MPC Assignment 1

Daniel Pihlquist danpihl@student.chalmers.se
Raman Haddad ramanh@student.chalmers.se

February 9, 2015

Unconstrained controller

In this task, the control system has no constraints and the control signal is allowed to reach whichever value the controller finds to be optimal.
To find the optimal control, an objective function is defined as

$$\begin{aligned} &\min_{u(k:k+N-1)} q \cdot ||x(k+N)||^2 + \sum_{i=0}^{N-1} \left(q \cdot y^2(k+i) + r \cdot u^2(k+i)\right) \\ &\text{s.t. } x(k+i+1) = Ax(k+i) + Bu(k+i),\ i = 0, 1, \dots, N-1 \end{aligned} \tag{1}$$

is to be rewritten to

$$\begin{aligned} &\min_z \frac{1}{2} z^T H z \\ &\text{s.t. } A_{eq} z = b_{eq} \end{aligned} \tag{2}$$

where $z = \left[x^T(k+1) \dots x^T(k+N)\ u(k) \dots u(k+N-1)\right]^T$. Since the first term $x(k)$ (current state) is known, it is omitted from the objective function. The state evolution is described by the following equations:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ x(k+2) &= Ax(k+1) + Bu(k) \\ &\vdots \\ x(k+N) &= Ax(k+N-1) + Bu(k+N-1) \\ &\Longleftrightarrow \\ x(k+1) &= Ax(k) + Bu(k) \\ x(k+2) &= A^2x(k) + ABu(k) + Bu(k+1) \\ x(k+2) &= A^3x(k) + A^2Bu(k) + ABu(k+1) + Bu(k+2) \\ &\vdots \\ x(k+N) &= A^{N+1}x(k) + \sum_{i=0}^{N-1} A^{N-i}Bu(k+i) \end{aligned} \tag{3}$$

The matrix H which was used in the implementation has the following structure, with $N = 5$:

$$H = \begin{bmatrix} qC_c & 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} \\ 0_{n\times n} & qC_c & 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} \\ 0_{n\times n} & 0_{n\times n} & qC_cq & 0_{n\times n} & 0_{n\times n} & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} \\ 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & qC_c & 0_{n\times n} & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} \\ 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & qI_{n\times n} & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} \\ 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & r & 0 & 0 & 0 & 0 \\ 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0 & r & 0 & 0 & 0 \\ 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0 & 0 & r & 0 & 0 \\ 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0 & 0 & 0 & r & 0 \\ 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0 & 0 & 0 & 0 & r \end{bmatrix}, \tag{4}$$

where $C_c = C^T C$. The equality constraint matrix:

$$A_{eq} = \begin{bmatrix} I_{n\times n} & 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & -B & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} \\ 0_{n\times n} & I_{n\times n} & 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & -AB & -B & 0_{n\times 1} & 0_{n\times 1} & 0_{n\times 1} \\ 0_{n\times n} & 0_{n\times n} & I_{n\times n} & 0_{n\times n} & 0_{n\times n} & -A^2B & -AB & -B & 0_{n\times 1} & 0_{n\times 1} \\ 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & I_{n\times n} & 0_{n\times n} & -A^3B & -A^2B & -AB & -B & 0_{n\times 1} \\ 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & I_{n\times n} & -A^4B & -A^3B & -A^2B & -AB & -B \\ 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0 & 0 & 0 & 0 & 0 \\ 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0 & 0 & 0 & 0 & 0 \\ 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0 & 0 & 0 & 0 & 0 \\ 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0 & 0 & 0 & 0 & 0 \\ 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0_{1\times n} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5}$$

With the corresponding b_{eq} vector:

$$b_{eq} = \begin{bmatrix} Ax(k) \\ A^2x(k) \\ A^3x(k) \\ A^4x(k) \\ A^5x(k) \\ 0_{N\times 1} \end{bmatrix} \tag{6}$$

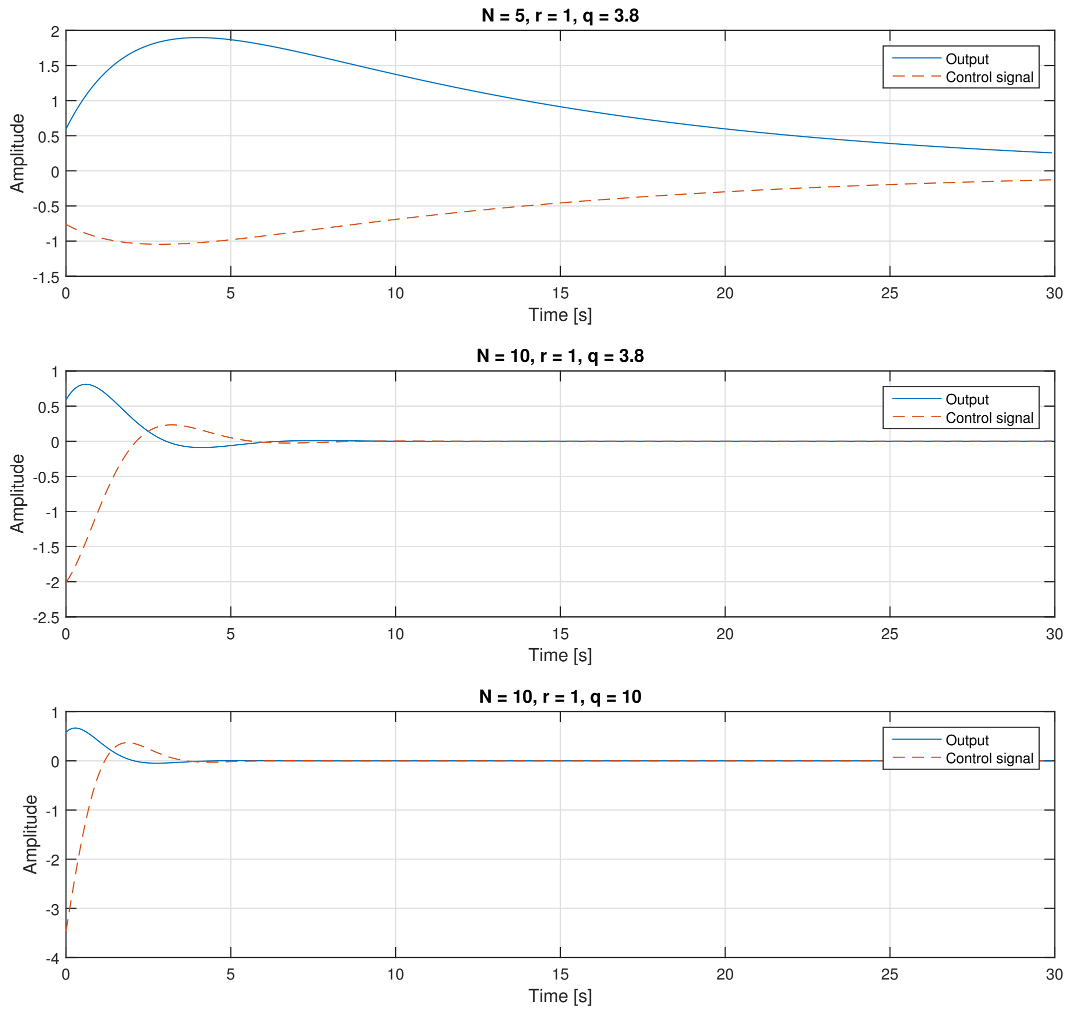

Figure 1: Simulation with unconstrained control signals

As we can se in figure 1, the controller where $N = 5$ is unable to see far enough into the future and is therefor not as fast as the controllers where $N = 10$. We also see that the state is corrected faster for when $q = 10$ than when $q = 3.8$.
Figure 2 shows the simulation of the equivalent LQ controller when $r = 1$ and $q = 3.8$.

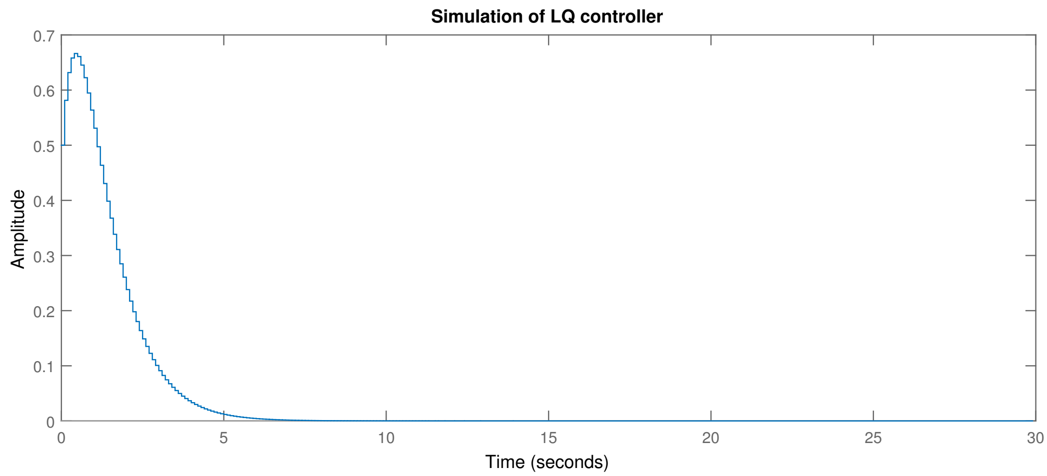

Figure 2: Finite horizon LQ controller

Constrained controller

In the case where the control signal output is limited, the following constrain is added to the optimization:

$$-1 \leq u(k+1) \leq 1,\ i = 0, 1, \dots N-1 \tag{7}$$

The quadratic programming allows for such a constraint, in the form of

$$A_{in} z \leq b_{in} \tag{8}$$

The constraint is rewritten to $u(k+i) \leq 1 \land -u(k+1) \leq 1$ in order to satisfy the matrix equation, which puts A_{in} into the following form:

$$A_{in} = \begin{bmatrix} 0_{N\times nN} & I_{N\times N} \\ 0_{N\times nN} & 0_{N\times N} \\ 0_{N\times nN} & -I_{N\times N} \end{bmatrix}, \tag{9}$$

and b_{in} to

$$b_{in} = \begin{bmatrix} 1_{N\times 1} \\ 0_{N\times 1} \\ 1_{N\times 1} \end{bmatrix} \tag{10}$$

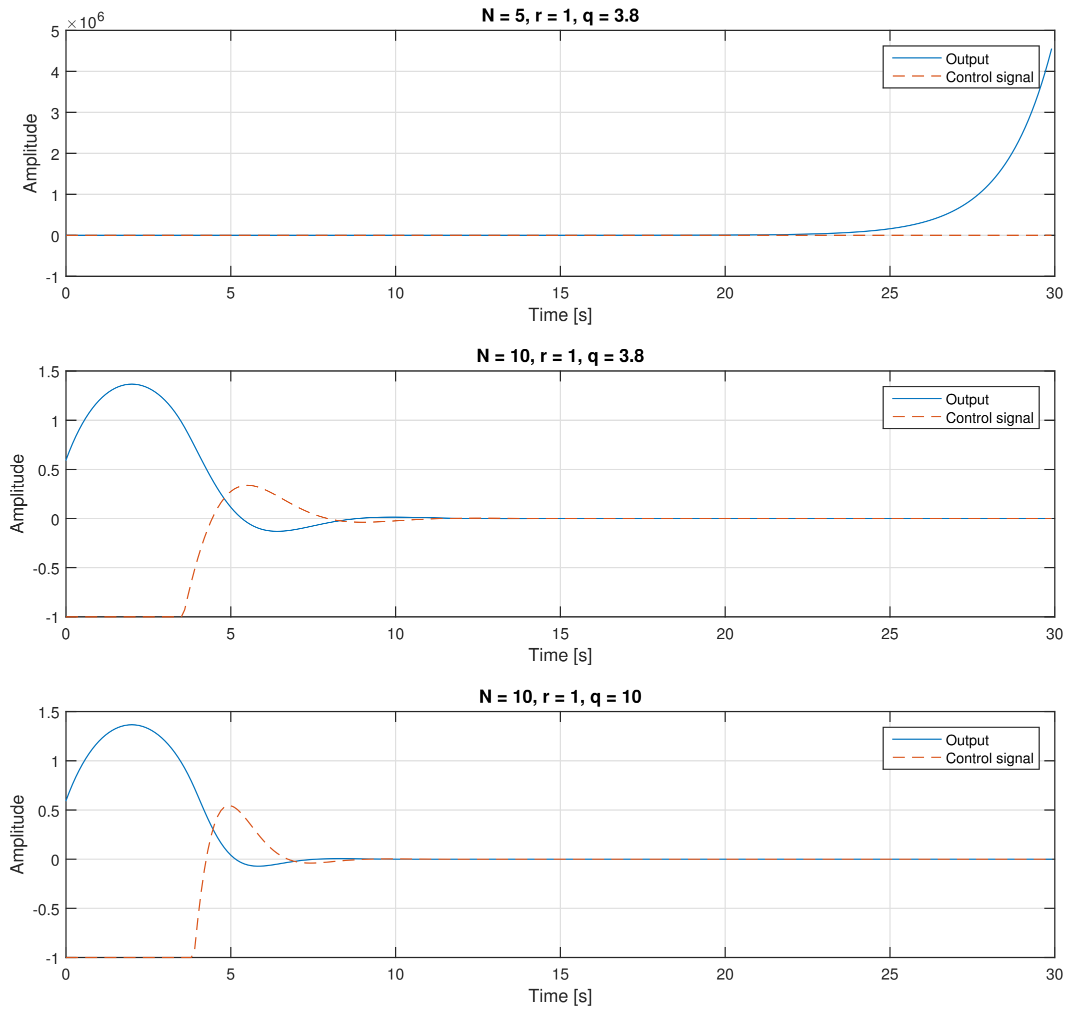

Figure 3: Simulation with constrained control signals

Here we see that the controller where $N = 5$ is unable to compensate for the high initial condition for the angular velocity. It doesn't see far enough into the future, which makes it fail to compensate for the escalation. The controllers where $N = 10$ are however able to see that things are about to break down, and saturates the control signal, which is enough compensation for the output to swing back to 0. The difference between the two cases of q isn't particularly large; one can observe a slightly faster recovery of the output when $q = 10$, which makes the cost of the state higher.
Since this is a linearized model, the model breaks down beyond a certain angle, and the output increases towards infinity. In reality, the pendulum would simply fall down and swing around 180 degrees.

Execution time

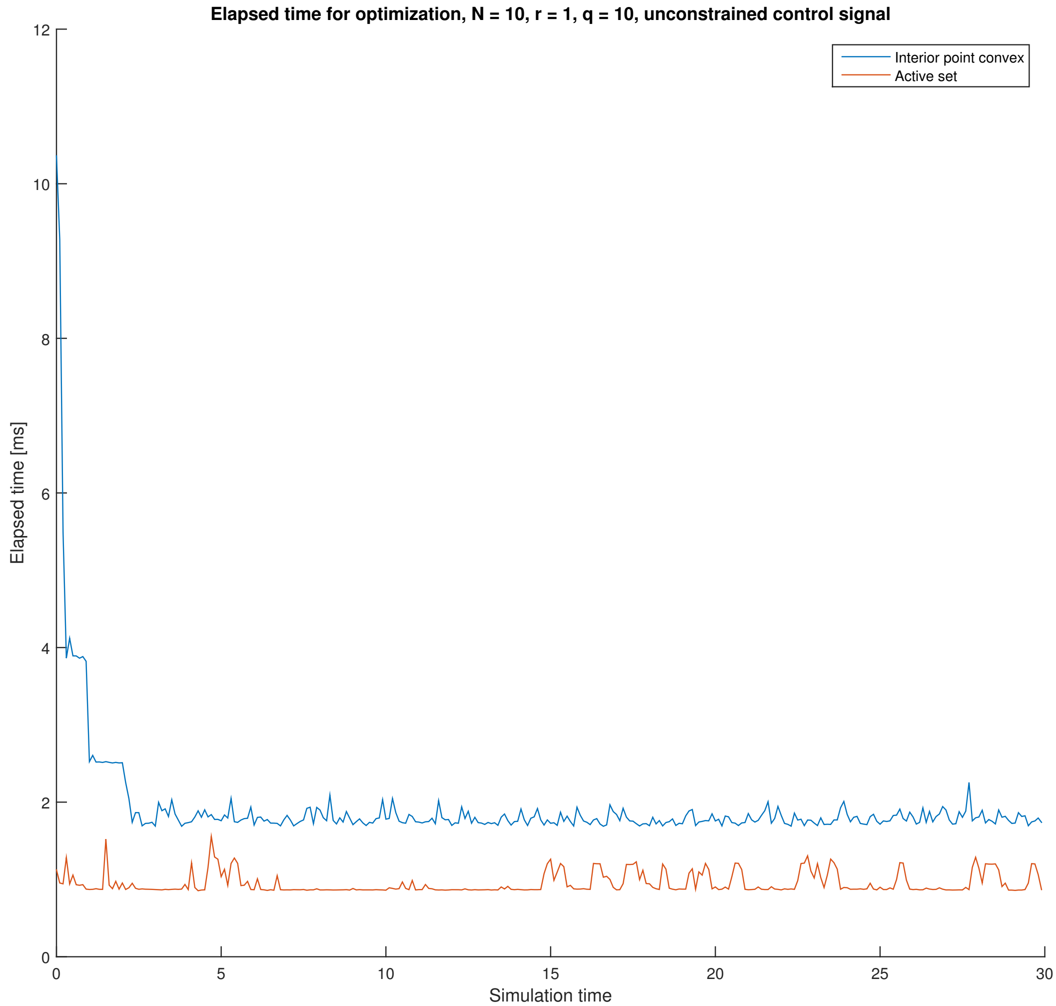

Figure 4: Unconstrained

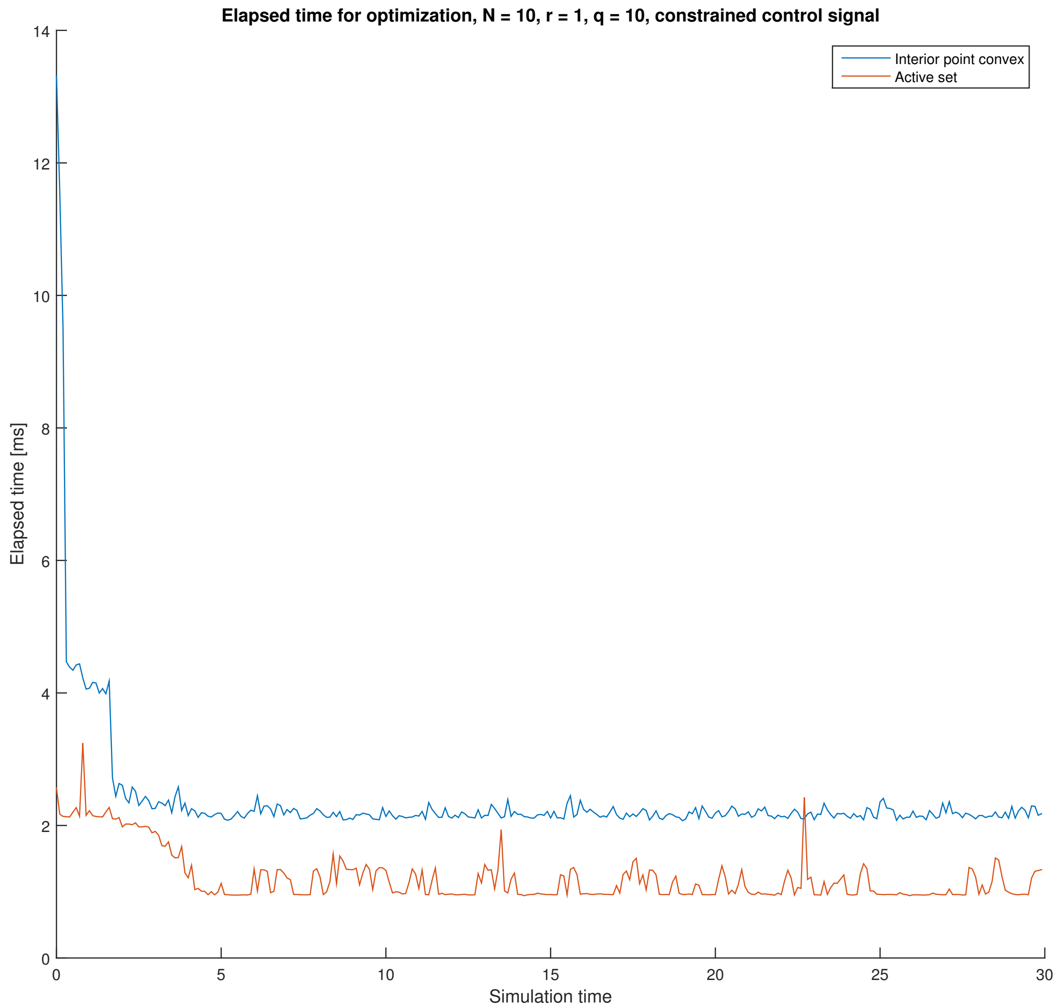

Figure 5: Constrained control signal

The *interior point convex* is always slower than the *active set* algorithm. The *interior point convex* also has a initial computational time which is very high, comparing to the steady state. One thing to note is that over several simulations, the location of the peaks in the execution time vary. This is most likely because of the fact that Matlab runs on an operating system which isn't a real time operating system. The general characteristics are however the same.

Reducing the number of optimization variables

Now the task was to optimize only with respect to the control signal, in the following way:

$$\begin{aligned}\min_{u_N} \frac{1}{2}u_N^T H u_N + f^T u_N \\ \text{s.t. } A_{eq}u_N = b_{eq} \\ A_{in}u_N \leq b_{in}\end{aligned} \tag{11}$$

Since there are no constraints on the control signal, other than it being between -1 and 1, the equality constraint can be set to 0. In the previous task, the state variables also acted as optimization variables, which posted some constraints on the optimization. If for example $x(k+3)$ was set to a certain value by the optimization solver, the future and past states are no longer arbitrary; they have to follow the state dynamics, i.e. the equality constraint. This is however no longer the case when u_N is the optimization variable, which is why $A_{eq} = 0_{1\times N}$ and $b_{eq} = 0_{N\times 1}$. The inequality constraint is still present and almost the same as in previous tasks, only less number of columns:

$$A_{in} = \begin{bmatrix} I_{N\times N} \\ 0_{N\times N} \\ -I_{N\times N}\end{bmatrix}, \tag{12}$$

and b_{eq} to

$$b_{in} = \begin{bmatrix} 1_{N\times 1} \\ 0_{N\times 1} \\ 1_{N\times 1}\end{bmatrix} \tag{13}$$

For the H-matrix and f-vector, we simply express all the future terms of the states with $x(k)$ and u_N. Since terms with only $x(k)$ and no factors from u_N are constant, they do not move the global minimum of the objective function, and can therefor be neglected. For $N = 3$:

$$H = \begin{bmatrix} H_{1,1} & 0 & 0 \\ H_{2,1} & H_{2,2} & 0 \\ H_{3,1} & H_{3,2} & H_{3,3}\end{bmatrix} \tag{14}$$

where

$$H_{1,1} = r + B^T C_c B + B^T A^T C_c AB + B^T A^{2T} A^2 B \tag{15}$$

$$H_{2,2} = r + B^T C_c B + B^T A^T AB \tag{16}$$

$$H_{3,3} = r + B^T B \tag{17}$$

$$H_{2,1} = B^T A^T C_c B + B^T C_c AB + B^T A^{2T} AB + B^T A^T A^2 B \tag{18}$$

$$H_{3,1} = B^T A^{2T} B + B^T A^2 B \tag{19}$$

$$H_{3,2} = B^T A^T B + B^T AB \tag{20}$$

$$f = \begin{bmatrix} f_1 \\ f_2 \\ f_3\end{bmatrix} \tag{21}$$

where

$$f_1 = {x_0}^T A^T C_c B + B^T C_c A x_0 + x_0 A^{2T} C_c AB + B^T A^T C_c A^2 x_0 + {x_0}^T A^{3T} A^2 B + B^T A^{2T} A^3 x_0 \tag{22}$$

$$f_2 = {x_0}^T A^{2T} C_c B + B^T C_c A^2 x_0 + {x_0}^T A^{3T} A B + B^T A^T A^3 x_0 \tag{23}$$

$$f_3 = {x_0}^T A^{3T} B + B^T A^3 x_0 \tag{24}$$