SSY281 Model Predictive Control

Assignment 2

Linear Quadratic and Receding Horizon Control

$\boxed{\text{Due February 5 at 23:59}}$

Systems & Control

Department of Electrical Engineering

Chalmers University of Technology

January 2020

Instructions

This assignment is **individual** and must be solved according to the following rules and instructions:

- Written report:
  - For those questions where a text (motivations, explanations, observations from simulations) rather than a numerical value is asked, the answers should be concisely written in the report.
  - Figures included in the report should have legends, and axes should be labelled.
  - The report should be uploaded *before the deadline* in Canvas.
  - Name the report as A2_X.pdf, where X is your *group* number (see Canvas page).
- Code:
  - Your code should be written in the Matlab template provided with this assignment following the instructions therein.
  - Name the Matlab scripts as A2_X.m, DP_X.m, BS_X.m, where X is your *group* number.
- Grading:
  - This assignment if worth **15 points** in total. Each question is worth 1 point.

1 Dynamic Programming solution of the LQ problem

Consider the following system

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

and the following quadratic cost function

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

with Q, R, P_f positive definite matrices. A finite-time LQ controller can be found as by solving the following Problem (3)

$$\begin{aligned} \min_{u(0:N-1)} \quad & V_N(x(0), u(0:N-1)) \\ \text{s.t.} \quad & x(k+1) = Ax(k) + Bu(k), \end{aligned} \tag{3}$$

where

$$u(0:N-1) = \{u(0), u(1), \ldots, u(N-1)\}.$$

The following values in (1) and (2) have to be used in the rest of the assignment.

$$A = \begin{bmatrix} 1.0025 & 0.1001 \\ 0.0500 & 1.0025 \end{bmatrix}, \; B = \begin{bmatrix} 0.0050 \\ 0.1001 \end{bmatrix}, \; C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \tag{4}$$

$$Q = \begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}, \; P_f = \begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}, \; R = 0.5. \tag{5}$$

Hint. You can use `eig` function in MATLAB to check the stability of the system and `dare` function to solve the discrete-time algebraic Riccati equation.

Question 1 (**Points: 2**). *Find a finite-time LQ controller by solving the Problem* (3) *in a Dynamic Programming fashion. Write your solution in the function* `DP.m`*.* **Note.** *Do not change the inputs and outputs of the function in the template.*

Question 2 (**Points: 2**). *With the numerical values in* (4) *and* (5)*, use the function* `DP.m` *to find the shortest N that makes the system* (1)*, with $u(k) = K_0 x(k)$, asymptotically stable, where $[K_0, \ldots, K_{N-1}]$ is the solution of the problem.* **Note.** *If you return $-K_0$ instead of K_0, your answer would be considered incorrect!*

Question 3 (Points: 2). *Find the stationary solution P_f of the Riccati equation in the `DP.m` function using the values in* (4) *and* (5)*. Consider this new P_f as the terminal cost gain, defined in* (2)*, and find the shortest N that makes the system* (1) *asymptotically stable with $u(k) = K_0 x(k)$; explain in the report the difference with the N found the previous question.*

2 Batch solution of the LQ problem

For the system (1), the state trajectory over a horizon of N steps can be computed based on the initial condition and the input trajectory as follows

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

where

$$\mathbf{x} = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix}, \ \mathbf{u} = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix}. \tag{6}$$

Hint. The Matlab commands `blkdiag` and `kron` can be used to build the matrices Ω, Γ.

Question 4 (Points: 1). *Solve Problem* (3) *with the batch approach and write your solution in the template function `BS.m`.*

Question 5 (Points: 1). *With the numerical values in* (4) *and* (5)*, use the function `BS.m` to find the shortest N that makes the system* (1)*, with $u(k) = K_0 x(k)$, asymptotically stable, where*

$$\begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix} = \begin{bmatrix} K_0 \\ K_1 \\ \vdots \\ K_N - 1 \end{bmatrix} x(0).$$

3 Receding horizon control

Consider the cost function defined in (2) with Q, P_f as in (5) and the following sets of tuning parameters:

1. $R = 0.5$ and $N = 5$,
2. $R = 0.5$ and $N = 15$,

3. $R = 0.05$ and $N = 5$,

4. $R = 0.05$ and $N = 15$.

Question 6 (**Points: 2**). *Design four RHC controllers with the given set of parameters. Simulate the obtained closed-loop systems for 20s starting from the initial condition* $x(0)^\top = \begin{bmatrix} 1 & 0 \end{bmatrix}$. *Plot the system inputs and outputs for the four controllers in the same figure and explain in the report how the four different tunings affect the controller behavior.*

4 Constraint receding horizon control

Consider system (1) with cost function (2) and the following constraints:

$$\begin{aligned} F_1\mathbf{x} + G_1\mathbf{u} &= h_1, \\ F_2\mathbf{x} + G_2\mathbf{u} &\leq h_2, \end{aligned} \tag{7}$$

where $\mathbf{x}$ and $\mathbf{u}$ are defined in (6). To solve the optimization problem for the following questions, you should use `quadprog` function in MATLAB.

Question 7 (**Points: 1**). *Fill in function* `CRHC1.m`*, which calculates the optimal solution for system* (1) *when the cost function and its constraints are given by* (2) *and* (7)*, respectively. Note that the optimization variables are considered as*

$$z = \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}. \tag{8}$$

Question 8 (**Points: 2**). *Fill in function* `CRHC1.m`*, which calculates the optimal solution for system* (1) *when the cost function and its constraints are given by* (2) *and* (7)*, respectively. Use the batch solution and consider the optimization variables as*

$$z = \mathbf{u}. \tag{9}$$

Note that the system dynamics should not be included in the equality constraints anymore.

Question 9 (**Points: 2**). *Solve the problem described in Question 6 using* `CRHC1.m` *or* `CRHC2.m` *with the following constraints. Note that first, you should form up* F_2 *and* G_2 *matrices and then give them as input to* `CRHC1.m` *or* `CRHC2.m` *as inputs.*

$$|x_2(k)| \leq 0.5, \quad |u(k)| \leq 0.7 \tag{10}$$