Optimalisering
og
regulering

Øving 6

MTTK

Sigurd Hellesvik

SIGUHE

Vår 2019

1a) $m=1$, $F=u$ $x=\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} p \\ v \end{bmatrix}$

$\dot{p}=v$

$ma=F$

$\dot{v}=u$

$\dot{x}=A_c x + b_c u$

$$\dot{x}=\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

b) $A=e^{A_c T}$, $b=\left(\int_0^T e^{A_c \tau} d\tau\right) b_c$

$T=0,5$

$$e^{A_c t} = I + A_c t + \frac{t^2}{2!} A_c^2 + \frac{t^3}{3!} A_c^3 + \dots$$

$$A_c^2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} = 0$$

$$\Rightarrow e^{A_c t} = I + A_c t = \begin{bmatrix} 1 & 0,5 \\ 0 & 1 \end{bmatrix}$$

$$b = \int_0^T (I + A_c t) d\tau \, b$$

$$= \left(TI + \frac{1}{2} A_c T^2\right) b = \begin{bmatrix} 0,5 & 0,125 \\ 0 & 0,5 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow$$

1 b)

$$\rightarrow = \begin{bmatrix} 0{,}125 \\ 0{,}5 \end{bmatrix}$$

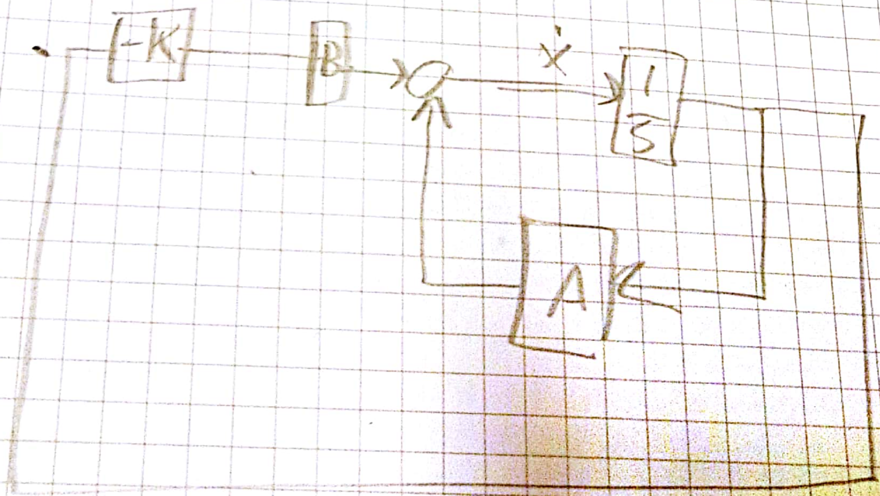
1 c)

$$Q = \begin{bmatrix} 8 & 0 \\ 0 & 2 \end{bmatrix}$$

$$P_t = Q_t + A_t^T P_{t+1} \left(I + B_t R_t^{-1} B_t^T P_{t+1}\right)^{-1} A_t$$

$$P_N = Q_N$$

$$K_t = R^{-1} b^T P_{t+1} \left(I + b R^{-1} b^T P_{t+1}\right)^{-1} A$$



~~1d)~~

1e) We use an LQ-controller.
This works if A and B
are stabilizable
and A and D are detectable

2a) $x_{t+1} = \overset{a}{3} x_t + \overset{b}{2} u_t$

$$f^{\infty}(z) = \frac{1}{2} \sum_{t=0}^{\infty} \{ \overset{q}{q} x_{t+1}^2 + \overset{R=1}{u_t^2} \}, \; q > 0$$

$$p = q + a p \left(I + b \frac{1}{r} \cdot b \cdot p \right)^{-1} a$$

$$= q + \frac{a^2 p r}{r + b^2 p}$$

$q = 2$

$$\Rightarrow p = 2 + \frac{9 \cdot p \, 1}{1 + 4p}$$

$$p + 4p^2 = 2 + 8p + 9p$$

$$p^2 - 4p - \frac{1}{2} = 0$$

$$p = 4{,}12 \;\vee\; \cancel{p = -0{,}12}$$

2b) $u = -K x_t \Rightarrow$

$K = R^{-1} b^T P (I + b R^{-1} b^T P)^{-1} A$

$= 1 \cdot 2 \cdot 4{,}12 (1 + 2^2 \cdot 4{,}12)^{-1} \cdot 3$

$= 1{,}414$

2c) same as 1e)

$\Rightarrow (A, B)$ stabilizable

and (A, D) ~~dete~~ stable

$+ Q = D^T D$

3b We get the same y_t in both 3a and 3b

This used 2 for iterations, which is better than 3b, which used 5.

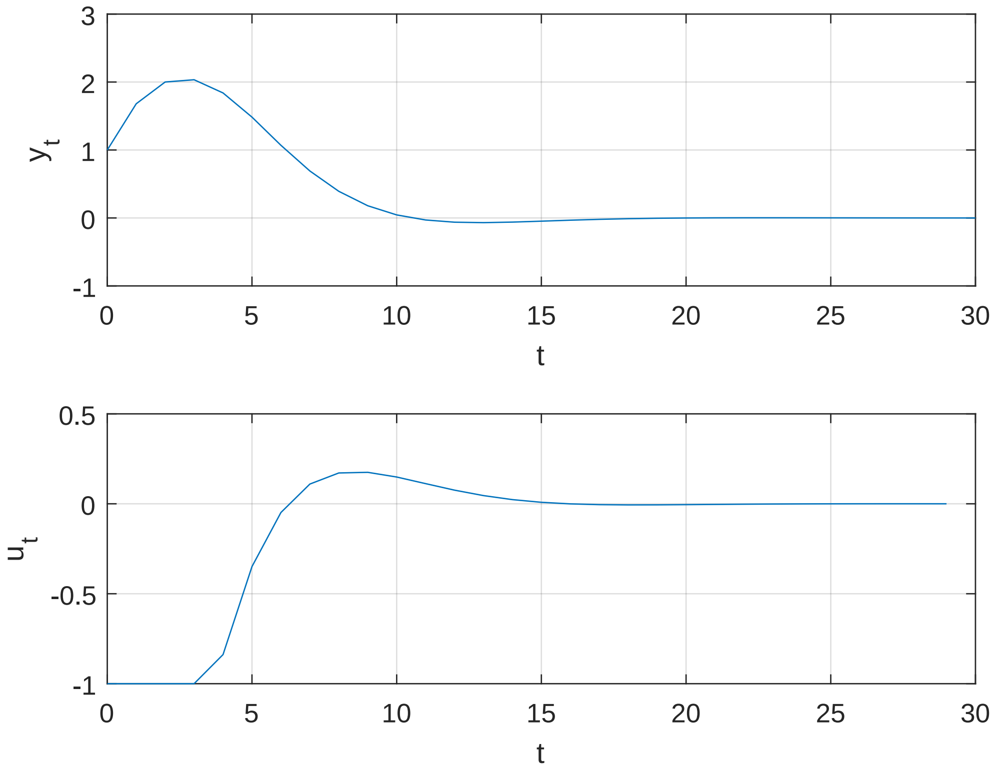
3d) With the new blocks the quadprog used 4 iterations but the system uses less u to achieve the same Y as in 3b.

Scanned by CamScanner

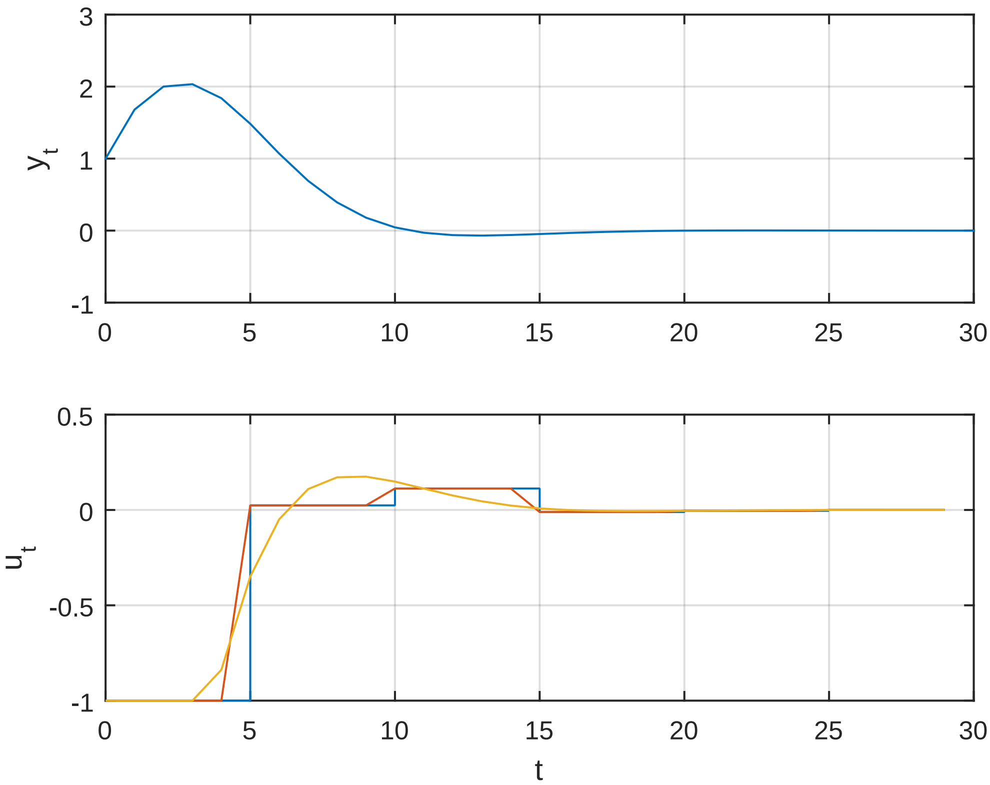
3e) We see that we get the same u and y with input blocking as without.
BUT the system use fewer iterations.

3f) From the previous tasks, we see that input blocking is very close ~~to~~ as good at regulating as non-inputblock. But it is faster to compute, since we get fewer different cases to calculate oppg1d.m

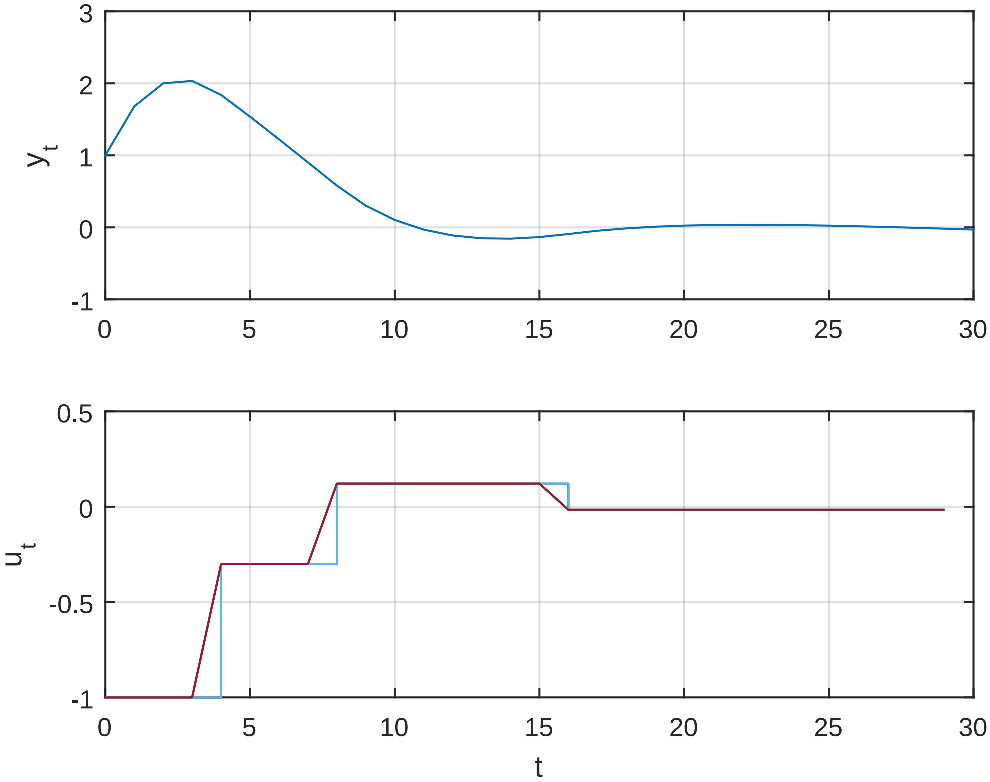
```matlab
A = [1  0.5;
     0  1  ];
B = [0.125;  0.5];
Q = diag([2 2]);
R = 2;

[K,S,e]=dlqr(A,B,Q,R,0);

S=1/2*S;

eigStab=eig(A-B*K)

%
% eigStab =
%
%    0.6307 + 0.1628i
%    0.6307 - 0.1628i
%    => Stable system
```



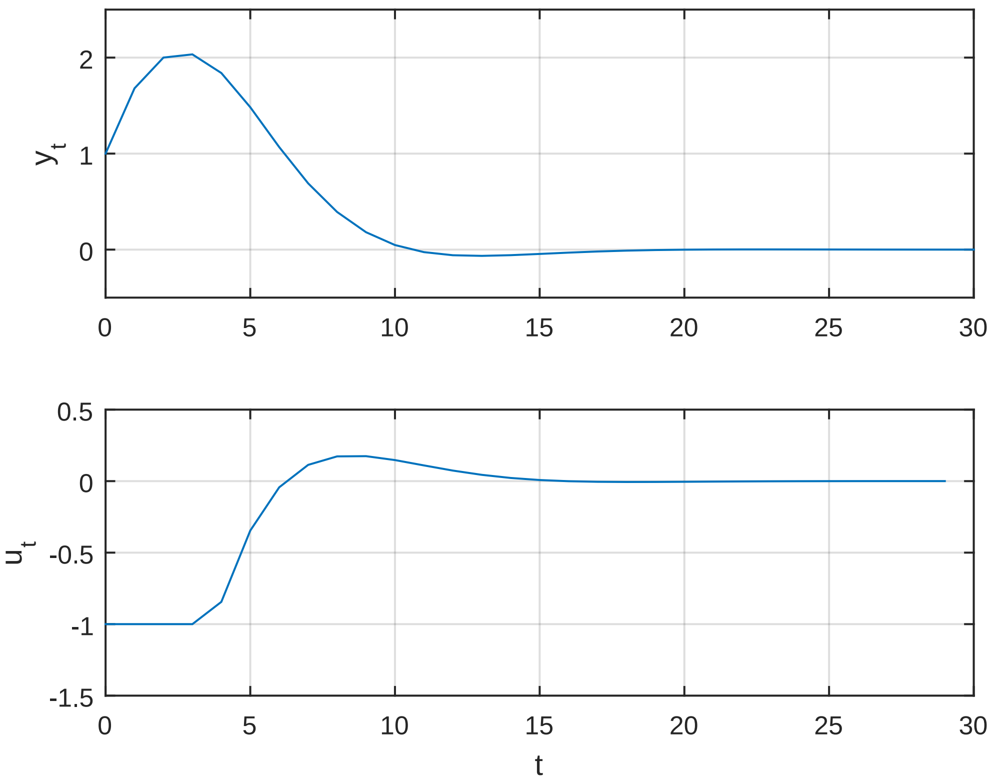
3
2
1
0
-1
y_t
0
5
10
15
20
25
30
t
0.5
0
-0.5
-1
u_t
0
5
10
15
20
25
30
t



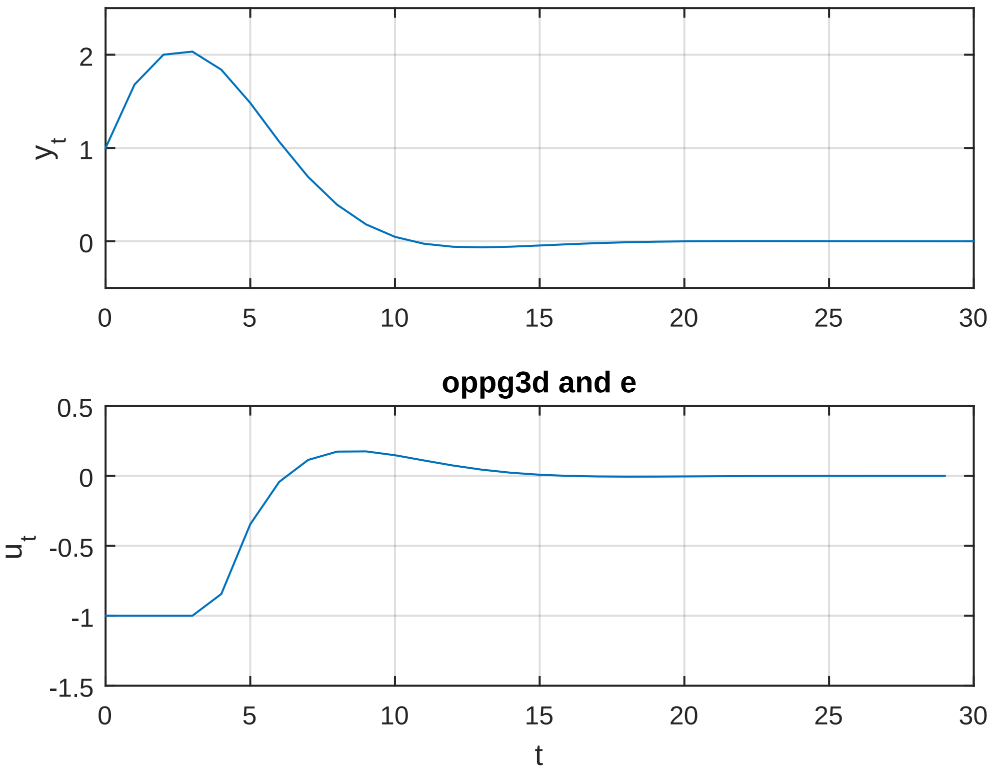
3
2
1
0
-1
y_t
0
5
10
15
20
25
30
0.5
0
-0.5
-1
u_t
t



3
2
1
0
-1
y_t
0
5
10
15
20
25
30
0.5
0
-0.5
-1
u_t
0
5
10
15
20
25
30
t



y_t
2
1
0
0
5
10
15
20
25
30
u_t
0.5
0
-0.5
-1
-1.5
0
5
10
15
20
25
30
t



2
1
0
y_t
0
5
10
15
20
25
30
oppg3d and e
0.5
0
-0.5
-1
-1.5
u_t
0
5
10
15
20
25
30
t