# SSY281
# Assignment 2
# Linear Quadratic and Receding Horizon Control

Manish Suvarna

February 5, 2020

## Dynamic Programming Solution of the LQ problem

## 1 Question 1

For the given discrete system and its cost function as shown in Equation 1 and 2

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{1}$$

$$V_N(x(0), u(0:N-1)) = \sum_{k=0}^{N-1} (x(k)^T Q x(k) + u(k)^T R u(k) + x(N)^T P_f x(N) \tag{2}$$

Using the solution of Riccati's equation obtained using the values of the matrices Q, R and $P_f$, a finite time LQ controller is designed using MATLAB in the provided template DP_57.m.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{3}$$

Where,

$$A = \begin{bmatrix} 1.0025 & 0.1001 \\ 0.05 & 1.0025 \end{bmatrix} \quad B = \begin{bmatrix} 0.005 \\ 0.1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{4}$$

## 2 Question 2

The values of the LQ controller gain obtained using the function DP_57.m was found to be, see Equation 5 and the shortest value of N such that the eigen values of the matrix $(A + B * K0)$ were within the unit circle was found by checking the values of these eigen values starting from $N = 1$. At $N = 4$, the system was found to have its closed loop poles within the unit circle in the complex plane for the first time as shown in 6.

$$K0 = \begin{bmatrix} -0.8012 & -0.9150 \end{bmatrix} \tag{5}$$

$$\lambda_i(A + B \cdot K0) = \begin{bmatrix} 0.9547 + 0.031\mathrm{i} \\ 0.9547 - 0.031\mathrm{i} \end{bmatrix} \tag{6}$$

## 3 Question 3

The stationary solution of the Riccati equation was found by using the command 'dare' in MATLAB which resulted in the terminal cost matrix as shown in Equation 7. This matrix was then used in the MATLAB template DP_57.m to calculate the LQ controller gain, $K0_3$ as shown in 8 which was used to check the stability of the system by finding the closed loop poles of the matrix $(A + B * K0_3)$. These poles were found to be stable at the step $N = 1$.

$$Pf_3 = \begin{bmatrix} 51.70 & 18.55 \\ 18.55 & 15.94 \end{bmatrix} \tag{7}$$

$$K0_3 = \begin{bmatrix} -3.241 & -2.798 \end{bmatrix} \tag{8}$$

The difference in the shortest step at which the system becomes stable as compared to the previous question is because the command 'dare' gives a unique stabilizing stationary solution of the discrete algebraic Riccati equation, which implies that at every step, the stage cost remains the same. Hence the final stage cost is achieved just in one step, since it was the same in the previous step as well.

# Batch solution of the LQ problem

## 4 Question 4

The batch solution equations were obtained with an attempt to solve the optimal cost to go functions from initial state over the length of horizon N, based on the initial state values and implementing them in the MATLAB template file BS_57.m. The LQ controller gain calculated using batch solution was found to be as shown in Equation 9

$$K0_5 = \begin{bmatrix} -0.8012 & -0.915 \\ -0.4548 & -0.641 \\ -0.209 & -0.396 \\ -0.0591 & -0.1821 \end{bmatrix} \tag{9}$$

## 5 Question 5

Using the controller gain obtained in the batch solution as shown in Equation 9, the shortest value of horizon length at which the system was found to be stable was at the value $N = 4$ with its closed loop poles being at, see Equation 10

$$\lambda_i(A + B \cdot K0_5) = \begin{bmatrix} 0.960 + 0.0380\text{i} \\ 0.960 - 0.0380\text{i} \end{bmatrix} \tag{10}$$

One can observe that the shortest required step at which the stability of the system is asymptotically obtained and the system's closed loop poles are almost the same values as obtained by Dynamic programming as shown in Equation 5 and 6.

# Receding Horizon Control

## 6 Question 6

The system outputs and inputs for the four RHC controllers with the following parameters as shown in 11 were obtained by simulating the closed loop system over a timeline of 20 seconds as shown in Figures 1 and 2

$$\begin{aligned} 1.R = 0.5 &\quad N = 5 \\ 2.R = 0.5 &\quad N = 15 \\ 3.R = 0.05 &\quad N = 5 \\ 4.R = 0.05 &\quad N = 15 \end{aligned} \tag{11}$$

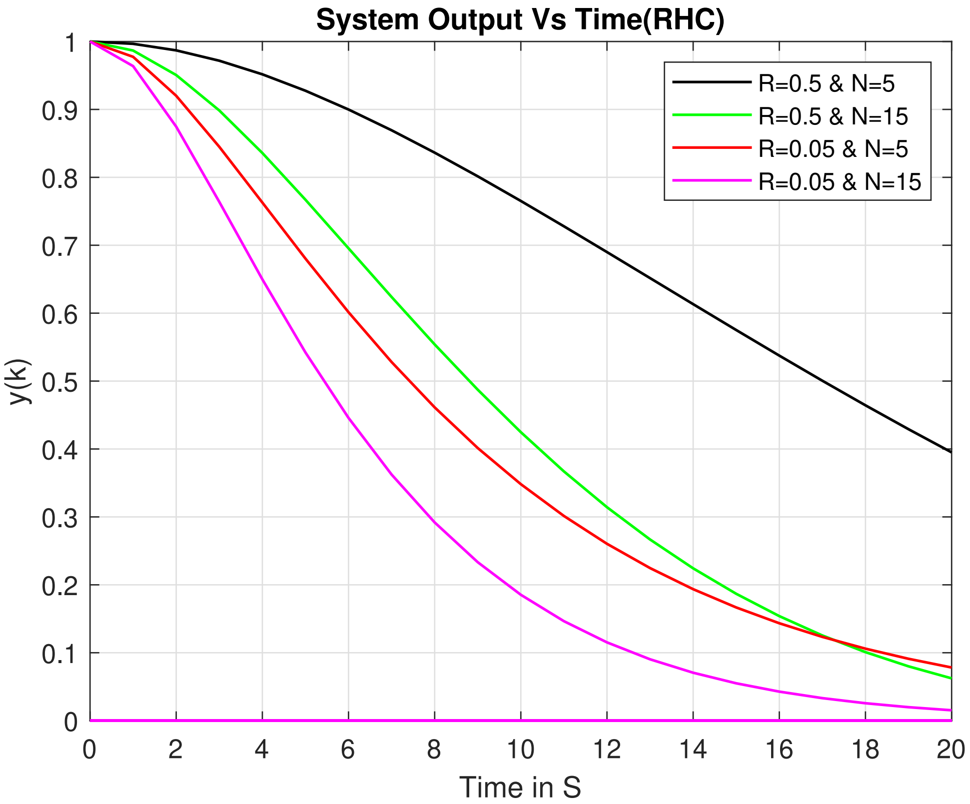


*Figure 1: System output using RHC for different tuning parameters*

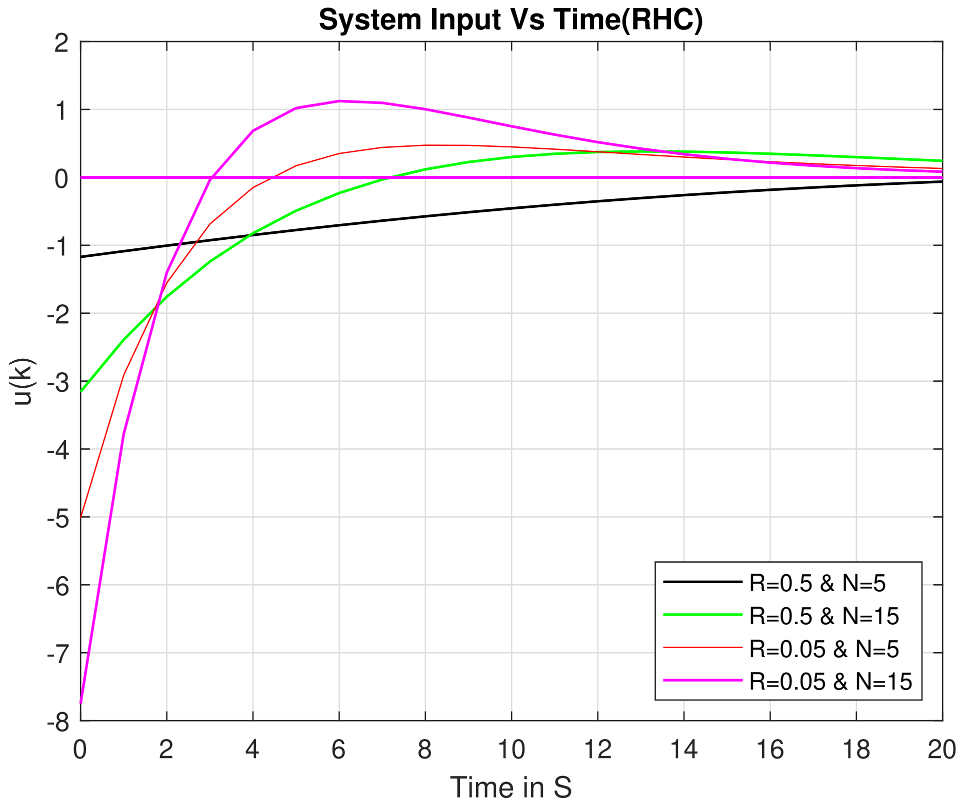


*Figure 2: System inputs using RHC for different tuning parameters*

From these figures, one can observe that the higher the values of R implies higher penalty function on the control signal resulting in more time for the output response to stabilize. Increasing N results in more stable feedback inputs to the system from the batch solution, resulting in faster stabilization.

# Constraint Receding Horizon Control

## 7 Question 7

The constrained RHC implementation can be seen in the MATLAB template CRHC1_57.m file using quadprog command. The equations in this template were implemented as derived in the Lecture 4, for the optimization variable as

$$z = \begin{bmatrix} x \\ u \end{bmatrix} \tag{12}$$

## 8 Question 8

The change in optimization variables as $z = u$ causes the change in the cost to go function along with the constraints as described in the algorithm of CRHC2_57.m

## 9 Question 9

The constraints for the state $x_2(k)$ and control signal u(k) is as mentioned in the equation 13. The system inputs and outputs were plotted against time along with constraints involved in the dynamics using CRHC1_57.m as shown in the Figures 3 and 4 for different tuning parameters as mentioned in 11

$$| x_2(k) | \leq 0.5 \quad | u(k) | \leq 0.7 \tag{13}$$

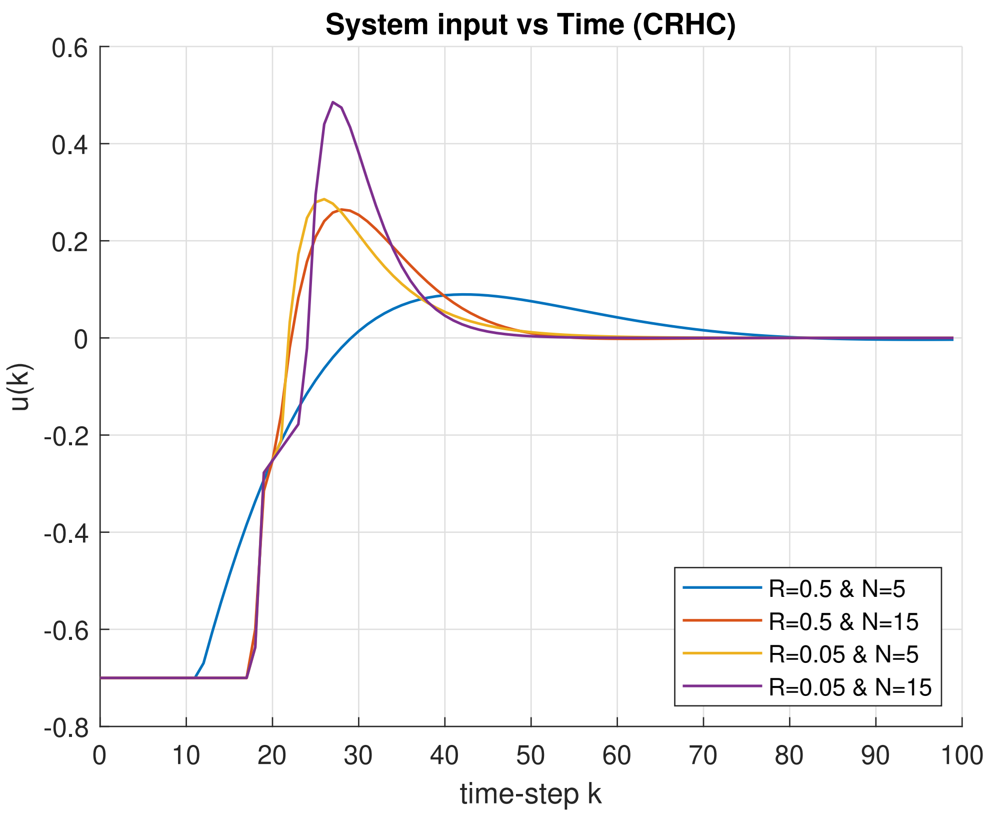


*Figure 3: System inputs using CRHC for different tuning parameters*

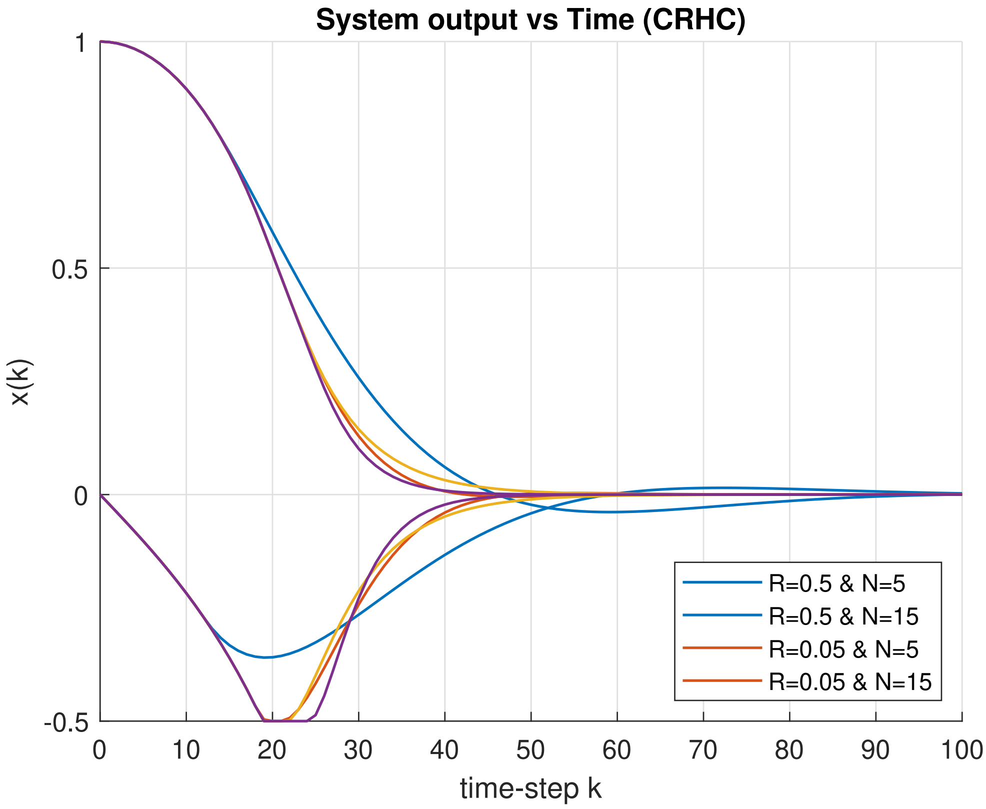


*Figure 4: System outputs using CRHC for different tuning parameters*