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

Homework 3— (Recommended) Due date: Friday November 29th
Starter code: `https://github.com/UW-CTRL/AA598-aut24`

**Goal.** To become familiar with reachability concepts and computation of safe control sets.
**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 previous homework to your own forked repo and switch back the main branch! In addition to this homework's files, you will need a few additional packages. Activate your virtual environment and run

```
pip install --upgrade hj-reachability
pip install tqdm
```

**NOTE:** If you are working on a machine with a GPU, you can install JAX with GPU support, and it will make computing the sets much faster! See JAX installation instructions for more details.

## 1 HJ reachability

In class, we learned about HJ reachability and how it is used to compute backward reachable sets/tube. The Hamilton-Jacobi-Isaacs (HJI) PDE for a backward reachable *tube* (BRT) for an avoid set is given below.

$$\frac{\partial V(x,t)}{\partial t} + \max_{u\in\mathcal{U}} \min_{d\in\mathcal{D}} \min\left\{0, \frac{\partial V(x,t)}{\partial x}\cdot f(x,u,d)\right\} = 0$$
$$V(x,0) = F(x)$$

Intuitively, if the target set $\mathcal{T}$ is the set of state the system wishes to *avoid* and $\mathcal{T} = \{x \in \mathcal{X} \mid F(x) \le 0\}$ (i.e., negative values is *inside* the set), then we seek to find optimal controls $u$ for the system that will keep it outside $\mathcal{T}$ (i.e., *max*) subject to adversarial disturbances $d$ that aims to push the system inside $\mathcal{T}$ (i.e., *min*). A simple and intuitive way to define $F$ is to use the signed distance function—the signed distance between two circles is very easy to compute but can be more complicated for complex shapes.

We can use HJ reachability to compute BRTs describing interactions between a robot and human agent. We achieve this by considering the *relative dynamics* between a robot and human agent, and then the human's control inputs are treated as disturbances.

In class, we discussed that if we assume that both the robot and human can take *any* control/disturbance inputs from their respective feasible sets, then this can lead to overly conservative. This short paper discusses this issue and presents research ideas towards the concept of "data-driven safety" and how we should learn from data to encode reasonable assumptions about the behavior of agents. This paper presents an approach to learn state-dependent control sets from data using Control Barrier Functions and incorporate them into the HJ reachability computation to reduce the over-conservatism in the BRT.

In this problem, you will use the HJ Reachability toolbox to compute the BRT with different choices of control and disturbance bounds, and see how the size and shape of BRTs changes.

Let us model both the robot and human agent as dynamically-extended unicycles where $\mathbf{x} = [x, y, \theta, v]^T$ and $\mathbf{u} = [\omega, a]^T$.

$$\dot{\mathbf{x}}_{\text{rob}} = \begin{bmatrix} v_{\text{rob}}\cos\theta_{\text{rob}} \\ v_{\text{rob}}\sin\theta_{\text{rob}} \\ \omega_{\text{rob}} \\ a_{\text{rob}} \end{bmatrix}, \qquad \dot{\mathbf{x}}_{\text{hum}} = \begin{bmatrix} v_{\text{hum}}\cos\theta_{\text{hum}} \\ v_{\text{hum}}\sin\theta_{\text{hum}} \\ \omega_{\text{hum}} \\ a_{\text{hum}} \end{bmatrix}$$

We can also impose velocity limits on each agent.

We define the relative *position*, centered on the robot, and relative heading to be,

$$\mathbf{p}_{\text{rel}} = \begin{bmatrix} x_{\text{rel}} \\ y_{\text{rel}} \end{bmatrix} = \begin{bmatrix} \cos\theta_{\text{rob}} & \sin\theta_{\text{rob}} \\ -\sin\theta_{\text{rob}} & \cos\theta_{\text{rob}} \end{bmatrix} \begin{bmatrix} x_{\text{hum}} - x_{\text{rob}} \\ y_{\text{hum}} - y_{\text{rob}} \end{bmatrix}, \qquad \theta_{\text{rel}} = \theta_{\text{hum}} - \theta_{\text{rob}}$$

The relative state becomes $\mathbf{x}_{\text{rel}} = [x_{\text{rel}}, y_{\text{rel}}, \theta_{\text{rel}}, v_{\text{rob}}, v_{\text{hum}}]^T$.

(a) Write down the dynamics describing the relative system. That is, what is the expression for $\dot{\mathbf{x}}_{\text{rel}}$? (Hint: You will need to apply the chain rule and some algebraic manipulation.)

(b) Are the relative dynamics control and disturbance affine? If so, express the dynamics to clearly indicate the drift, control, and disturbance term.

(c) Take a look at the example notebook and starter code to get a sense of how to run a BRT computation and how to construct your own custom dynamics. Fill in the `RelativeDynamicUnicycle` class given the dynamics you derived above.

(d) If the Euclidean distance between the robot and human is less than $r_{\text{col}}$, then that is considered a collision. Run the next few cells in the starter code to compute the BRT, and analyze the resulting sets. Select three different configurations (i.e., different relative headings and velocities) and provide a brief explanation of the shape/size of the BRT. Does it align with your intuition? Include the plots.

(e) Each agent could take any controls/disturbances from their respective feasible control/disturbance sets $\mathcal{U}$ and $\mathcal{D}$. The provided starter code you ran in the previous question assumed $\mathcal{U}$ and $\mathcal{D}$ reflect the physical limits of a unicycle model. However, we can choose the set of controls/disturbance bounds to be whatever we like. For instance, $\mathcal{U}$ and $\mathcal{D}$ do not need to be the same.

(i) Compute the BRT if we assume each agent *cannot* accelerate nor rotate, i.e., $a \in [0, 0]$ and $\omega \in [0, 0]$? What is the interpretation of the results? (Hint: it is a technique we discussed in the prediction module). Select three different configurations (i.e., different relative headings and velocities) and provide a brief explanation of the shape/size of the BRT. Does it align with your intuition? Include the plots.

(ii) Compute the BRT if we assume each agent will *maximally brake* but is free to rotate. What is the interpretation of the results? Select three different configurations (i.e., different relative headings and velocities) and provide a brief explanation of the shape/size of the BRT. Does it align with your intuition? Include the plots.

(iii) Suppose you are designing a cat-catching robot. We model both the robot and cat with the unicycle dynamics but assume the cat can change its direction faster than the robot, but cannot (de)accelerate as much as the robot. (Feel free to use different dynamics to make this a bit more realistic). In this case, the robot wants to *reach* the target set, while we assume the cat wants to evade capture and therefore wants to *avoid* the target set. Set up a BRT problem to reflect these assumptions and compute the corresponding BRT. Select three different configurations (i.e., different relative headings and velocities) and provide a brief explanation of the shape/size of the BRT. Does it align with your intuition? Include the plots.

(iv) (Optional) Come up with different of assumptions about the dynamics/interaction and describe how this is reflected in the BRT set up.

(v) (Optional) We can also free to select the initial value $F$ however we like, as long as $\mathcal{T} = \{x \in \mathcal{X} \mid F(x) \leq 0\}$. With the signed distance function, we only consider the distance to the collision set using position information only. As such this choice of $F$ does not consider *collision severity*. For instance, a head on collision at high speeds is more dangerous than a side collision at low speeds. Try to come up with different choices of $F$, explain your intuition behind this choice, and describe the shape and size of the BRT you get from using it.

## 2 Safety Filter

A safety filter is a module we add on top of our planning and control stack that will adjust the desired control as necessary to ensure a safety criterion is met. One such way to construct a safety filter is to use Control Barrier Functions. Suppose we have a *valid* CBF $b : \mathbb{R}^n \to \mathbb{R}$ and extended class $\mathcal{K}_\infty$ function $\alpha : \mathbb{R} \to \mathbb{R}$ where,

$$\max_{u \in \mathcal{U}} \nabla b(x)^T f(x, u) \geq -\alpha(b(x)), \quad x \in \mathcal{X}.$$

For a given state $x \in \mathcal{X}$, the set of feasible safe controls as determined by the CBF is,

$$\mathcal{U}_{\text{safe}} = \{u \in \mathcal{U} \mid \nabla b(x)^T f(x, u) \geq -\alpha(b(x))\} \tag{1}$$

We can project the desired control into the set of safe controls defined by the CBF inequality in (1). To perform this projection, we can solve the following optimization problem (assuming each control input are independent and must be within some interval). For a given $x \in \mathcal{X}$ and a desired control input $u^{\text{des}}$, the filtered safe control $u^{\text{safe}}$ is,

$$\begin{aligned} u^{\text{safe}} = \underset{u}{\operatorname{argmin}} \quad & \|u - u^{\text{des}}\|_2^2 \\ \text{s.t.} \quad & \nabla b(x)^T f(x,u) \geq -\alpha(b(x)) \\ & u_{\min} \leq u \leq u_{\max} \end{aligned}$$

Note: we can add a slack variable to the CBF constraint to ensure the problem is always feasible, and this can be handy since it is generally difficult to find a valid CBF, similar to how it is difficult to find a (control) Lyapunov function.

(a) Show that the above optimization problem is a *quadratic program* if the dynamics are control affine.

(b) Consider the following (kinematic) unicycle model with state $\mathbf{x} = [x, y, \theta]^T$ and controls $\mathbf{u} = [v, \omega]^T$.

$$\dot{\mathbf{x}} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix}$$

Let $b(\mathbf{x}) = x^2 + y^2 - r^2$ and $\alpha(x) = ax, a > 0$. Write down the expression for $\nabla b(x)^T f(x,u) \geq -\alpha(b(x))$. Describe the safe control set $\mathcal{U}^{\text{safe}}$ as $r$ and $a$ varies. For a fixed value of $r$, what is an interpretation for $a$? How does it affect your safety filter? Is this $b$ a good choice? Why or why not?