AA 598: Decision-making and Control for Safe Interactive Autonomy

Homework 2— (Recommended) Due date: Friday November 8th
Starter code: `https://github.com/UW-CTRL/AA598-aut24`

Goal. To become familiar with some planning approaches to navigate moving (stochastic) obstacles.
Starter code instructions: Pull the latest changes from `https://github.com/UW-CTRL/AA598-aut24/tree/main`. Don't forget to first commit and push your homework 1 to your own forked repo and switch back the main branch! In addition to homework 2 files, you will need a few additional packages. Activate your virtual environment and run

```
pip install equinox cvxpy git+https://github.com/UW-CTRL/cbfax.git
```

Problem set up. Let $x_{\text{rob}}^{(t)} \in \mathcal{X}_{\text{rob}} \subseteq \mathbb{R}^n$ and $u_{\text{rob}}^{(t)} \in \mathcal{U} \subseteq \mathbb{R}^m$ be the state and control of a robot at time t. Similarly, let $x_{\text{hum}}^{(t)} \in \mathcal{X}_{\text{hum}} \subseteq \mathbb{R}^n$ and $u_{\text{hum}}^{(t)} \in \mathcal{U}_{\text{hum}} \subseteq \mathbb{R}^m$ be the state and control of a human at time t. The goal is to have the robot navigate safely around the human, whilst minimizing control effort, and making progress towards tracking the horizontal axis.

$$\min_{u_{\text{rob}}^{(0:T-1)}} \quad \frac{1}{T}\sum_{t=0}^{T-1} J_t(x_{\text{rob}}^{(t)}, u_{\text{rob}}^{(t)}, x_{\text{hum}}^{(t)}, u_{\text{hum}}^{(t)}) + J_T(x_{\text{rob}}^{(T)}, x_{\text{hum}}^{(T)}) \tag{1}$$

$$\text{s.t.} \quad x_{\text{rob}}^{(t+1)} = f_{\text{rob}}(x_{\text{rob}}^{(t)}, u_{\text{rob}}^{(t)}), \qquad t = 0, \dots, T-1 \tag{2}$$

$$x_{\text{hum}}^{(t+1)} = f_{\text{hum}}(x_{\text{hum}}^{(t)}, u_{\text{hum}}^{(t)}), \qquad t = 0, \dots, T-1 \tag{3}$$

$$\|\text{pos}(x_{\text{hum}}^{(t)}) - \text{pos}(x_{\text{rob}}^{(t)})\|_2 \geq r_{\min}, \qquad t = 0, \dots, T \tag{4}$$

$$u_{\text{rob}}^{\min} \leq u_{\text{rob}}^{(t)} \leq u_{\text{rob}}^{\max}, \qquad t = 0, \dots, T-1 \tag{5}$$

$$0 \leq \text{speed}(x_{\text{rob}}^{(t)}) \leq v_{\max}, \qquad t = 0, \dots, T-1 \tag{6}$$

$$x_{\text{rob}}^{(0)} = x_{\text{rob}}^{\text{initial}}, \quad x_{\text{hum}}^{(0)} = x_{\text{hum}}^{\text{initial}} \tag{7}$$

There is uncertainty in how the human behaves, i.e., $u_{\text{hum}}^{(t)} \sim p(u_{\text{hum}} \mid x_{\text{rob}}^{(t)}, x_{\text{hum}}^{(t)}, \dots)$. Unless there is some structure about the uncertainty (e.g., Gaussian) and/or dynamics (e.g., linear), ensuring safety within some probability is challenging. Instead, we can make some approximations: (i) sample what the human(s) can do in the future and plan around that, or (ii) assume some deterministic human model (e.g., constant velocity) and plan around that.

In the starter code, we assume the human and robot agents follow a dynamically extended bicycle model with length L. Given continuous-time dynamics, we can derive discrete-time dynamics via integration (this is done in the code already).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \frac{v}{L}\tan\delta \\ a \end{bmatrix} \tag{8}$$

For simplicity, we assume $u_{\text{hum}}^{(t)} \sim \mathcal{N}(0, \Sigma)$ in addition to some saturation limits. In other words, humans will noisily continue moving along at the same initial heading and the same speed. While this is a very simple "prediction model", this can, in theory, be substituted with a more complex prediction model like some of the ones we discussed in Module #1.

1 Model Predictive Path Integral (MPPI)

In this problem, you will use MPPI for a simple two-agent system and gain some intuition into the influence of hyperparameters on the problem. The starter code provides a basic MPPI implementation.

(a) Review the code to gain some intuition into what is happening at each line, including how the cost is defined, samples are generated, and control inputs are updated. The `evaluate_trajectory_cost` is made up of various cost terms. Describe what each one of them is. How do they relate to the trajectory planning problem outlined above.

(b) Run the code a few times and observe the trajectories the robot ends up taking. Comment on the different types of behaviors you observe.

(c) You may find that the human and robot (sometimes) experience a collision (their circles overlap). What are some things you could do to prevent this, or at least minimize, this from happening?

(d) (Optional) Given your response to (c), try some of your ideas out and see if there are any notable changes. Think about how you would quantify it.

(e) (Optional) Come up with different cost functions and see what kind of behaviors you want! For example, you may want to try adding a markup term and/or inconvenience term as described in this this paper

2 Sequential Quadratic Program (SQP)

In this problem, we approach the same problem but via sequential quadratic programming. Essentially, we take the optimal control problem described above and repeatedly approximate it as a quadratic program (quadratic cost and linear constraints). The starter code provides a working implementation of the SQP algorithm (with some minor modification such as a trust region penalty and a slack on collision avoidance). To compute the collision avoidance constraint, we need access to the future states of humans when computing the constraints. Since we don't have access to the true future human states, we instead make a single prediction of where the human will be in the future (i.e., draw one sample from a prediction model), and compute the collision avoidance constraint about that sample. Like what we saw with the previous problem, we can repeatedly replan and (hopefully) be able to react to changes to the human's motion.

(a) Review the code to gain some intuition into what is happening at each line, including how the cost is defined and how the constraints are handled. What are some key differences between this approach and MPPI?

(b) Run the code a few times, try different coefficients or objective functions, and observe the trajectories the robot ends up taking. Comment on the different types of behaviors you observe.

(c) What are some limitations of this approach and what are some things you could do to address those limitations?

(d) Given the problem of a robot navigating around human agents (e.g., in autonomous driving, navigating Red Square, or in a warehouse), which method would you propose using? Why? What if you where flying a (small) drone around humans? Would your answer stay the same?

(e) (Optional) Given your response to (c), try some of your ideas out and see if there are any notable changes. Think about how you would quantify it.