Model Predictive Control - EL2700

Assignment 4 : Model predictive control

2019

Automatic control
School of Electrical Engineering and Computer Science
Kungliga Tekniska Högskolan

Assignment 4: Model predictive controller

In this assignment, we will implement a linear MPC to achieve swing-up and balancing control of an inverted pendulum on a cart. In Part 1, we start by considering an inverted pendulum with simplified cart dynamics. We will compute backwards reachable sets to visualize the feasible initial states of the problem, and design an MPC to stabilize the inverted pendulum through cart acceleration. In Part 2, we will include the dynamics of the cart and implement an error free reference tracking MPC for moving the cart from one position to the other while maintaining the pendulum upright.

Predefined source code As before, we have predefined most of the required parameters and inline MATLAB functions required to solve the problems. These are given between '=' delimiters in the given skeleton MATLAB files and should NOT be edited. Your task will only revolve around defining cost functions and formulating the model predictive control problem.
To perform geometric computations, we will use MPT3 Toolbox[1]. To download and install MPT3 Toolbox, run `install_mpt3.m` script in MATLAB[2].
To help formulate the MPC problem, Yalmip[3] is used. If you have not already done so, prepare yourself for this assignment by completing the computer exercises with Yalmip. You can find these exercises on Canvas.
We will solve optimization problems using GUROBI[4] solver. To download the software and access your license, visit `DOWNLOAD GUROBI OPTIMIZER`[5] page. Download the latest version of `GUROBI OPTIMIZER` and `README.txt` using your KTH email address. Follow the instructions on `README.txt` to get your license key. Finally, to set up GUROBI MATLAB interface, visit the documentation `Setting up the Gurobi MATLAB interface`[6]

PART I: MPC for control of inverted pendulum

MPC formulation We will now design a linear MPC to stabilize the inverted pendulum through cart acceleration. The discrete-time linearized dynamics of the inverted pendulum subjected to state and control constraints can be represented by

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t, \quad y_t = Cx_t \\ u_t &\in \mathcal{U}, \quad x_t \in \mathcal{X} \qquad \forall t \geq 0 \end{aligned}$$

with $x_t = [\theta \quad \dot{\theta}]^T \in \mathbb{R}^2$, $u_t \in \mathbb{R}$, and $y_t = \theta \in \mathbb{R}$. The sets $\mathcal{U}$, $\mathcal{X}$ are polyhedrons described by the following linear inequalities

$$\mathcal{U} = \{u : H_u u \leq h_u\}, \quad \mathcal{X} = \{x : H_x x \leq h_x\}$$

In this design task, we will consider the following state and control constraints

$$x_{min} \leq x_t \leq x_{max}, \quad u_{min} \leq u_t \leq u_{max} \tag{1}$$

with $x_{min} = [-\pi/4 \ \ -\pi/2]^T$, $x_{max} = [\pi/4 \ \ \pi/2]^T$, $u_{min} = -5$, and $u_{max} = 5$. We will now perform finite-horizon optimization in a receding-horizon fashion and solve the following optimization problem

$$\begin{array}{lll} \text{minimize} & \sum_{k=0}^{N-1} x_{t+k|t}^T Q x_{t+k|t} + u_{t+k|t}^T R u_{t+k|t} + q_f(x_N) & \\ \text{subject to} & x_{t+1+k|t} = Ax_{t+k|t} + Bu_{t+k|t} & k = 0, 1, \ldots, N-1 \\ & x_{t+k|t} \in \mathcal{X}, \quad u_{t+k|t} \in \mathcal{U} & k = 0, 1, \ldots, N-1 \\ & x_{t+N|t} \in \mathcal{X}_f & \\ & x_{t|t} = x_t & \end{array} \tag{2}$$

[1] https://www.mpt3.org/Main/HomePage
[2] https://www.mpt3.org/Main/Installation
[3] https://yalmip.github.io/
[4] https://www.gurobi.com/academia/academic-program-and-licenses/
[5] https://www.gurobi.com/downloads/
[6] https://www.gurobi.com/documentation/8.1/refman/matlab_setting_up_the_guro.html

where $q_f(x_N) = x_N^T Q_f x_N$ with $Q_f \succeq 0$, and $\mathcal{X}_f$ are the terminal cost and terminal constraint set respectively. If the terminal constraint set $\mathcal{X}_f$ is control invariant, then the MPC problem (**??**) is recursively feasible. In this design task, we will consider the following two control invariant sets as the terminal constraint set:

- the zero terminal constraint set, $\mathcal{X}_f^0 = \{0\}$
- the invariant set for the closed-loop dynamics under the infinite-horizon LQR control $\mathcal{X}_f^{\mathrm{LQR}}$.

For the MPC problem (2) to be feasible, the controller must be able to steer the system state x_t to the terminal set $\mathcal{X}_f$ in N, while satisfying the state and control constraints. To this end, we will first compute the N-step controllable set $\mathcal{K}_N(\mathcal{S})$. We provide you the MATLAB script `compute_N_step_controllable_set.m` to compute $\mathcal{K}_N(\mathcal{S})$. Your tasks are as follows:

1. **Influence of terminal sets**

   Comment on how the size of the terminal set influences the N-step controllable set.

2. For terminal sets $\mathcal{X}_f = \{0\}$ and $\mathcal{X}_f = \mathcal{X}_f^{\mathrm{LQR}}$ respectively:

   (a) **Influence of control horizon**
   Compute the control invariant set $\mathcal{K}_N(\mathcal{X}_f)$ for varying control horizons $N = 5, 10$, and 20. Keep the state and control constraints as given in (1). Comment on the influence of N on $\mathcal{K}_N(\mathcal{X}_f)$.

   (b) **Influence of control constraints** Compute the control invariant set $\mathcal{K}_N(\mathcal{X}_f)$ for varying control constraints. Keep the state constraints as given in (1) and use $N = 5$. Comment on the influence of the control constraint on $\mathcal{K}_N(\mathcal{X}_f)$.

Simulate the inverted pendulum You will now simulate an inverted pendulum with MPC control by running the `run_simulation.m` script. The initial state for the simulation is set to $x_0 = [0.2 \quad 0]^T$.

- Formulate the optimization problem in Yalmip, by filling out `stabilizing_mpc_controller.m`. You will need to add the following to the script
  - The discrete time A and B system matrices
  - The input and state constraint sets on H-polyhedron form

    $$H_x x \leq h_x, \qquad H_u u \leq h_u$$

  - The objective and constraint formulation in Yalmip
- Simulate your controller with $\mathcal{X}_f = \{0\}$ and $\mathcal{X}_f = \mathcal{X}_f^{\mathrm{LQR}}$ with $N = 20$ and 3 respectively.
  - Verify that the initial state remains inside the $\mathcal{K}_N(\mathcal{X}_f)$ set from the last task.
  - What issue arises when you set $N = 3$?

PART II: MPC for reference tracking

We will now design a linear MPC for an inverted pendulum including the dynamics of the cart. The objective is to move the cart from one point to the other while keeping the pendulum upright. The controller to be designed should meet the following three performance criteria:

1. The magnitude of the control input should be limited to $|u_t| \leq 5$
2. The pendulum should be kept within $\pm 10°$ around the vertical position while the cart should avoid overshoot.

To this end, we will implement reference tracking MPC. We will reuse the discrete-time linear model of the cart pendulum system derived in the first design project

$$\begin{aligned} \boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + Bu_t + B_w w_t, \quad y_t = C\boldsymbol{x}_t \\ u_t &\in \mathcal{U}, \quad \boldsymbol{x}_t \in \mathcal{X} \;\; \forall t \geq 0 \end{aligned} \tag{3}$$

where $\boldsymbol{x}_t \in \mathbb{R}^4$, $u_t \in \mathbb{R}$, $w_t \in \mathbb{R}$, and $y_t \in \mathbb{R}$ are the system state, control input, disturbance input, and output respectively. Error-free reference tracking in stationarity requires the existence of a constant control signal $\bar{u} \in \mathcal{U}$ and an associated state $\bar{\boldsymbol{x}} \in \mathcal{X}$ such that

$$\begin{bmatrix} A-I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{x}} \\ \bar{u} \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} \tag{4}$$

Note that we can find $(\bar{\boldsymbol{x}}, \bar{u})$ for every right hand side of (**??**) if and only if $\begin{bmatrix} A-I & B \\ C & 0 \end{bmatrix}$ has full rank. We can now design an MPC for tracking the system states and control input to $(\bar{\boldsymbol{x}}, \bar{u})$, by formulating the optimal control problem in terms of the difference variables ($\Delta\boldsymbol{x}_t = \boldsymbol{x}_t - \bar{\boldsymbol{x}}, \Delta u_t = u_t - \bar{u}$) as follows:

$$\begin{array}{lll} \text{minimize} & \sum_{k=0}^{N-1} \Delta\boldsymbol{x}_{t+k|t}^T Q \Delta\boldsymbol{x}_{t+k|t} + \Delta u_{t+k|t}^T R \Delta u_{t+k|t} + & \\ & \quad + \Delta\boldsymbol{x}_{t+N|t}^T Q_f \Delta\boldsymbol{x}_{t+N|t} + & \\ & \quad + \Delta\boldsymbol{s}_{t+k|t}^T Q_{slack} \Delta\boldsymbol{s}_{t+k|t} & \\ \text{subject to} & \Delta\boldsymbol{x}_{t+k+1|t} = A\Delta\boldsymbol{x}_{t+k|t} + B\Delta u_{t+k|t} & k = 0, 1, \ldots, N-1 \\ & H_x \left(\Delta\boldsymbol{x}_{t+k|t} + \bar{\boldsymbol{x}}\right) \leq h_x + \boldsymbol{s}_{t+k|t} & k = 0, 1, \ldots, N-1 \\ & H_u \left(\Delta u_{t+k|t} + \bar{u}\right) \leq h_u & k = 0, 1, \ldots, N-1 \\ & \Delta\boldsymbol{x}_{t+N|t} + \bar{\boldsymbol{x}} \in \mathcal{X}_f & \\ & \Delta\boldsymbol{x}_{t|t} + \bar{\boldsymbol{x}} = \boldsymbol{x}_{t|t} & \\ & \begin{bmatrix} A-I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{x}} \\ \bar{u} \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} & \end{array} \tag{5}$$

in variables $\left\{\Delta\boldsymbol{x}_{t+k|t}, \Delta u_{t+k|t}\right\}$, $\bar{\boldsymbol{x}}$, and $\bar{u}$, and then apply the control $u_t = \Delta u_{t|t}^\star + \bar{u}$. Slack variables $\boldsymbol{s}_{t+k|t}$ have been added to soften the state constraints. The matrix Q_{slack} contains very large values, which will force the controller to avoid going over the constraint values.

As discussed in the lecture notes, instead of a terminal set $\mathcal{X}_f$ we can instead check the constraints explicitly over a fixed horizon N_c. Here, this is done with the infinite horizon optimal control $\Delta u_t = -L\Delta x_t$ as follows

$$\begin{aligned} H_x(A-BL)^k \Delta\boldsymbol{x}_{t+N|t} + H_x\bar{\boldsymbol{x}} &\leq h_x \qquad k = 1, \ldots, N_c \\ -H_u L(A-BL)^{k-1} \Delta\boldsymbol{x}_{t+N|t} + H_u\bar{u} &\leq h_u \qquad k = 1, \ldots, N_c \end{aligned} \tag{6}$$

Your tasks are as follows:

- Verify that the system is reachable, and that we can find $(\bar{\boldsymbol{x}}, \bar{u})$ for every reference signal r.
- Implement the controller in (5). You will need to add the following to the script:
  - The discrete time A and B system matrices
  - The input and state constraint sets on H-polyhedron form

    $$H_x x \leq h_x, \qquad H_u u \leq h_u$$

  - The Yalmip formulation has already been filled out and should not be changed
- How does this controller track the constant reference? Are the state and control constraints satisfied?
- Add both positive and negative small constant disturbances (e.g. $w_t = \pm 0.05$) to mimic the presence of the horizontal wind force on the inverted pendulum. Comment on the tracking ability of the controller in this scenario.

PART III: Submission

To complete this design project, you should upload filled in MATLAB skeleton files:

- `compute_N_step_controllable_set.m` and `stabilizing_mpc_controller.m` for **PART I**
- `tracking_mpc_controller.m` for **PART II**

We have provided space for you to answer the questions in these MATLAB skeleton files. Use that space to write your reflections. Please do not upload any additional report.

Good Luck!