Advanced Model Predictive Control — Fall 2023

Prof. M. Zeilinger
Dr. A. Carron
Dr. L. Hewing
Dr. J. Köhler

Programming Exercise 2

Linear Robust MPC

Alexandre Didier, Jérôme Sieber and Rahel Rickenbach

1 Exercise

Linear Robust MPC

Implementation of linear robust MPC in the `rmpc.py` file.

a. **(Graded)** Consider the optimization problem

$$\min_{E,Y,c_{x,j}^2,c_{u,j}^2,\bar{w}^2} \quad \frac{1}{2(1-\rho)}\left((n_x+n_u)\bar{w}^2+\sum_{j=1}^{n_x} c_{x,j}^2+\sum_{j=1}^{n_u} c_{u,j}^2\right) \tag{1a}$$

$$\text{s.t.} \quad E \succeq \mathbb{I}, \tag{1b}$$

$$\begin{bmatrix} \rho^2 E & (AE+BY)^\top \\ AE+BY & E \end{bmatrix} \succeq 0, \tag{1c}$$

$$\begin{bmatrix} c_{x,j}^2 & [A_x]_j E \\ E^\top [A_x]_j^\top & E \end{bmatrix} \succeq 0,\ j \in [1, n_x], \tag{1d}$$

$$\begin{bmatrix} c_{u,j}^2 & [A_u]_j Y \\ Y^\top [A_u]_j^\top & E \end{bmatrix} \succeq 0,\ j \in [1, n_u], \tag{1e}$$

$$\begin{bmatrix} \bar{w}^2 & v_w^\top \\ v_w & E \end{bmatrix} \succeq 0,\ \forall v_w \in \mathcal{V}(\mathcal{W}). \tag{1f}$$

Implement (1) in the `compute_tightening` method in the `rmpc.py` file and compute the sublevel δ such that $\mathcal{E} = \{e \mid \|e\|_P \leq \delta\}$ is RPI and the corresponding state and input constraint tightenings.
Note: In the provided code framework the variables $\tilde{b}_x$ and $\tilde{b}_u$ are indicated with `x_tight` and `u_tight`. Furthermore, calling your solver without further specification may result in MOSEK usage. This can be avoided by setting the argument `solver='SCS'`.

b. **(Graded)** Compute the constraint tightenings for different choices of ρ and observe how the tightenings and the RPI set $\mathcal{E}$ change. Fix ρ for the remainder of the exercise.

c. **(Graded)** Consider the robust MPC problem

$$\min_{V,z_0} \quad \sum_{i=0}^{N-1} z_i^T Q z_i + v_i^T R v_i \tag{2a}$$

$$\text{s.t.} \quad \forall i = 0,\dots,N-1, \tag{2b}$$

$$z_{i+1} = Az_i + Bv_i, \tag{2c}$$

$$[A_x]_j z_i \leq [b_x]_j - \tilde{b}_{x,j}, \quad j \in [1, n_x], \tag{2d}$$

$$[A_u]_j v_i \leq [b_u]_j - \tilde{b}_{u,j}, \quad j \in [1, n_u], \tag{2e}$$

$$z_N = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{2f}$$

$$\|x(k) - z_0\|_P \leq \delta, \tag{2g}$$

Implement (2) in the provided `PE2.ipynb` file.

Note: The system and parameter objects are directly passed to the constructor of the `RMPC` class. This means you can access system properties and parameter values, like e.g. the state constraints or the control parameter Q, directly through the `sys` and `params` object respectively, i.e., `sys.X` and `params.Q`.