SSY281 Model Predictive Control
Assignment 1 – Linear Quadratic and Receding Horizon Control

The purpose of this assignment is to become familiar with the formulation and solution of basic unconstrained and constrained receding horizon control algorithms based on linear quadratic control.

Instructions

The assignments comprise an important part of the examination in this course. Hence, it is important to comply with the following rules and instructions:

- The assignment is pursued and reported individually.
- The findings from each assignment are described in a short report, written by each student independently.
- The report should provide clear and concise answers to the questions, including your motivations, explanations, observations from simulations, etc. Conclusions should be supported by relevant results if applicable; e.g., the system is stable since the eigenvalues, $[0.5\ , 0.2 + 0.5j\ , 0.2 - 0.5j]$, are inside the unit circle. Figures included in the report should have legends, should be readable, should have proper scaling to illustrate the relevant information, and axes should be labeled. Try to verify your solutions if possible; e.g., plot the inputs and outputs and see whether they respect the constraints.
- Since the assignments are part of the examination in the course, plagiarism is of course not allowed. If we observe that this happens anyway, it will be reported.
- The report should be uploaded to Canvas *before the deadline*. A report uploaded a second or a day after the deadline are penalized equally. Name the report as A1.pdf.
- A MATLAB code should be uploaded which reproduces all numbers and figures in your report. Make sure that one can run your code and see your results without any error. Name the MATLAB script as A1.m.

Table 1: Points per question

Question:	1	2	3	4	5	Total
Points:	2	4	1	4	4	15

1. **Discretization of a state-space model**

A ball and wheel system controlled by a DC motor, giving an external torque u, can be described by the following differential equation:

$$\dot{x} = f(x) + g(x)u, \tag{1}$$

where

$$f(x) = \begin{bmatrix} x_2 \\ ax_4 + b\sin(x_1) \\ x_4 \\ px_4 + q\sin(x_1) \end{bmatrix}, \quad g(x) = \begin{bmatrix} 0 \\ c \\ 0 \\ r \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix},$$

and θ_1 and θ_2 are angles of the ball and the wheel respectively, as depicted in Figure 1. In this system, the electric motor can rotate the wheel, the bigger circle in the figure, and the controller should stabilize the ball, the red circle, on top of the wheel.

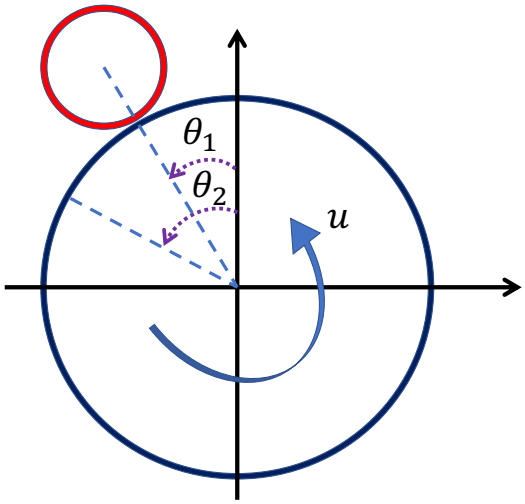

Figure 1: Ball and wheel system; click here to see how it works.

Using $\sin(x) \approx x$ for small x, an approximate linear model in the state-space form can be rewritten as

$$\begin{aligned} \dot{x}(t) &= A_c x(t) + B_c u(t), \\ y(t) &= C_c x(t). \end{aligned} \tag{2}$$

(a) [1p] Using $y(t) = \theta_1(t)$, the sampling interval $h = 0.1\,\text{s}$ and

$$a = -0.9421,\ b = 82.7231,\ c = 14.2306,\ p = -3.7808,\ q = 4.9952,\ r = 57.1120,$$

in (2), find the matrices A, B, C in the the following discrete-time model.

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

Hint: See Lemma 2.1 from the Lecture notes.

(b) [1p] In case that the continuous system (2) has an input delay of $0.8h$ seconds, calculate A_a, B_a, C_a in the following model using MATLAB.

$$\begin{aligned} \zeta(k+1) &= A_a\zeta(k) + B_a u(k) \\ y(k) &= C_a\zeta(k) \end{aligned}, \quad \zeta(k) = \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} \tag{4}$$

Find the eigenvalues of A_a and compare to those of A. What do you observe?

Hint: See Lemma 2.2 from the Lecture notes.

2. **Dynamic Programming solution of the LQ problem**

Consider the following system (discrete model of the ball and wheel system in the previous question with $h = 0.01$)

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{5}$$

and the following quadratic cost function

$$V_N(x(0), u(0:N-1)) = \sum_{k=0}^{N-1} \left(x(k)^\top Q x(k) + u(k)^\top R u(k) \right) + x(N)^\top P_f x(N), \tag{6}$$

with $Q,\ R,\ P_f$ positive definite matrices. A finite-time LQ controller can be found by solving the following Problem (7)

$$\begin{aligned} &\min_{u(0:N-1)} V_N(x(0), u(0:N-1)) \\ &\text{s.t.} \quad x(k+1) = Ax(k) + Bu(k), \end{aligned} \tag{7}$$

where

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

The following values in (5) and (6) should be used in the rest of the assignment:

$$A = \begin{bmatrix} 1.0041 & 0.0100 & 0 & 0 \\ 0.8281 & 1.0041 & 0 & -0.0093 \\ 0.0002 & 0.0000 & 1 & 0.0098 \\ 0.0491 & 0.0002 & 0 & 0.9629 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0007 \\ 0.1398 \\ 0.0028 \\ 0.5605 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{8}$$

$$Q = I_4, \quad P_f = 10 \times I_4, \quad R = 1. \tag{9}$$

(a) [2p] With the numerical values in (8) and (9), use Dynamic Programming to find the shortest N that makes the system (5), with $u(k) = Kx(k)$, asymptotically stable, where K is the solution of the problem.

Hint. You can use the `eig` function in MATLAB to check the stability of the system.

(b) [1p] Find the stationary solution P_∞ of the Riccati equation and the problem values in (8) and (9) using `dare`/`idare` MATLAB function. Now try to find the same P_∞ but using Dynamic Programming. How many iterations are needed until convergence? Do you get the same stationary Riccati matrix? As a stopping criterion use $\text{norm}(P(k+1) - P(k)) \leq 10^{-1}$.

(c) [1p] Consider the stationary matrix P_∞ from (b) as the new terminal cost gain P_f, see (6). Provide the new control gain and find the shortest N that makes the system (5) asymptotically stable with $u(k) = Kx(k)$; explain in the report the difference with the N found in question (a).

3. **Batch solution of the LQ problem**

For the system (5), the state trajectory over a horizon of N steps can be computed based on the initial condition and the input trajectory as follows

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

where

$$\boldsymbol{x} = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix}. \tag{10}$$

(a) [1p] With the numerical values in (8) and (9), use the batched solution to find the shortest N that makes the system (5), with $u(k) = K_0 x(k)$, asymptotically stable, where

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

4. [4p] **Receding horizon control**

Consider the cost function defined in (6) with Q, P_f as in (9) and the following sets of tuning parameters:

1. $R = 1$ and $N = 40$,
2. $R = 1$ and $N = 80$,
3. $R = 0.1$ and $N = 40$,
4. $R = 0.1$ and $N = 80$.

Design four RHC controllers with the given set of parameters. Simulate the obtained closed-loop systems for 200 time steps starting from the initial condition $x(0) = \begin{bmatrix} \frac{\pi}{38} & 0 & 0 & 0 \end{bmatrix}^\top$, i.e., the ball is five degrees away from its equilibrium point. Plot the system inputs and states (ball angle, ball angular speed, and wheel angle) for the four controllers in the same figure (four subplots, one per each variable) and explain in the report how the four different tunings affect the controller behavior. Plot (x_1 and x_2 for 100 time steps, x_3 for 200 time steps, and u for 30 time steps to have more informative plots)

Note. Use either Dynamic Programming or batched solution.

5. **Constrained receding horizon control**

To solve the optimization problem for the following question, you should use `quadprog` function in MATLAB.

(a) [4p] Solve the problem described in Question 4 with the following constraints:

$$|x_2(k)| \leq 1, \quad |u(k)| \leq 8. \tag{11}$$

Comment on your observations and compare with the unconstrained case.