Trust-based Rate-Tunable Control Barrier Functions in 3D Multi-Agent System

ROB 590, Fall 2022
University of Michigan, Ann Arbor
Nicole Campbell, nicoca@umich.edu

I. Introduction

In this work, the method of Trust-based Rate-Tunable Control Barrier Function (CBF) from [1] is applied in 3D for the application of quadrotors. The original method developed a trust metric for ego agents to form its own belief of how cooperative other agents in the environment are. The metric is used to adjust the rate at which the CBF allows the system trajectories to approach the boundaries of the safe region. The method was tested in simulations with a mix of uncooperative, adversarial, and cooperative agents in a 3D environment simulated in Python.

The motivation for this type of trust metric is to allow agents to adapt their safety set with respect to other agents (i.e., change how close they can get to other agents) based on the likelihood that the agent is of a certain identity (e.g., cooperative, uncooperative, adversarial). Typically the parameter that influences the safety set in the CBF is constant but it would be more efficient to be able to update this parameter based on the behaviors of the agents. For example, it is more efficient to be less conservative around cooperative agents rather than treating all agents as adversarial.

II. Problem Setup and Methodology

In our simulated environment, there are N agents with x_i and u_i representing the state and control input of agent i, respectively. Each agent has access to the states of all other agents in the environment however the identity of the other agents are unknown. Each agent has a goal state in the environment they are trying to achieve which is represented by a Control Lyapunov Function (CLF). The safety set which determines how close an agent can get to other agents is represented by a CBF. The goal is to have a subset of the agents be ego agents, who are by default cooperative, that we design the trust metric for within the CBF.

A. Control Barrier Function

The CBF ensures safety by restricting the rate of change of the barrier function, h, as shown in (1) where it is defined between two agents, i and j.

$$\dot{h_{ij}} = \frac{\delta h_{ij}}{\delta xi}\dot{x}_i + \frac{\delta h_{ij}}{\delta xj}\dot{x}_j \geq -\alpha_{ij} h_{ij} \tag{1}$$

B. Control Lyapunov Function

The CLF is defined for each agent i in (2) where x_i is the current state and x_i^g is the goal state.

$$V_i = ||x_i - x_i^g|| \tag{2}$$

C. Agent Types

1) Cooperative: In an intuitive sense, a cooperative agent actively avoids other agents in the environment to prevent collisions. This is defined as an agent j is a cooperative to agent i if for all $\dot{x}_i$, agent j chooses its control input u_j, and hence its dynamics $\dot{x}_j$, such that (1) holds for some finite $\alpha_{ij} > 0$.

2) Uncooperative: Conversely to the cooperative agents, uncooperative agents will not actively avoid other agents. An agent j is considered uncooperative to agent i if the control law for u_j is not a function of the state x_i of agent i.

3) Adversarial: Adversarial agents not only do not actively avoid other agents, but actively seek another agent as their target or goal. An agent j is considered adversarial to agent i if agent j chooses its control input u_j , and hence its dynamics $\dot{x}_j$, such that

$$\frac{\delta h_{ij}}{\delta xj}\dot{x}_j \leq 0 \tag{3}$$

D. Controller

All agents in the environment use the quadratic problem (4) with the CLF constraint (5) to solve for their control. The additional CBF constraint (6) is applied for the ego (cooperative) agents. Note that (4) has a slack variable, δ, to avoid infeasible solutions. All agents in the system follow a single integrator where $u_i = \dot{x}_i$.

$$u_i^r(x) = argmin_u(u^T u + \delta) \tag{4}$$

$$s.t.\ \dot{V} \leq -kV \tag{5}$$

$$and \ \ \frac{\delta h_{ij}}{\delta xi}\dot{x}_i + \frac{\delta h_{ij}}{\delta xi}\dot{x}_j \geq -\alpha_{ij}h_{ij} \tag{6}$$

E. Trust Metric

To ensure inter-agent safety, each ego agent i wants to avoid the point at which agent i cannot find a feasible solution to (4). This is formulated in (7). This constraint induces a half space with a hyperplane separating the safe and unsafe regions.

$$\frac{\delta h_{ij}}{\delta x_j}\dot{x}_j \geq -\alpha_{ij}\frac{\delta h_{ij}}{\delta x_i}\dot{x}_i \tag{7}$$

$$max_{u_i}\frac{\delta h_{ij}}{\delta x_i}\dot{x}_i s.t. \ \ \dot{h}_{ik} \leq -\alpha_{ik}h_{ik} \tag{8}$$

The total trust score is represented as $\dot{\alpha}_{ij}$ and is defined in (9) where there is both a distance-based and orientation-based trust score. The CBF parameter, α can then be calculated as in (10) where β is the learning rate, a hyper-parameter that needs tuning. For the simulations shown in this work, $\beta = 3$.

$$\dot{\alpha}_{ij} = \rho_d \cdot \rho_\theta \tag{9}$$

$$\alpha_{ij,t} = \alpha_{ij,t-1} + \beta * \dot{\alpha}_{ij} * dt \tag{10}$$

1) Direction Parameter: The $\bar{d}$ in (11) is a hyper-parameter representing the distance threshold for safety. This was tuned by decreasing the value to make the ego agents act more conservatively around agents while still maintaining $\dot{\alpha}_{ij} > 0$. For the simulations shown, $\bar{\delta}$ was tuned 0.08 meters.

$$\rho_d = tanh(d_j) - tanh(\bar{d}) \tag{11}$$

$$d_j = \frac{\delta h_{ij}}{\delta x_j}\dot{x}_j + \alpha h + \frac{\delta h_{ij}}{\delta x_i}\dot{x}_i \tag{12}$$

2) Orientation Parameter: The orientation parameter is a ratio of the angle between surface normal of the hyperplane and nominal agent trajectory, $\hat{n}_j$, θ_2, and the angle between surface normal of the hyperplane and actual agent trajectory, $\dot{x}_j$.

$$\rho_\theta = tanh(\frac{\theta_2}{\theta_1}) \tag{13}$$

$$\theta_1 = cos^{-1}\frac{\frac{\delta h_{ij}}{\delta x_j}\dot{x}_j}{||\frac{\delta h_{ij}}{\delta x_j}|| \cdot ||\dot{x}_j||} \tag{14}$$

$$\theta_2 = cos^{-1}\frac{\frac{\delta h_{ij}}{\delta x_j}\hat{n}_j}{||\frac{\delta h_{ij}}{\delta x_j}|| \cdot ||\hat{n}_j||} \tag{15}$$

$$\hat{n}_j = -\frac{\frac{\delta v_j^i}{\delta x_j}\hat{n}_j}{||\frac{\delta v_j^i}{\delta x_j}||} \tag{16}$$

F. Algorithm

Algorithm 1 Trust-based CBF

for $t = 0, \ldots, T$ **do**
  for all non-ego agents, k **do**
    Solve for u_k using (4)–(6)
    Update $X_{t,k}$
  end for
  for all ego agents, i **do**
    for all non-ego agents, j **do**
      Solve for u_i using (4)–(6)
      Update $X_{t,i}$
      Compute ρ_d, ρ_θ using (11), (13)
      Update α_{ij}
    end for
  end for
end for

III. Results

The results from the simulation we performed with 3 cooperative (ego) agents, 2 uncooperative agents, and 1 adversarial agent for $T = 1000$ timesteps are shown below. The trajectories can be seen in Figure 1 where the colormaps identify at which timestep the agent was. The advesarial agent is chasing the 0th cooperative (ego) agent which is the left most cooperative agent in the figure. The yellow spheres represent the starting points for the cooperative (ego) agents and the green spheres represent the goal states.

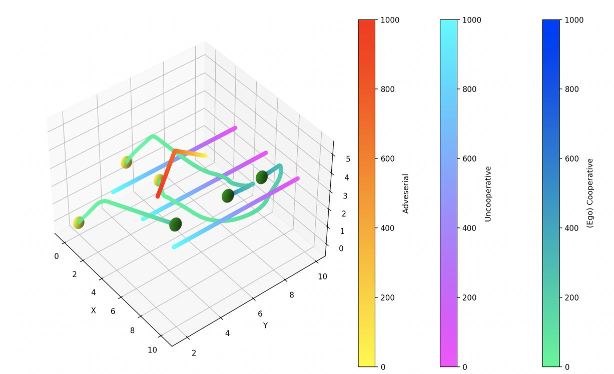

Fig. 1: Agent Trajectories

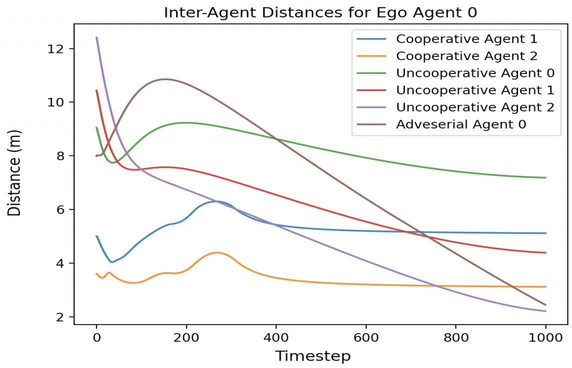

Fig. 2: Inter-Agent Distances for Ego Agent 0

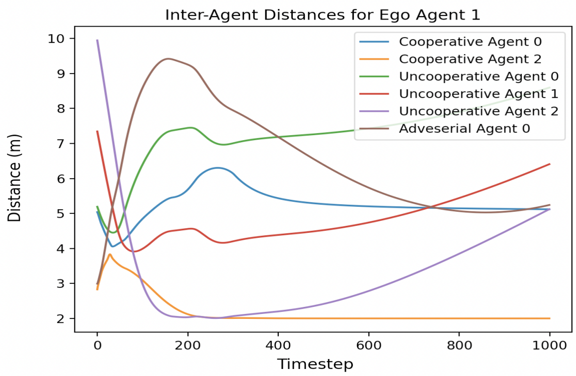

Fig. 3: Inter-Agent Distances for Ego Agent 1

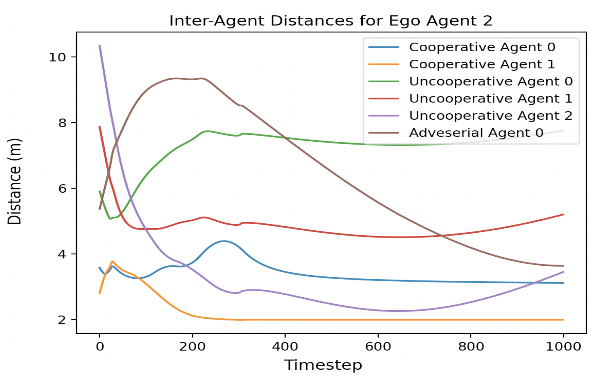

Fig. 4: Inter-Agent Distances for Ego Agent 2

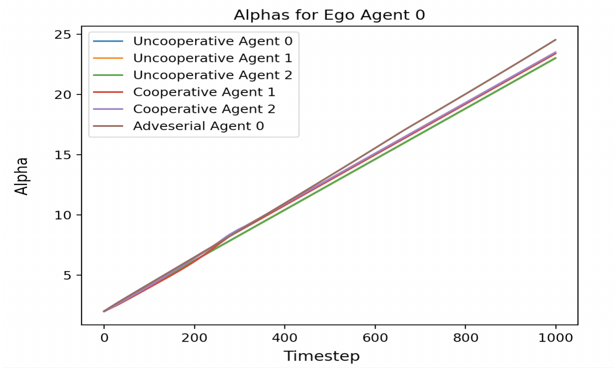

Fig. 5: Alpha for Ego Agent 0

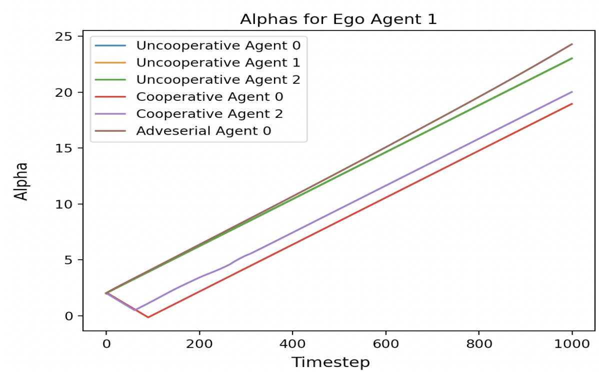

Fig. 6: Alpha for Ego Agent 1

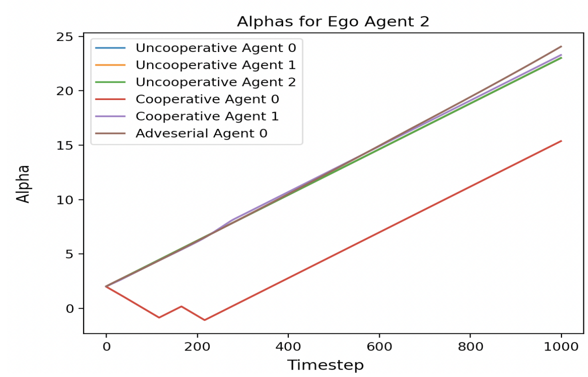

Fig. 7: Alpha for Ego Agent 2

IV. Discussion

By visual inspection of the agent trajectories in Figure 1 we can observe that the 3 uncooperative agents make no active maneuvers away from their nominal path to avoid collisions with any other agents. On the other hand, we see the cooperative (ego) agents actively avoiding both the uncooperative and adversarial agents.

The CBF parameter, α gives good insight on how much trust the ego agents have about the other agents in the environment. For example, we can see in Figures 6 and 7 that α decreases initially for the other cooperative agents. This happens because to avoid collisions with the adversarial and uncooperative agents, the cooperative agents have to maneuver closer to the other cooperative agents. This maneuver in the ego agent's perspective means the agent is deviating from its nominal path and could be adversarial. As shown in the figures, this assumption quickly corrects itself as α increases again for the cooperative agents. Although not intuitive at first, the reason why the α for the adversarial agents is increasing is because as the adversarial agent gets closer to its target, its velocity is slowing down, no longer posing a threat to the ego agent.

V. References

[1] Parwana, Hardik Panagou, Dimitra. (2022). Trust-based Rate-Tunable Control Barrier Functions for Non-Cooperative Multi-Agent Systems.