Learning-Based Predictive Control EECI 2023
Dr. L. Hewing
Prof. M. Zeilinger
Prof. L. Fagiano

Exercise 4
Invariance-Based & Predictive Safety Filter

1 Programming Exercise

Invariance-Based & Predictive Safety Filter

Preliminaries

You are provided with the following coding template files:

`E04_SafetyFilter.mlx`	Main (live) script	(to be edited throughout)
`IBSF.m`	Linear Invariance-based Filter Class	(to be edited in Section 2)
`MPSF.m`	Predictive Safety Filter Class	(to be edited in Section 3)
`Explorer.m`	Exploration Controller Class	
`LinearSystem.m`	Helper Class	
`SimulateSF.m`	Helper Function	

Areas in the code to be modified are indicated by

```
% ----- Start Modifying Code Here ------
%
% The existing (incomplete) code in the modifiable areas serves as a hint
%
% ----- End Modifying Code Here ------
```

Problem Description

Consider the system defined as follows

$$x(k+1) = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} T_s^2/2 & 0 \\ T_s & 0 \\ 0 & T_s^2/2 \\ 0 & T_s \end{bmatrix} u(k)$$

where the states $x(k) = \begin{bmatrix} p_x & v_x & p_y & v_y \end{bmatrix}^\top$ are the 2D position and velocity of a robotic agent.

The robot is operating inside of a polytopic room and its maximal velocity is limited such that the constraints can be given as

$$x(k) \in \mathcal{X} = \{x \,|\, H_x x \leq h_x\}.$$

Additionally, the robot is subject to polytopic input constraints

$$u(k) \in \mathcal{U} = \{u \,|\, H_u u \leq h_u\}.$$

The system uses a simple exploration strategy which consists of applying random inputs as long as the system is within the constraint, and a stabilizing linear controller if the system state lies outside the constraints.

The goal is to safely explore the room without hitting the walls or violating the velocity constraints.

1. Introduction to semidefinite programming in Yalmip: Find the maximum ellipse $\mathcal{E} = \{x \,|x^T E x \leq 1\}$ in $\mathbb{R}^2$ inscribed in a rectangle $\mathcal{H} = \{x \,|\, -1 \leq x_1 \leq 1, \, -2 \leq x_2 \leq 2\}$

(a) Set up the rectangle as an MPT polytope with $\mathcal{H} = \{x \mid Hx \leq h\}$.

(b) Set up the semidefinite program maximizing the volume of $\mathcal{E}$ subject to the rectangular constraints in Yalmip using the following semidefinite program

$$\begin{aligned} &\min_{E^{-1}} - \log\det E^{-1} \\ &\text{s.t. } [H]_i E^{-1} [H]_i^T \leq h_i^2 \ \forall i = 1, \dots, 4 \end{aligned}$$

Hint: Yalmip provides a dedicated concave logdet function

(c) Solve the optimization problem and visualize the solution

2. To ensure safety of the system, we first investigate and invariance-based safety filter based on ellipsoidal safe sets. Using Yalmip we find the maximum volume invariant ellipse $\mathcal{S} = \{x \mid x^T P x \leq 1\}$ under a linear controller $\pi_s(x) = Kx$ within the state & input constraints: $\mathcal{S} \subseteq \mathcal{X}$, $K\mathcal{S} \subseteq \mathcal{U}$

   (a) Open `IBSF.m` and complete the definition of the optimization problem in the class constructor, optimizing over both P and K

   Hint: Use the lecture notes as your reference.

   (b) Complete the implementation of the `filter(x,u_L)` function of the IBSF class.

   (c) Evaluate the resulting safety filter.

3. Next, we design and implement a model predictive safety filters (MPSF) for the system, making use of the previously designed safe set as a *terminal set*

   (a) Open `MPSF.m` and complete the optimization problem in the class constructor.

   Note: To enhance numerical feasibility, the state constraints are implemented as soft-constraints.

   (b) Simulate the system using the invariance-based safe set from the first part as your terminal set. Evaluate the behavior for different prediction horizon lengths N.