## **Exercise 3: Linear Model Predictive Control**

Dr. Lilli Frison, Jochem De Schutter, Prof. Dr. Moritz Diehl

## 1. Discrete linear system models: Consider an inverted pendulum with nonlinear dynamics

$$\dot{x} = f(x, u) = \begin{bmatrix} x_2 \\ \sin x_1 - cx_2 + u \cos x_2 \end{bmatrix}$$
 (1)

with state vector  $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\top = \begin{bmatrix} \theta & \dot{\theta} \end{bmatrix}^\top \in \mathbb{R}^2$  and  $u \in \mathbb{R}$ . The variables  $\theta, \dot{\theta}$  represent the angle deviation and speed w.r.t. the top position, while the control variable u is a horizontal force applied at the tip of the pendulum.

(a) Linearize the system around  $x_{ss} = \begin{bmatrix} 0 & 0 \end{bmatrix}^{\top}$  and  $u_{ss} = 0$  to get the linearized system

$$\dot{x}(t) = A_{\rm c}x(t) + B_{\rm c}u(t) \tag{2}$$

What are the system matrices  $A_c$  and  $B_c$ ? Assume a damping constant c = 0.1.

$$A_{c} = \frac{\partial f}{\partial x}\Big|_{ss} = \begin{bmatrix} 0 & 1\\ \cos x_{1} & -c - u \sin x_{2} \end{bmatrix}\Big|_{ss} = \begin{bmatrix} 0 & 1\\ 1 & -c \end{bmatrix} = \begin{bmatrix} 0 & 1\\ 1 & -0.1 \end{bmatrix}$$
(3)

$$B_{\rm c} = \frac{\partial f}{\partial u}\Big|_{\rm ss} = \begin{bmatrix} 0\\\cos x_2 \end{bmatrix}\Big|_{\rm ss} = \begin{bmatrix} 0\\1 \end{bmatrix} \tag{4}$$

(b) Now discretize the state space model with a sampling time  $T_s = 0.1s$ . What are the discrete-time system matrices A and B? Use the analytic solution for continuous-time LTI systems based on the matrix exponential

$$e^X = \sum_{k=0}^{\infty} \frac{1}{k!} X^k$$
 (5)

(7)

To evaluate the matrix exponential, cut off the sum at k = 1. Terms proportional to  $T_s^k$ ,  $k \ge 2$ , can be neglected.

$$A = e^{A_{\rm c}T_{\rm s}} = \sum_{k=0}^{1} \frac{1}{k!} (A_{\rm c}T_{\rm s})^k = I + A_{\rm c}T_{\rm s} = \begin{bmatrix} 1 & T_{\rm s} \\ T_{\rm s} & 1 - 0.1T_{\rm s} \end{bmatrix} = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 0.99 \end{bmatrix}$$
(6)

$$B = A \int_0^{T_{\rm s}} e^{-A_{\rm c}\theta} \mathrm{d}\theta B_{\rm c} = A \int_0^{T_{\rm s}} (I - A_{\rm c}\theta) \mathrm{d}\theta B_{\rm c} = A \left[\theta I - A_{\rm c}\theta^2/2\right]_0^{T_{\rm s}} B_{\rm c}$$
(8)

$$= (I + A_{\rm c}T_{\rm s})(T_{\rm s}I - A_{\rm c}T_{\rm s}^2/2)B_{\rm c} \approx T_{\rm s}B_{\rm c} = \begin{bmatrix} 0\\0.1\end{bmatrix}$$
(9)

(c) Is the resulting discrete-time system controllable?

The system is controllable if the controllability matrix

$$S_{\rm C} = \begin{bmatrix} B & AB \end{bmatrix} = \begin{bmatrix} 0 & 0.1\\ 0.1 & -0.01 \end{bmatrix}$$
(10)

is full rank. We compute  $det(S_C) = -0.01 \neq 0$ , so that we can conclude that the system is controllable.

2. Linear quadratic regulator: Let us now assume that the exact system dynamics are given by the discrete-time model

$$x_{k+1} = \begin{bmatrix} 1 & 0.1\\ 0.1 & 0.99 \end{bmatrix} x_k + \begin{bmatrix} 0\\ 0.1 \end{bmatrix} u_k$$
(11)

We want to design an infinite-horizon LQR controller to control the system, using weight matrices Q = I, R = 1.

(a) Compute the infinite horizon cost-to-go weight matrix  $P_{\infty}$  using the Ricatti recursion:

$$P_{k+1} = A^{\top} P_k A - (A^{\top} P_k B)(R + B^{\top} P_k B)^{-1} (B^{\top} P_{\infty} A) + Q, \qquad (12)$$

with initialization  $P_0 = Q$ . You can start with the Python template ex3\_lmpc\_example.py.

$$P_{\infty} = \begin{bmatrix} 12.25 & 1.80\\ 1.80 & 1.88 \end{bmatrix}$$
(13)

(b) Compute the LQR feedback matrix  $K_{\infty}$ .

$$K_{\infty} = -(B^{\top}P_{\infty}B + R)^{-1}B^{\top}P_{\infty}A = \begin{bmatrix} -0.69 & -0.71 \end{bmatrix}$$
(14)

- (c) Compute the closed-loop system matrix  $A_{\rm CL} = A + BK_{\infty}$ . Simulate and plot the closed-loop response from the initial condition  $x_0 = \begin{bmatrix} \frac{\pi}{6} & 0 \end{bmatrix}^{\top}$  for  $N_{\rm sim} = 100$  steps.
- 3. Linear model predictive control: We now introduce the input constraint  $-1 \le u \le 1$ .
  - (a) Formulate the linear MPC controller for the discretized system with horizon N = 10. Choose as terminal weight matrix the infinite-horizon cost-to-go. We do not use a terminal region. The optimal control problem reads as:

- (b) Implement the optimal control problem with help of the Opti class in CasADi.
- (c) Compute the closed-loop response for the same scenario as in Task (2c) and compare to the LQR response. The MPC response converges to the origin only slightly slower but respects the input constraints.

