Advanced Model Predictive Control
Prof. M. Zeilinger
Dr. A. Carron
Dr. L. Hewing
Dr. J. Köhler

Fall 2023

Exercise

Programming Exercise 1

Alexandre Didier, Jérôme Sieber, and Rahel Rickenbach

1 Exercise

Nominal Nonlinear MPC

0. Copy the provided files into your existing AMPC code folder (`ampyc/`). Replace old files with the new provided files (such as, e.g., `__init__.py`).
1. **(graded)** Consider the nominal nonlinear MPC problem

$$
\begin{aligned}
\min_{x,u} \quad & l_f(x_N) + \sum_{i=0}^{N-1} l(x_i, u_i) && \text{(1a)}\\
\text{s.t.} \quad & \forall i = 0, \dots, N-1, && \text{(1b)}\\
& x_{i+1} = f(x_i, u_i), && \text{(1c)}\\
& x_i \in \mathcal{X},\ \ u_i \in \mathcal{U}, && \text{(1d)}\\
& x_N \in \mathcal{X}_f,\ \ x_0 = x(k). && \text{(1e)}
\end{aligned}
$$

Implement (1) in the provided `nonlinear_mpc.py` file, using the following choices of cost function, nonlinear segway dynamics, constraints, and terminal ingredients:

$$
\begin{aligned}
l(x,u) &= x^\top Q\, x + u^\top R\, u,\\
f(x,u) &= \begin{bmatrix} x_1 + \delta t \cdot x_2 \\ x_2 + \delta t(-k x_1 - c x_2 + {}^{g}/_{l} \cdot \sin x_1 + u) \end{bmatrix},\\
\mathcal{X} &= \{x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mid A_x x \le b_x\},\\
\mathcal{U} &= \{u \mid A_u u \le b_u\},\\
l_f(x) &= 0,\\
\mathcal{X}_f &= \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}.
\end{aligned}
$$

Use the following choices for the parameters

$$
\begin{gathered}
Q = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, \qquad R = 10, \qquad N = 10,\\
k = 4, \qquad c = 1.5, \qquad l = 1.3,\\
\begin{bmatrix} -45^\circ \\ -60^\circ \end{bmatrix} \le x \le \begin{bmatrix} 45^\circ \\ 60^\circ \end{bmatrix} \qquad -5 \le u \le 5
\end{gathered}
$$

Hint: The control parameters, e.g. Q and R, are loaded by the `ControllerBase` class (super class) constructor. Therefore, you can access them with `params.Q` inside your controller. Additionally, the system object is directly passed to the constructor of the `NonlinearMPC` class. This means you can access system properties, like e.g. the state constraints, directly through the `sys` object, i.e., `sys.X`.

2. **(optional; not graded)** Consider now the same nonlinear segway system but with additive disturbances.

   a. Observe how the initial state and the disturbance affect the feasibilty of the closed-loop trajectories.

   b. Use different choices of initial states and disturbance sizes. Observe how these two parameters affect the closed-loop trajectories and the cost decrease.