



- 1 a) Since $a = \dot{v} = \ddot{s}$ we get the following:

$$\begin{aligned}\dot{x}_2 &= a = F \\ \dot{x}_1 &= v = x_2 \\ \Rightarrow \dot{x} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u\end{aligned}$$

- b) First calculate $e^{A_c\tau}$:

$$\begin{aligned}e^{A_c\tau} &= \sum_{k=0}^{\infty} \frac{1}{k!} (A_c\tau)^k \\ &= I + A_c\tau + \frac{A_c^2\tau^2}{2!} + \dots + \frac{A_c^n\tau^n}{n!} \\ A_c \cdot A_c &= \mathbf{0} \\ \Rightarrow e^{A_c\tau} &= I + A_c\tau = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}\end{aligned}$$

For A we then have:

$$e^{A_cT} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix}$$

For b :

$$\begin{aligned}\int_0^T e^{A_c\tau} d\tau b_c &= \int_0^T e^{A_c\tau} b_c d\tau \\ &= \int_0^T \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} d\tau \\ &= \int_0^T \begin{bmatrix} \tau \\ 1 \end{bmatrix} d\tau \\ &= \begin{bmatrix} \frac{\tau^2}{2} \\ \tau \end{bmatrix} \Big|_{\tau=0}^{\tau=T} \\ &= \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} \\ &= \begin{bmatrix} 0.125 \\ 0.5 \end{bmatrix}\end{aligned}$$

c) The Riccati-equation is given by:

$$P_t = Q + A^\top P_{t+1} (I + bR^{-1}b^\top P_t + 1)^{-1} A$$

$$P_N = Q$$

The solution of this (P_t) is used in the definition of the feedback gain matrix K_{t-1} :

$$K_{t-1} = R^{-1}b^\top P_t (I + bR^{-1}b^\top P_t)^{-1} A$$

The feedback gain matrix is then used in the controller u_t which minimizes $f(z)$:

$$u_t = -K_t x_t$$

The solution is illustrated in Figure 1.

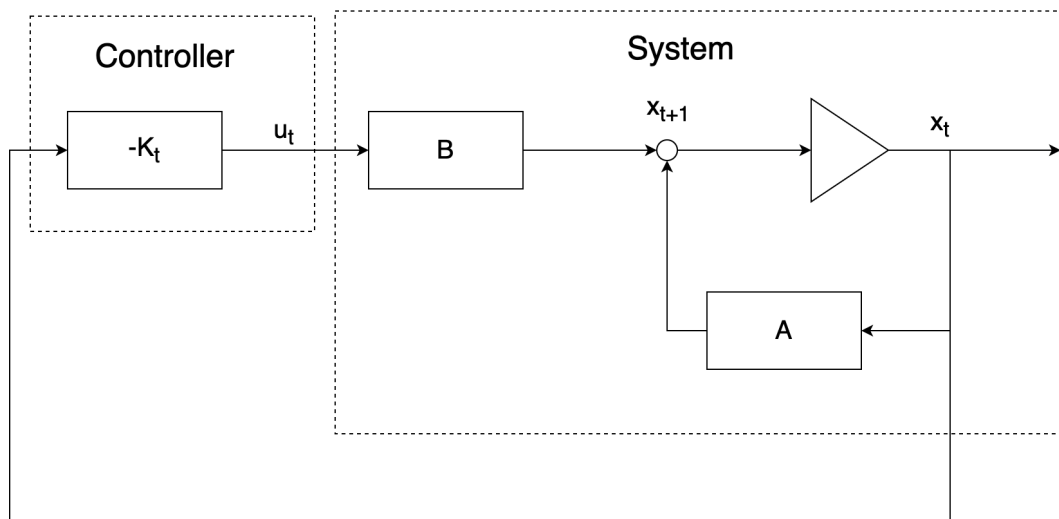


Figure 1: The system and controller.

d) To solve the stationary Riccati equation I used the following code:

```

1  Q = 1/2*[
2      2  0
3      0  2
4  ];
5  R = 2/2;
6  A = [
7      1  0.5
8      0  1
9  ];
10 B = [
11     0.125
12     0.5
13 ];
14 [K,S,e] = dlqr(A,B,Q,R,[]);

```

```

15 disp(K);
16 disp(S);
17 disp(e);

```

This gives that:

$$P = \begin{bmatrix} 4.0350 & 2.0616 \\ 2.0616 & 4.1438 \end{bmatrix}$$

$$K = \begin{bmatrix} 0.6514 \\ 1.3142 \end{bmatrix}$$

$$\lambda = 0.6307 \pm 0.1628i$$

The resulting system is stable if $|\text{eig}(A - BK)| < 1$. Since

$$|\lambda_1| = |\lambda_2| = \sqrt{0.6307^2 + 0.1628^2} = 0.6514 < 1,$$

the closed-loop system is stable.

- e) The closed-loop system is stable if the system (A, B) is stabilizable and the system (A, D) is detectable, where D is defined by $Q = D^\top D$. The first part means that the system must be stabilizable, since we have a infinite horizon. The second part means that we must be able to influence all unstable modes of the system. In order for us to influence all unstable modes we must be able to detect them in our objective function (through D).

- 2 a) The stationary one-dimensional Riccati equation is given by:

$$p = q + ap(1 + \frac{b^2 p}{r})^{-1}a$$

Some rest

$$p = q + ap(1 + \frac{b^2 p}{r})^{-1}a$$

$$p = q + a^2 p \frac{1}{1 + \frac{b^2 p}{r}}$$

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

With $q = 2$, $a = 3$, $b = 2$, $r = 1$ we get:

$$p = 2 + \frac{3^2 p}{1 + 2^2 p}$$

$$p(1 + 4p) = 2(1 + 4p) + 9p$$

$$4p^2 - 16 - 2 = 0$$

$$\Rightarrow p = \frac{16 \pm \sqrt{16^2 - 4(-2)4}}{8} = 2 \pm \frac{3}{\sqrt{2}}$$

Since p must be above 0 (see equation 4.32d in "Merging Optimization and Control"), we choose $p = 2 + \frac{3}{\sqrt{2}}$

b) The feedback gain is defined by:

$$k = \frac{bp}{r} \left(1 + \frac{b^2 p}{r}\right)^{-1} a$$

Setting in all the values and $p = 2 + \frac{3}{\sqrt{2}}$ gives us:

$$k = \sqrt{2}$$

This gives us that the optimal control law is given by:

$$u_t = -\sqrt{2}x_t$$

c) Same answer as Task 1e.

3 a) The result from the previous assignment are shown in Figure 2.

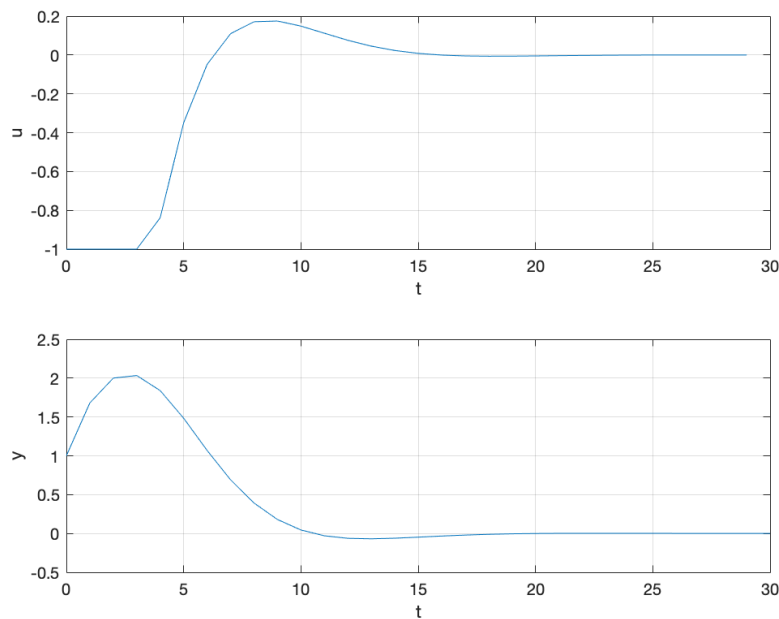


Figure 2: u_t and y_t for inequality constrained control input.

b) Our z will now only have $N/6$ control inputs. These control inputs are repeated for $n = 5$ time steps each, so we will have the same control input for 5 rows each in A_{eq} . The right part of A_{eq} will then have the following structure (simplified

by just showing the row for u_0 and u_1):

$$\begin{bmatrix} -B & 0 & 0 & 0 & 0 & 0 \\ -B & 0 & 0 & 0 & 0 & 0 \\ -B & 0 & 0 & 0 & 0 & 0 \\ -B & 0 & 0 & 0 & 0 & 0 \\ -B & 0 & 0 & 0 & 0 & 0 \\ -B & 0 & 0 & 0 & 0 & 0 \\ 0 & -B & 0 & 0 & 0 & 0 \\ 0 & -B & 0 & 0 & 0 & 0 \\ 0 & -B & 0 & 0 & 0 & 0 \\ 0 & -B & 0 & 0 & 0 & 0 \\ 0 & -B & 0 & 0 & 0 & 0 \\ 0 & -B & 0 & 0 & 0 & 0 \end{bmatrix}$$

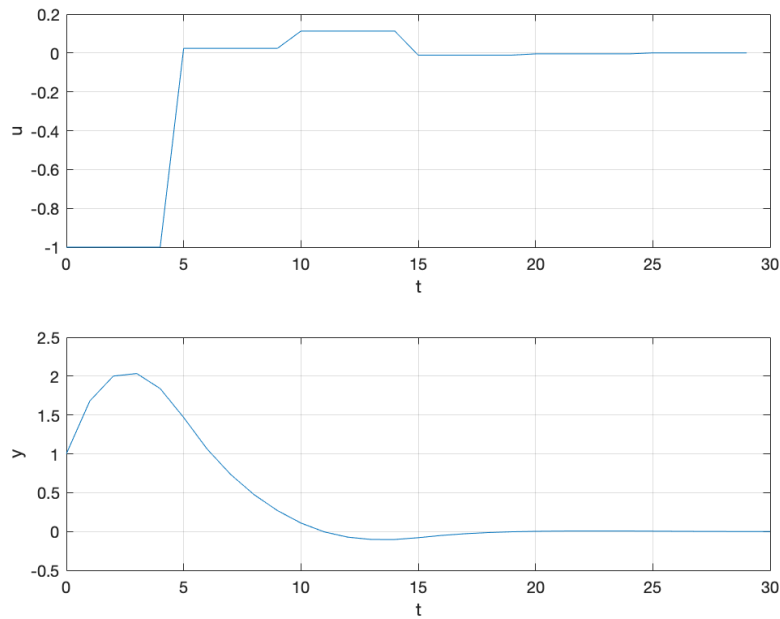
Simplifying by using $\bar{B} = \begin{bmatrix} -B \\ -B \\ -B \\ -B \\ -B \\ -B \end{bmatrix}$ gives us the following A_{eq} matrix:

$$A_{eq} = \begin{bmatrix} I & 0 & \dots & \dots & 0 & -\bar{B} & 0 & \dots & \dots & 0 \\ -A & I & \ddots & & \vdots & 0 & \ddots & \ddots & & \vdots \\ 0 & -A & \ddots & \ddots & \vdots & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 & \vdots & & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -A & I & 0 & \dots & \dots & 0 & -\bar{B} \end{bmatrix}$$

The right part will now only have 6 columns due to the reduced number of control inputs.

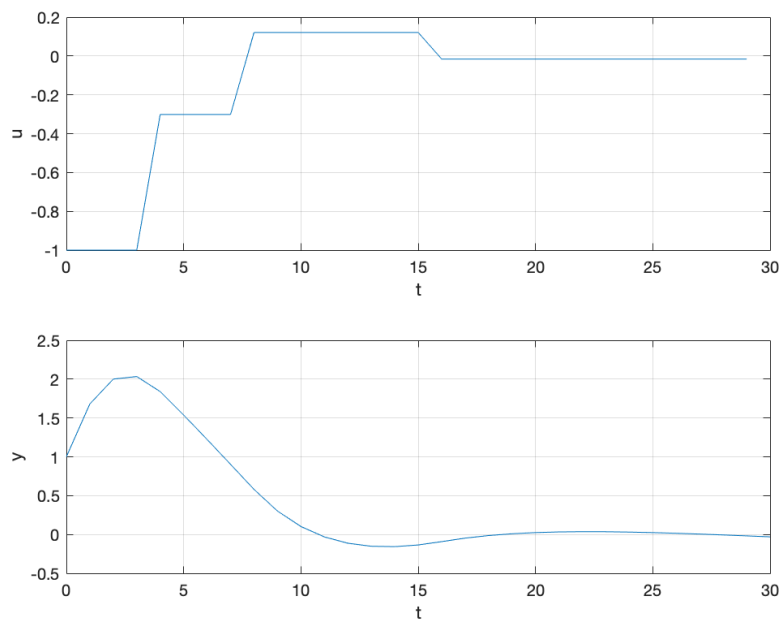
The modified matrices can be found in the file `task_3b.m`.

The results using input plotting can be found in Figure 3. As we can see the control inputs are repeated for 5 time steps each. The resulting y graph is almost the same as the one in Task 3a, even though we are input blocking and therefore have a smaller z vector.

Figure 3: u_t and y_t for even input blocking scheme.

`quadprog` still used 5 iterations with the interior-point-convex method which is the same as without input blocking. However when using the active-set method it went down from 5 iterations to 2. As we can see the choice of method used influence the results a lot.

c) The results using better input parametrization are found in Figure 4

Figure 4: u_t and y_t for parametrized input blocking scheme.

As we can see the results are pretty much the same. Even though we have given the controller more freedom in the beginning, this doesn't really matter as the control input is at the limit -1 anyways.

We still get 5 iterations using the interior-point-convex method.

- d) The results from the previous assignment are shown in Figure 5.

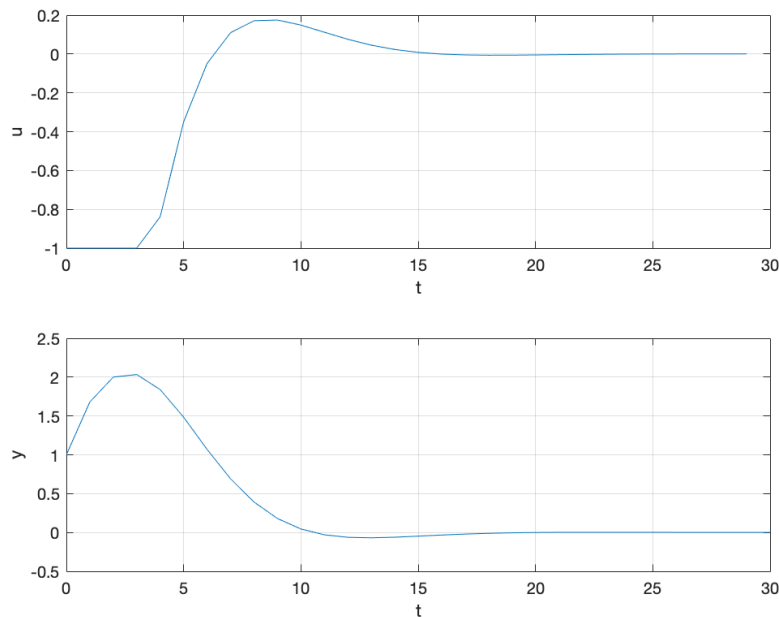


Figure 5: u_t and y_t for MPC with horizon of $n = 30$ with no input blocking.

- e) The results with input blocking in MPC are shown in Figure 6. As we can see the difference is neglectable between the MPC with no input blocking and the MPC with input blocking.

Using the interior-point-convex method the performance gain was neglectable, but with active-set method this performance gain was pretty significant, especially since we solve the optimization problem at each step.

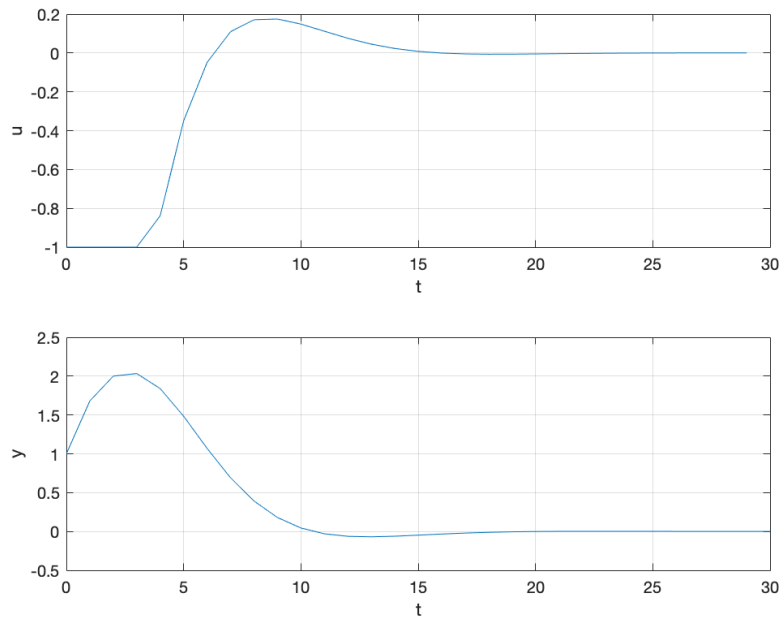


Figure 6: u_t and y_t for MPC with horizon of $n = 30$ with input blocking.

- f) The effect of using input blocking is that we allow for fewer control inputs, which leads to smaller dimensions in the matrices used in constraints, objective function etc. Using the interior-point-convex method this performance gain was neglectable in this particular problem formulation, but with active-set method the decrease of iterations could be significant given that in MPC the optimization problem is solved at each time step.

The reason for choosing input blocks of increasing length is to allow for the controller to make larger adjustments in the initial more unstable phase and when it goes towards a steady-state it can keep the same control input for longer.