

$$u_k = u_{k-1} + g_0 e_k + g_1 e_{k-1} + g_2(y_k - 2y_{k-1} + y_{k-2}) \tag{31}$$

where

$$g_0 = K_p,\ \ g_1 = -K_p(1 - \frac{\Delta t}{T_i}),\ \ g_2 = -\frac{K_p T_d}{\Delta t}. \tag{32}$$

c) Using the trapezoid method for integrating the controller state space model (19) gives

$$\frac{z_{k+1} - z_k}{\Delta t} = \frac{1}{2}\frac{K_p}{T_i} e_k + \frac{1}{2}\frac{K_p}{T_i} e_{k+1} \tag{33}$$

As we see, it is not possible to formulate an implementable dicrete state space model for the PID controller of the same form as when the Explicit Euler method was used, as in Equations (25) and (26). The reason for this is that we do not know $e_{k+1} = r_{k+1} - y_{k+1}$ which in this last case is needed in order to compute and update the controller state z_{k+1}.

However, we may use the trapezoid method in order to formulate the PID controller on deviation (incremental) form. Using that

$$z_k - z_{k-1} = \frac{\Delta t}{2}\frac{K_p}{T_i}(e_{k-1} + e_k) \tag{34}$$

and putting this into (28) gives

$$\begin{aligned} u_k &= u_{k-1} + K_p(1+\frac{\Delta t}{2T_i})e_k - K_p(1-\frac{\Delta t}{2T_i})e_{k-1} \\ &- \frac{K_pT_d}{\Delta t}(y_k - 2y_{k-1} + y_{k-2}) \end{aligned} \tag{35}$$

This gives the controller formulation

$$u_k = u_{k-1} + g_0e_k + g_1e_{k-1} + g_2(y_k - 2y_{k-1} + y_{k-2}) \tag{36}$$

where

$$g_0 = K_p(1+\frac{\Delta t}{2T_i}),\ \ g_1 = -K_p(1-\frac{\Delta t}{2T_i}),\ \ g_2 = -\frac{K_pT_d}{\Delta t}. \tag{37}$$

Task 3

```
% ov10_oppg3.m
% Loesningsforslag til oppgave 3 i oeving 10.
% DDiR 15. oktober 2002

T1=1; T2=0.5; k=0.5; tau=2;         % Modelleparametre.

A=[-1/T1,0;1/T2,-1/T2];             % Matriser i kontinuerlig tilstandsrom-
B=[k/T1;0]; D=[0,1];                % modell, dot(x)=Ax+Bu, y=Dx
                                    % uten transportforsinkelsen.

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

nt=floor(2/dt); yt=zeros(nt,1);     % Array for implementering av transport
                                    % forsinkelse.

Phi=inv(eye(2)-dt*A/2)*(eye(2)+dt*A/2);  % Matriser i diskret tilstandsrom-
Delta=dt*inv(eye(2)-dt*A/2)*B            % modell med Trapes-metoden.

Kp=0.56; Ti=1.25;                   % PI-reg param.
g0=Kp; g1=-Kp*(1-dt/Ti);                      % Eksplisitt Euler.
g0=Kp*(1+dt/(2*Ti)); g1=-Kp*(1-dt/(2*Ti));   % Trapes

r=1;                                % referansesignal.

u=0;
x=[0;0];
e_old=0; z=0;
ireg=2;
for i=1:N
```

```
   y=D*x;
   yp=yt(nt);                       % Implementering av transportforsinkelse.
   for k=nt:-1:2
      yt(k)=yt(k-1);
   end
   yt(1)=y;

   e=r-yp;

   if ireg == 1
    u=u+g0*e+g1*e_old;
    e_old=e;
   else
    u=Kp*e+z;                        % Implementering av PI-regulator.
    z=z+dt*Kp*e/Ti;                  % (integrerer med eksplisitt Euler.)
   end

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

   x = Phi*x+ Delta*u;
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')
```