SSY281 Model Predictive Control

Assignment A1

Qun Zhang)

February 3, 2023

Question 1

(a) Since a ball-wheel system is controlled by a DC motor with an external force u, then the equation could be expressed below:

$$\dot{x} = \underbrace{\begin{bmatrix} x_2 \\ ax_4 + bsin(x_1) \\ x_4 \\ px_4 + qsin(x_1) \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} 0 \\ c \\ 0 \\ r \end{bmatrix}}_{g(x)} u \tag{1}$$

Through viewing the $\theta_1, \dot{\theta_1}, \theta_2, \dot{\theta_2}$ as sate space x_1, x_2, x_3, x_4, then we can get our state-space equation related to ball-wheel system. Additionally, since the angle is very small, we can assume $x \approx sinx$.

$$\underbrace{\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix}}_{\dot{X}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ b & 0 & 0 & a \\ 0 & 0 & 0 & 1 \\ q & 0 & 0 & p \end{bmatrix}}_{A_c} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{X} + \underbrace{\begin{bmatrix} 0 \\ c \\ 0 \\ r \end{bmatrix}}_{B_c} u \tag{2}$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}}_{C_c} X \tag{3}$$

Taking characters a = 0.9421, b = 82.7231, c = 14.2306, p = 3.7808, q = 4.9952, r = 57.1120 and h=0.1s into the equation and using the matlab function c2d, then we can get the state-space equation of discrete time.

$$\underbrace{\begin{bmatrix} x_1(K+1) \\ x_2(K+1) \\ x_3(K+1) \\ x_4(K+1) \end{bmatrix}}_{X(K+1)} = \underbrace{\begin{bmatrix} 1.4421 & 0.1143 & 0 & -0.0045 \\ 9.4370 & 1.4421 & 0 & -0.0908 \\ 0.0237 & 0.0008 & 1.0000 & 0.0833 \\ 0.4814 & 0.0237 & 0 & 0.6845 \end{bmatrix}}_{A_d} \underbrace{\begin{bmatrix} x_1(K) \\ x_2(K) \\ x_3(K) \\ x_4(K) \end{bmatrix}}_{X(K)} + \underbrace{\begin{bmatrix} 0.0677 \\ 1.3715 \\ 0.2530 \\ 4.7660 \end{bmatrix}}_{B_d} u \tag{4}$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}}_{C_d} X \tag{5}$$

(b) The new system has 0.8h (h=0.1s) delay, then calculation of the new system have to follow the equation shown below:

$$\xi(K+1) = \begin{bmatrix} X(K+1) \\ u(K) \end{bmatrix} = \begin{bmatrix} A & B_1 \\ 0 & 0 \end{bmatrix} \xi(K)_+ \begin{bmatrix} B_2 \\ I \end{bmatrix} u(K) \tag{6}$$

The characters in the matrix could be calculated with equations shown below:

$$A = A_{h\tau} A\tau, B_1 = A_{h\tau} B_\tau, B_2 = B_{h\tau} \tag{7}$$

Where:

$$\begin{bmatrix} A_\tau & B_\tau \\ 0 & I \end{bmatrix} = e^{\begin{bmatrix} A_c & B_c \\ 0 & 0 \end{bmatrix}}, \quad \begin{bmatrix} A_{h\tau} & B_{h\tau} \\ 0 & I \end{bmatrix} = e^{\begin{bmatrix} A_c & B_c \\ 0 & I \end{bmatrix}^{(h-\tau)}} \tag{8}$$

Through calculation of Matlab, the state-space equation of the time-delay system is shown below:

$$\xi(K+1) = \underbrace{\begin{bmatrix} 1.4421 & 0.1143 & 0 & -0.0045 & 0.0649 \\ 9.4370 & 1.4421 & 0 & -0.0908 & 1.0958 \\ 0.0237 & 0.0008 & 1.0000 & 0.0833 & 0.2419 \\ 0.4814 & 0.0237 & 0 & 0.6845 & 3.6657 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A_{delay}} \xi(K)_+ \underbrace{\begin{bmatrix} 0.0028 \\ 0.2757 \\ 0.0111 \\ 1.1002 \\ 1.0000 \end{bmatrix}}_{B_{delay}} u \tag{9}$$

Since we have the A_d and A_{delay} matrix, through using "eig" then can we get the eigenvalues of the discrete system and delay system. The eigenvalues of the discrete system is :

$$\begin{bmatrix} 1.0000 & 2.4781 & 0.4008 & 0.6899 \end{bmatrix} \tag{10}$$

The eigenvalues of the delay system is :

$$\begin{bmatrix} 1.0000 & 2.4781 & 0.4008 & 0.6899 & 0 \end{bmatrix} \tag{11}$$

From the above results, we can know that except for the additional eigenvalue 0 in the delay system, the rest of the eigenvalue are the same.

Question 2

In questions shown below, we have to use new state-space equations, where:

$$A = \begin{bmatrix} 1.0041 & 0.0100 & 0 & 0 \\ 0.8281 & 1.0041 & 0 & -0.0093 \\ 0.0002 & 0.0000 & 1 & 0.0098 \\ 0.0491 & 0.0002 & 0 & 0.9629 \end{bmatrix}, B = \begin{bmatrix} 0.0007 \\ 0.1398 \\ 0.0028 \\ 0.5605 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{12}$$

$$Q = I_4, P_f = 10 * I_4, R = 1 \tag{13}$$

(a) In the DP programming,$X(K+1) = (A+BK)X(K)$. Additionally, the K is calculated by the **Control policy** and **Riccati equation** of DP:

$$K(K) = -(R + B^T P(K+1)B)^{-1} B^T P(K+1)A \tag{14}$$

Where the **Riccati equation** is:

$$P(K-1) = Q + A^T P(K)A - A^T P(K)B(R + B^T P(K)B)^{-1} B^T P(K)A \tag{15}$$

$$P(N) = P_f \tag{16}$$

Through calculating **Riccati equation** in the matlab and the check the eigenvalues of **A+BK**, we found that the shortest **N** to make the system asymptotically stable is **N=33**, and the **K** is shown below:

$$K = \begin{bmatrix} -46.1566 & -5.2278 & 0.0155 & 0.4031 \end{bmatrix} \tag{17}$$

(b) Through using the idare command, the P_∞ is calculated:

$$P_\infty = 10^4 \begin{bmatrix} 4.8730 & 0.5323 & -0.1038 & -0.1154 \\ 0.5323 & 0.0587 & -0.0114 & -0.0127 \\ -0.1038 & -0.0114 & 0.0125 & 0.0027 \\ -0.1154 & -0.0127 & 0.0027 & 0.0030 \end{bmatrix} \tag{18}$$

Using the DP method to calculate the P_∞ with $\text{norm}(P(K+1)P(K)) \leq 10^{-1}$, the the necessary iteration number **i=426**. Additionally, the P calculated by the DP method is shown below:

$$P = 10^4 \begin{bmatrix} 4.8726 & 0.5322 & -0.1037 & -0.1154 \\ 0.5322 & 0.0587 & -0.0114 & -0.0127 \\ -0.1037 & -0.0114 & 0.0125 & 0.0027 \\ -0.1154 & -0.0127 & 0.0027 & 0.0030 \end{bmatrix} \tag{19}$$

(c) If we take the P_f calculated above into the equation, then the new gain K is shown below:

$$K = \begin{bmatrix} -67.7421 & -7.5226 & 0.6935 & 0.9024 \end{bmatrix} \tag{20}$$

Additionally, the times of iteration is **N=1**, because under the P_f we calculated above, the function could converge and the system become stable through only one step(each element of the eigenvalue inside the unit disk). However, since the P_f we used is larger than the initial $P_f(10 * I_4)$, the result may be not so ideal. If we want more optimized result, we should make the N longer.

Question 3

(a) Batch solution of LQ problem is based on the equation shown below:

$$\mathbf{x} = \Omega x(0) + \Gamma \mathbf{u} \tag{21}$$

Where

$$\underbrace{\begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\Omega} x(0) + \underbrace{\begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}}_{\Gamma} \underbrace{\begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix}}_{\mathbf{u}} \tag{22}$$

$$u^*(k) = u(0:N-1|K) = \underbrace{-(\Gamma^T \bar{Q} + \bar{R})^{-1} \Gamma^T \bar{Q} \Omega}_{K_b} x(k) \tag{23}$$

Additionally, the iteration time of Batch is **N=33** the K_b calculated by the Batch solution is shown below, which is same as what we calculated in the DP solution:

$$K_b = \begin{bmatrix} -46.1566 & -5.2278 & 0.0155 & 0.4031 \end{bmatrix} \tag{24}$$

Question 4

(a) For the question 4, we choose the Batch method to calculate the Receding Horizon Control, and the RHC follows the equation shown below:

$$K_b = -(\Gamma^T \bar{Q} + \bar{R})^{-1} \Gamma^T \bar{Q} \Omega \tag{25}$$

$$X(K+1) = AX(K) + Bu(K) = (A + B * K_b) X(K) \tag{26}$$

$$u(K) = K_b X(K) \tag{27}$$

Taking the initial condition into consideration-$X(0) = [\pi/38\ 0\ 0\ 0]^T$ and $u(0) = 0$ (assume no initial input), we take the time steps of x_1 and x_2 for 100, x_3 for 200, and input u for 30. The plot of RHC is shown in figure 1. According to the knowledge of the RHC, both N and R will affect the result of RHC, where R matrix affects the input and N decides predicted steps. If the R is small, cost function allows to give more input to the system to make it response. By increasing the number of prediction steps, we can obtain a more optimized and stable model. As is shown in the figure 1-4, comparing with R=0.1 and

R=1, State decay of the system with R=0.1 will be faster and the input with R=0.1 is also larger. Compared with the N=40, N=80 takes more optimized and global results. If the N tend to ∞, then result will be equal to LQR.

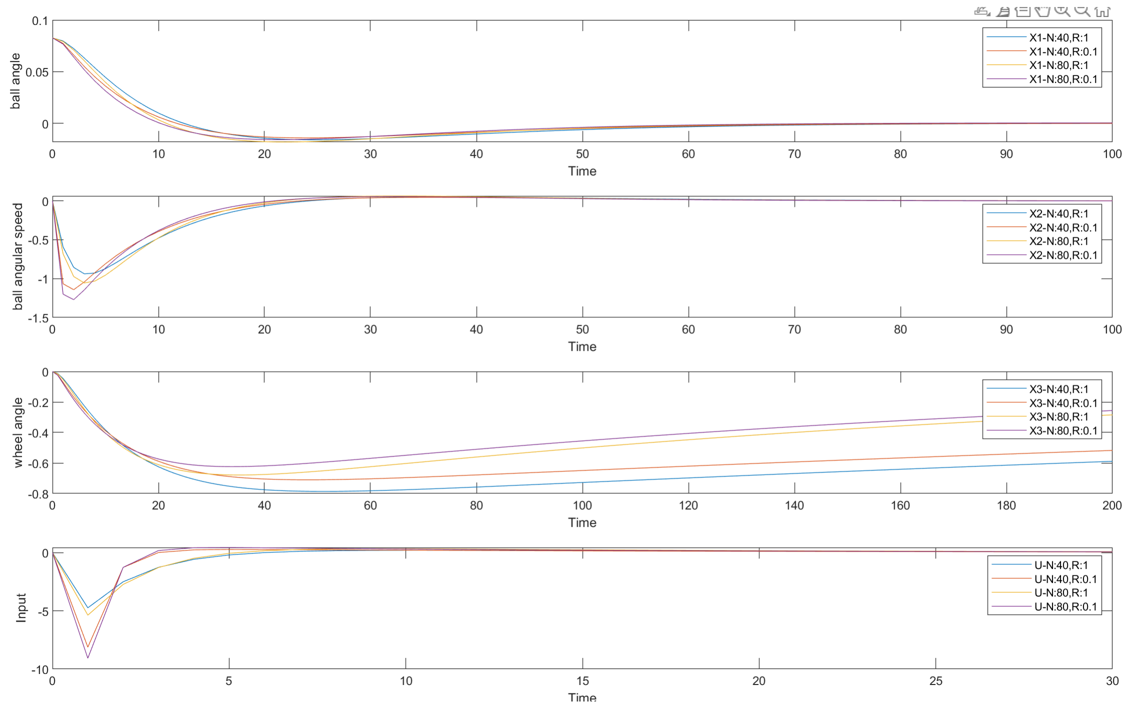

Figure 1: Plot of RHC

Question 5

(a) Rather than RHC, CRHC question considers more about the constraint which is the Significant difference compared to LQR. In the question, we have to consider that $|x_2(K)| \leq 1$ and $|u(K)| \leq 8$. The system follow the equation shown below:

$$\begin{aligned} \underset{\mathbf{u},\mathbf{x}}{\text{minimize}} \quad & V_N(x, \mathbf{u}, \mathbf{x}) = x^\top Q x + \mathbf{x}^\top \bar{Q} \mathbf{x} + \mathbf{u}^\top \bar{R} \mathbf{u} \\ \text{subject to} \quad & x(k+1) = Ax(k) + Bu(k) \\ & F\mathbf{u} + G\mathbf{x} \leq h \end{aligned} \tag{28}$$

Through using the Matlab command *quadprog* to calculate the CRHC question with the time steps of x_1 and x_2 for 100, x_3 for 200, and input u for 30. The figure 2 shown below is the result of CRHC. Compared with the plot of RHC, the curve of CRHC is a bit same as the RHC. However, since we add the constraints $|x_2(K)| \leq 1$ and $|u(K)| \leq 8$ in the system. Then, we can see in the sub-figure 2 and sub-figure 4-the $x2$ can not break through the lower boundary -1 as before.Instead, it changes after a period of -1. Additionally, the input u of CRHC also could not break the constraint **-8**, rather than what it reached in the RHC (U-N 40R 0.1:-8.12,U-N 80R 0.1:-9.07). This is also what we should do in the reality, our X and u can't change without limit.

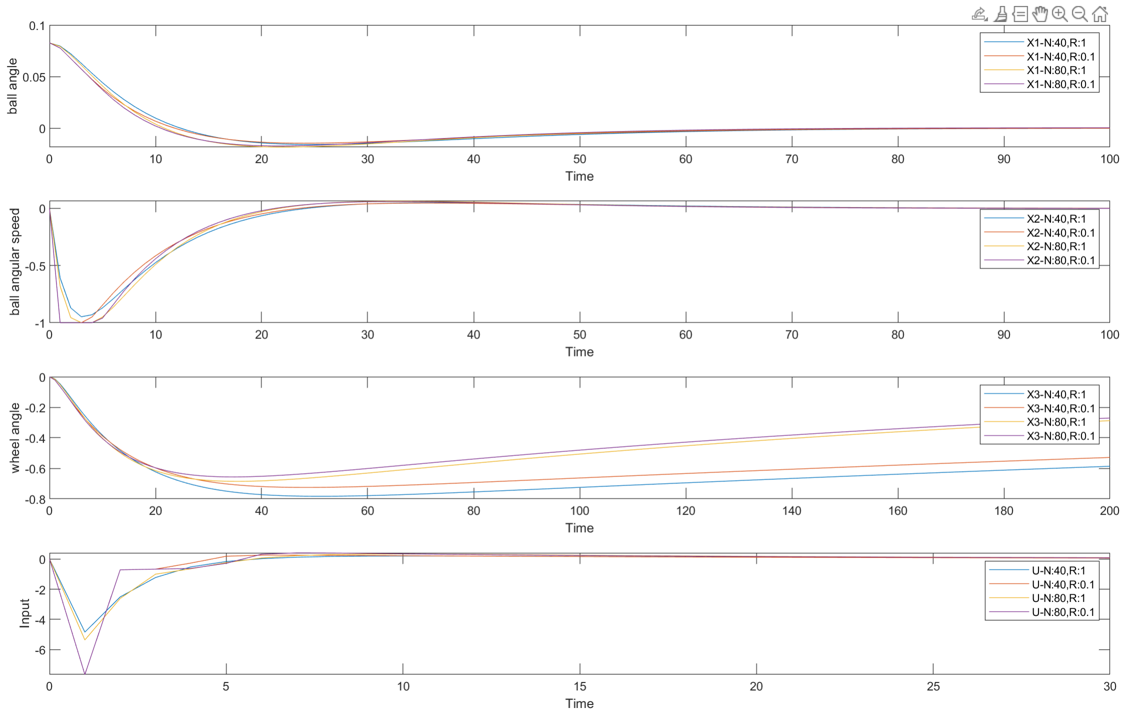

Figure 2: Plot of CRHC