SSY281 - Model Predictive Control
Assignment A02 - Linear Quadratic and Receding Horizon Control

Lucas Rath - **Group 09**

Question 1

The finite-time LQ controller using a Dynamic Programming fashion can be implemented by solving the Ricatti equations (19) of the lecture notes, for a horizon N and then receding backward until the actual time-step.

Question 2

The shortest horizon length can be found by increasing N iteratively until all the close-loop eigenvalues are stable: $|\lambda(A - B * K0)| < 1 \forall i$.

Listing 1: *Finite DP-LQ controller with the shortest horizon length that results in a stable close-loop*

```
N2 =
    4

P0 =
   25.3097    5.0183
    5.0183    5.8215

K0 =
    0.8013    0.9150

eig(A-B*K0) =           % abs(eig_i) < 1 for all i
   0.9547 + 0.0311i
   0.9547 - 0.0311i
```

Question 3

The stationary solution Pf3 of the Ricatti equation was calculated and it is given below.

Listing 2: *Dynamic programming solution for the finite LQ controller with the short1est horizon length that results in a stable close-loop*

```
Pf3 =
   51.7051   18.5517
   18.5517   15.9083

N3 =
     1
```

The mainly difference observed in comparison with the last question is that now the minimum horizon length is N3=1, i.e. just 1 step is enough to result in a controller that makes the close-loop system stable.

This happens because we have calculated a final cost function Pf that is the stationary solution of the Ricatti equation. In this case, no matter N, the result will always be the same, which is equal to the infinite-horizon solution, because P is already stationary since the first backward step.

Question 4

The batch solution has been implemented in Matlab, with the help of equations (21), (22) and the two consecutive formulas describing the optimal control vector and the optimal cost-to-go.

Question 5

As expected, the results obtained using the batch solution match exactly with the dynamic programming solution:

Listing 3: *Batch solution for the finite LQ controller with the shortest horizon length that results in a stable close-loop*

```
N5 =
    4

P0 =
   25.3097    5.0183
    5.0183    5.8215

K0 =
    0.8013    0.9150

eig(A-B*K0)  =            % abs(eig_i) < 1 for all i
   0.9547 + 0.0311i
   0.9547 - 0.0311i
```

Question 6

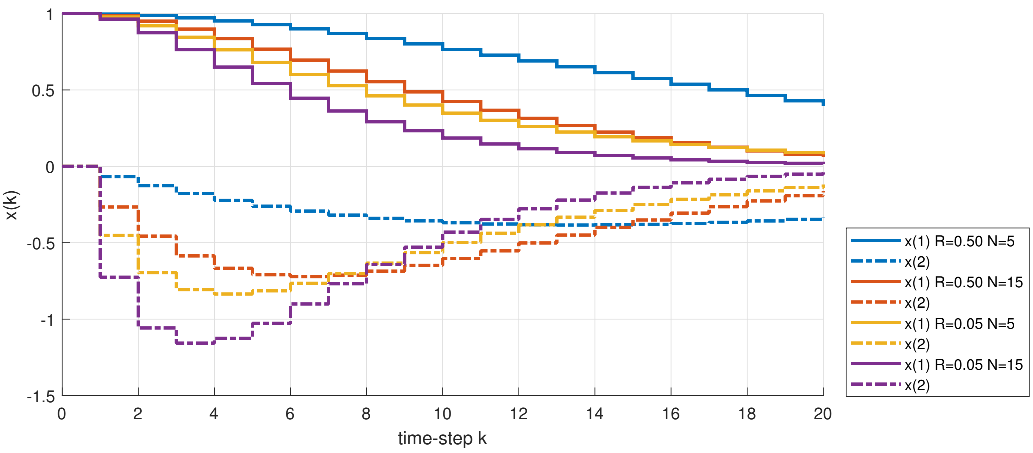

Figure 1: *Plant outputs for receding horizon control with different parameters*

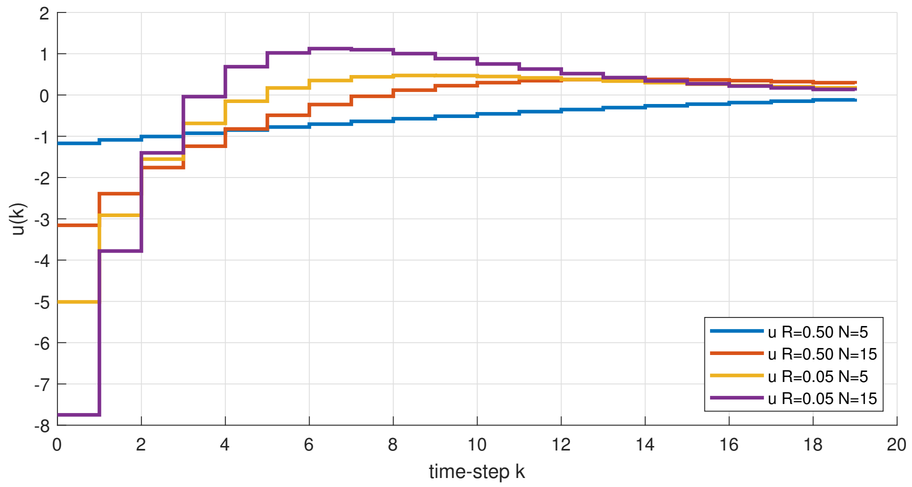

Figure 2: *Plant inputs for receding horizon control with different parameters*

As can be seen in figures 1 and 2, different N and R parameters will change completely the close-loop response. The more R is increased, more we penalize higher inputs. As a result, the system will take longer to stabilize to the zero states but will present smaller input magnitudes. On the other hand, as the horizon length N is increased, the poles of the system become faster and more control input is applied to the system. Therefore the stabilization becomes faster.

Question 7

The optimal solution for the constrained receding horizon control was implemented based on the algorithm presented in the page 42 of the lecture notes. The only difference is the Matlab required notation. The variable to be optimized is now z and the cost function must be defined as a matrix form instead of a summation. The proper implementation can be seen in the Matlab file.

Question 8

The implementation of the batch solution for the constrained receding horizon control can be found in the Matlab file. The cost function was defined according to equation (22) and the system dynamics equation (21) was used to replace x in the constraint equations.

Question 9

As expected, both CRHC1.m or CRHC2.m led to the exactly same results, which can be seen in figures 3 and 4. The only tricky part of this exercise is how the constraints in absolute terms can be defined to be accepted by the Matlab solver:

$$|x_2| \leq 0.5 \quad = \quad \begin{cases} x_2 \leq +0.5 \\ x_2 \geq -0.5 \end{cases} \quad = \quad \begin{cases} x_2 \leq 0.5 \\ -x_2 \leq 0.5 \end{cases} \tag{1}$$

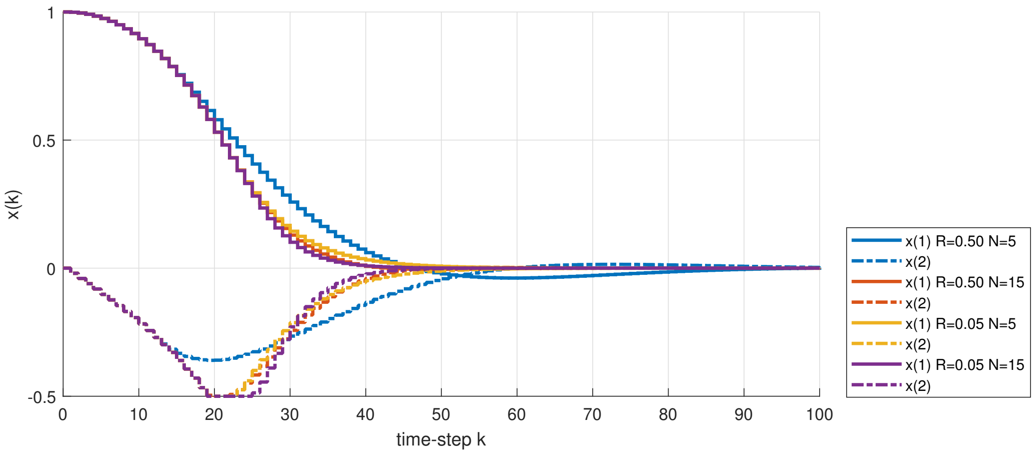

Figure 3: *Plant outputs for constrained receding horizon control with different parameters*

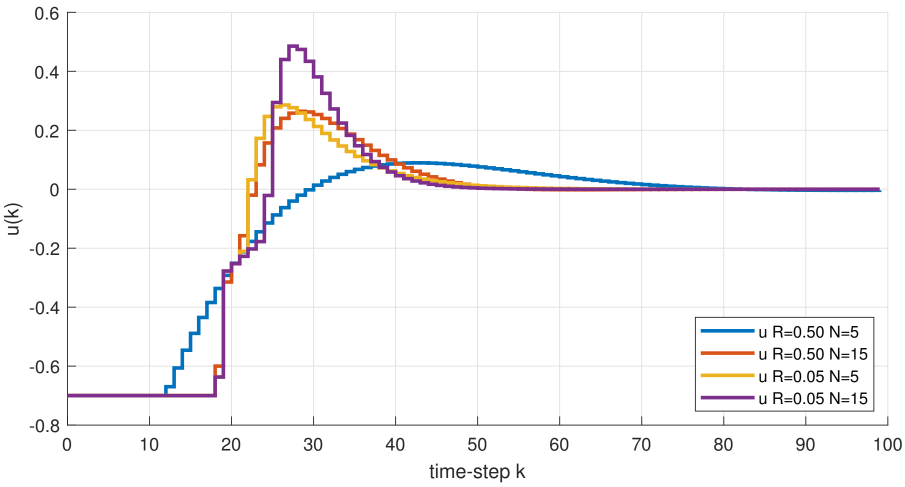

Figure 4: *Plant inputs for constrained receding horizon control with different parameters*