Trust-based Rate-Tunable Control Barrier Functions for Non-Cooperative Multi-Agent Systems

Hardik Parwana¹, Aquib Mustafa², and Dimitra Panagou³

Abstract—For efficient and robust task accomplishment in multi-agent systems, an agent must be able to distinguish cooperative agents from non-cooperative (i.e., uncooperative and adversarial) agents. In this paper, we first develop a trust metric based on which each agent forms its own belief of how cooperative the other agents are, i.e., of how much the other agents contribute to maintaining safety. With safety encoded as Control Barrier Functions (CBFs), the trust metric is in turn used to adjust the rate at which the CBFs allow the system trajectories to approach the boundary of the safe set. This is achieved via a novel Rate-Tunable CBF, which yields less conservative performance compared to an identity-agnostic implementation, where cooperative and non-cooperative agents are treated similarly. The proposed adaptation and control method is evaluated via simulations on heterogeneous multi-agent systems including non-cooperative agents.

I. INTRODUCTION

Teams of collaborating robots can enable tasks such as payload transportation, surveillance and exploration [1]. The ability to meet specifications such as safety and goal reaching even in the presence of non-cooperative agents is essential to successful operation and mission accomplishment. With safety specifications encoded as Control Barrier Functions (CBFs), this work develops a trust metric with which each agent distinguishes cooperative from non-cooperative agents (the latter include uncooperative and adversarial agents, as defined formally in Section III-2). The trust metric is then used to shape the response of each agent’s controller by adjusting its corresponding CBF parameters. As a motivational example, consider the scenario depicted in Fig. 1, where a green robot will be intercepted by a red robot if it follows its nominal trajectory of moving forward. The red robot wishes to cause no harm, but gives priority to its own task of moving along its own nominal trajectory. Hence if the green robot infers, based on previous observations, that the red robot is not chasing the green robot but rather moves along its nominal trajectory, then it (the green robot) can adjust its response and minimize deviation from its nominal trajectory, while maintaining safety with the red robot.

Fig. 1: The green robot is on a colliding path with the red uncooperative robot. Gradations to lighter colors show progress in time. Setting fixed values for the controller parameters (in 10) might force the green robot to back off, as in (a), and deviate significantly from its nominal trajectory. A trust-based relaxation adjusts the controller parameters to allow a more efficient motion in (b), where the green robot slows down enough to let the red robot pass first, and then continues its motion along its nominal direction.

Specifying tasks for multi-agent systems and designing safe controllers for successful execution has been an active research topic [2]–[5]. Control synthesis for multi-agent systems in the presence of non-cooperative agents has also received vivid interest, with many remarkable results [6]–[13]. In [6]–[10], the authors present resilient algorithms for flocking and active target-tracking applications, respectively. In [13] and [12], the authors design game-theoretic and adaptive control mechanisms to directly reject the effect of adversaries without the need of knowing or detecting their identity. Recently, CBF-based approaches have also been presented for achieving resilience under safety and goal-reaching objectives. In particular, [14] developed multi-agent CBFs for non-cooperative agents. [15] designed a mechanism to counter collision-seeking adversaries using CBF-based control that is robust to worst-case actions by the adversaries. All of the above studies either assume that the identities of adversarial agents is known a priori, or they design a robust response without knowing or detecting their identities; nevertheless, as also evidenced by our simulations, both cases can lead to conservative responses. Moreover, most of these works assume further that each agent knows the exact dynamics of the other agents. To the best of our knowledge, no prior studies have employed CBFs to make inferences from the behavior of surrounding agents, and tune their response to reduce conservatism while still ensuring safety.

In this paper, we make two contributions to mitigate the above limitations. First, we define a trust model based on which each agent develops its own belief of how cooperative
the other agents are. Since CBFs ensure safety by restricting the rate of change of barrier functions along the system trajectories [16], we design the trust metric based on how easily these restrictions can be imposed. Each agent rates the behavior of a neighbor agent on a continuous scale, as either 1) being cooperative, i.e., an agent that gives priority to maintaining safety, or 2) being uncooperative, i.e., an agent that does not try to collide, however, gives priority to its own tasks, and thus disregards safety, or 3) being adversarial, i.e., an agent that actively tries to collide with other agents. Second, we design an algorithm that tightens or relaxes a CBF constraint based on this trust metric, and develop Trust-based, Rate-Tunable CBFs (RT-CBFs) that allow online modification of a parameter of the class-\( K \) function employed with CBFs [16].

Our notion of trust shares inspiration with work that aims to develop beliefs on the behavior of other agents, and then use them for path planning and control. Deep Reinforcement Learning (RL) has been used to learn the underlying interactions in multi-agent scenarios [17], specifically for autonomous car navigation, to output a suitable velocity reference to be followed for maneuvers such as lane changing or merging. A few papers explicitly model trust for a multi-agent system [18]–[20]. In particular, [18] develops an intent filter for soccer-playing robots by comparing the actual movements to predefined intent templates, and gradually updating beliefs over them. Among the studies focusing on human-robot collaboration, [19] learns a Bayesian Network to predict the trust level by training on records of each robot’s performance level and human intervention. [20] defines trust by modeling a driver’s perception of vehicle performance in terms of states such as the relative distance and velocity to other vehicles, and then uses a CBF to maintain the trust above the desired threshold. The aforementioned works show the importance of making inferences from observations. However, they were either developed specifically for semi-autonomous and collaborative human-robot systems using offline data sets, or when they do consider trust, it is mostly computed with a model-free approach, such as RL. The above studies also use the computed trust as a monitoring mechanism rather than being related directly to the low-level controllers that decide the final control input of the agent and ensure safety-critical operation. In this work, we pursue a model-based design of trust metric that actively guides the low-level controller in relaxing or tightening the CBF constraints and helps to ensure successful task completion.

II. PRELIMINARIES

1) Notations: The set of real numbers is denoted as \( \mathbb{R} \) and the non-negative real numbers as \( \mathbb{R}^+ \). Given \( x \in \mathbb{R} \), \( y \in \mathbb{R}^n \), and \( z \in \mathbb{R}^{m_i \times m_e} \), \( |x| \) denotes the absolute value of \( x \) and \( ||y|| \) denotes \( L_2 \) norm of \( y \). The interior and boundary of a set \( C \) are denoted by \( \text{Int}(C) \) and \( \partial C \). For \( a \in \mathbb{R}^+ \), a continuous function \( \alpha : [0, a] \to [0, \infty) \) is a class-\( K \) function if it is strictly increasing and \( \alpha(0) = 0 \). Furthermore, if \( a = \infty \) and \( \lim_{r \to \infty} \alpha(r) = \infty \), then it is called class-\( K_{\infty} \). The angle between any two vectors \( x_1, x_2 \in \mathbb{R}^n, \forall i = 1, 2 \) is defined with respect to the inner product as \( \theta = \arccos(\frac{x_1^T \cdot x_2}{||x_1|| \cdot ||x_2||}) \). Also, \( \delta y \) represents a infinitesimal variation in \( y \).

2) Safety Sets: Consider a nonlinear dynamical system

\[
\dot{x} = f(x) + g(x)u, \tag{1}
\]

where \( x \in \mathcal{X} \subseteq \mathbb{R}^n \) and \( u \in \mathcal{U} \subseteq \mathbb{R}^m \) represent the state and control input, and \( f : \mathcal{X} \to \mathbb{R}^n, g : \mathcal{X} \to \mathbb{R}^{n_i \times m_i} \) are locally Lipschitz continuous functions. Let a set \( S \) of safe states be defined as the 0-superlevel set of a continuously differentiable function \( h(x) : \mathcal{X} \to \mathbb{R} \) as follows:

\[
S = \{ x \in \mathcal{X} : h(x) \geq 0 \}, \tag{2}
\]

\[
\partial S = \{ x \in \mathcal{X} : h(x) = 0 \}, \tag{3}
\]

\[
\text{Int}(S) = \{ x \in \mathcal{X} : h(x) > 0 \}. \tag{4}
\]

Definition 1. (CBF [21]) The function \( h : \mathcal{X} \to \mathbb{R} \) is a CBF for the system (1) on the set \( S \) defined in (4) if there exists an extended class-\( K_{\infty} \) function \( \nu \) such that

\[
\sup_{u \in \mathcal{U}} \left| \frac{\partial h}{\partial x} (f(x) + g(x)u) \right| \geq -\nu(h(x)), \forall x \in \mathcal{X}. \tag{5}
\]

Lemma 1. ([21, Theorem 2]) Let \( S \subset \mathcal{X} \) be the superlevel set of a smooth function \( h : \mathcal{X} \to \mathbb{R} \). If \( h \) is a CBF on \( S \), and \( \partial h/\partial x = 0 \) \( \forall x \in \partial S \), then any Lipschitz continuous controller for the system (1) belonging to the set

\[
K(x) = \{ u \in \mathcal{U} : \frac{\partial h}{\partial x}(f(x) + g(x)u) + \nu(h(x)) \geq 0 \}, \tag{6}
\]

renders the set \( S \) forward invariant (i.e. safe).

In this paper, we assume that \( \nu \) in (17) is given by

\[
\nu(h(x)) = \alpha h(x), \quad \alpha \in \mathbb{R}^+. \tag{7}
\]

With a slight departure from the notion of CBF defined above, \( \alpha \) is treated as a parameter in this work and its value is adjusted depending on the trust that an agent has built on other agents based on their behaviors. Since \( \alpha \) represents the maximum rate at which the state trajectories \( x(t) \) are allowed to approach the boundary \( \partial S \) of the safe set \( S \), a higher value of \( \alpha \) corresponds to relaxation of the constraint, which allows agents to get closer to each other, whereas a smaller value tightens it. The design of the trust metric is presented in Section IV.

III. PROBLEM STATEMENT

In this paper, we consider a system consisting of \( N \) agents with \( x_i \in \mathcal{X}_i \subset \mathbb{R}^{n_i} \) and \( u_i \in \mathcal{U}_i \subset \mathbb{R}^{n_i} \) representing the state and control input of the agent \( i \in \mathcal{V} = \{1, 2, ..., N\} \). The dynamics of each agent is represented as:

\[
\dot{x}_i = f_i(x_i) + g_i(x_i)u_i, \tag{8}
\]

where \( f_i : \mathcal{X}_i \to \mathbb{R}^{n_i}, g_i : \mathcal{X}_i \to \mathbb{R}^{n_i \times m_i} \) are Lipschitz continuous functions. The dynamics and the control input of an agent \( i \) is not known to any other agent \( j \in \mathcal{V} \setminus i \). However, we make the following assumption on the available observations for each agent.
Assumption 1. Let the combined state of all agents be \( x = [x_1^T, \ldots, x_N^T]^T \in \mathbb{R}^n \). Each agent \( i \) has access to the state vector \( x_i \). Furthermore, the control law for \( u_i \), and hence the closed-loop dynamics \( \dot{x}_i \), is a locally Lipschitz continuous function of \( x \), i.e., \( \dot{x}_i = F_i(x) \) where \( F_i : \mathbb{R}^n \to \mathbb{R}^n \) defines the closed-loop dynamics of agent \( i \).

The above assumption implies that each agent, including the adversarial and the uncooperative agents, have access to the states of all other agents. However, each agent computes its own control input based on its CBFs that can be different from the CBFs of its neighbors (i.e., \( h_{ij} \neq h_{ji}, \alpha_{ij} \neq \alpha_{ji} \) in Section III-A). We also have the following assumption on the estimate that agent \( i \) has regarding the closed-loop dynamics \( F_j \) of agent \( j \).

Assumption 2. Each agent \( i \) has knowledge of a set \( \hat{F}_j \) that contains an estimate of the true response \( \dot{x}_j \), i.e., such that there exists \( b_{F_j} : \mathbb{R}^n \to \mathbb{R}^+ \) for which

\[ \dot{x}_j \in \hat{F}_j = \{ \dot{x}_j \in \mathbb{R}^n \mid \| F_i(x) - \dot{x}_j \| \leq b_{F_i}(x) \}. \]

Remark 1. Assumption 2 can easily be realized for systems that exhibit smooth enough motions for a learning algorithm to train on. For example, a Gaussian Process [22] can be used to train on. For example, a Gaussian Process [22] can be used to train on.

Definition 3. (Adversarial Agent) An agent \( j \) is called adversarial to agent \( i \) if agent \( j \) chooses its control input \( u_j \), and hence its dynamics \( \dot{x}_j \), such that

\[ \frac{\partial h_{ij}}{\partial x_j} \dot{x}_j \leq 0 \quad \forall x_i \in \mathbb{R}^n_i, \quad x_j \in \mathbb{R}^n_j. \]  (11)

Definition 4. (Uncooperative agent) An agent \( j \) is called uncooperative to agent \( i \) if the control law for \( u_j \) is not a function of the state \( x_i \) of agent \( i \).

Note that the behaviors resulting from these three types of agents when more than two agents are present are not mutually exclusive, and that a cooperative agent might be forced to behave as an adversarial agent under the influence of other agents. Hence, while all the agents are assumed to behave in one of the above ways, the complex nature of interactions in a multi-agent system makes it hard to distinguish them. Moreover, it is not necessary that \( h_{ij} = h_{ji} \) and \( \alpha_{ij} = \alpha_{ji} \); for example, agents \( i \) and \( j \) may have different safety radii for collision avoidance. In addition, an agent \( i \) may perceive a cooperative agent \( j \) to be uncooperative if \( h \) cannot satisfy \( i \)'s prescribed level of safety, given by \( \alpha_{ij} \), under the influence of a truly uncooperative or adversarial agent \( k \). Therefore, rather than making a fine distinction, this work presents a trust metric that rates other agents on a continuous scale to be cooperative, adversarial, or uncooperative. Then, by leveraging the trust metric, an adaptive form of \( \alpha_{ij} \) in (10) is presented that adjusts the safety condition for agent \( i \) based on the observed behavior of agent \( j \).

3) Objective: In this subsection, we first present some definitions and then formulate the objective of this paper.

Definition 5. (Nominal Direction) A nominal motion direction \( \dot{x}_i^n \) for agent \( i \) is defined with respect to its global task and its Lyapunov function \( V_i \) as follows:

\[ \dot{x}_i^n = -\frac{\partial V_i}{\| \partial V_i \|}. \]  (12)

Definition 6. (Nominal Trajectory) A trajectory \( x_i^n(t) \) is called the nominal trajectory for an agent \( i \), if the agent follows the dynamics in (8) with some \( x_i(0) \in X \) and under control action \( u_i = -\gamma \frac{\partial V_i}{\| \partial V_i \|} \), where \( \gamma \in \mathbb{R}^+ \) denotes some design gain.

We now define the objective of this paper. The problem is as follows.

Objective 1. Consider a multi-agent system of \( N \) agents governed by the dynamics given by (1), with the set of agents denoted as \( \mathcal{V}, |\mathcal{V}| = N \), the set of intact agents denoted as \( \mathcal{N} \), the set of adversarial and uncooperative agents denoted as \( \mathcal{A} \neq \emptyset \), with \( \mathcal{V} = \mathcal{N} \cup \mathcal{A} \). Assume that the identities of the adversarial and the uncooperative agents are unknown. Each agent \( i \) has a global task encoded via a Lyapunov function \( V_i \), and \( N \) local tasks encoded as barrier functions \( h_{ij}, j \in \mathcal{V} \). Design a decentralized controller to be implemented by each intact agent \( i \in \mathcal{N} \) such that

1) \( h_{ij}(t) \geq 0, \forall t \geq 0 \), for all \( i \in \mathcal{N}, j \in \mathcal{V} \).
2) The deviation between the actual trajectory \( x_i(t) \) and the nominal trajectory \( x_i^n(t) \) for agent \( i \) is minimized, where the nominal trajectory is defined in Def. 6.

IV. METHODOLOGY

In this section, we start by introducing a standard form of a QP-based controller that can be used to enforce multiple barrier constraints simultaneously. Next, we propose our new formulation of Rate-Tunable CBFs that allow a dynamically changing class-K function to make the CBF derivative condition more or less strict. Finally, we introduce a trust metric that helps design the dynamics of the class-K function and enables adaptation in multi-agent systems.

The intact agents compute their control input in two steps. The first step computes a reference control input \( u_i^r(x) \) for an agent \( i \) that forces it to track its nominal trajectory \( x_i^r(t) \). The reference control input \( u_i^r \) is designed using the exponentially stabilizing CLF conditions as follows:

\[
    u_i^r(x) = \arg \min_{u \in U_i} \ u^T u \quad (13a)
\]

subject to

\[
    \dot{V}(x_i, u_i) \leq -kV(x_i, x_i^r), \quad (13b)
\]

where \( V_i = ||x_i - x_i^r||^2 \) and \( k > 0 \). In the second step, the following CBF-QP is defined to minimally modify \( u_i^r \) while satisfying (10) for all agents \( j \in V \setminus i \):

\[
    u_i(x, \alpha_{ij}) = \arg \min_u ||u - u_i^r||^2 \quad (14a)
\]

subject to

\[
    \frac{\partial h_{ij}}{\partial x_i} \dot{x}_i + \min_{\dot{x}_j \in P_j} \left\{ \frac{\partial h_{ij}}{\partial x_j} \right\} \geq -\alpha_{ij} h_{ij}, \quad \forall j \in V \setminus i \quad (14b)
\]

where \( \alpha_{ij} \in \mathbb{R}^+ \), and \( \dot{x}_j \) denotes all possible movements of agent \( j \) per Assumption 2. Eq. (14b) provides safety assurance for \( i \) w.r.t the worst-case predicted motion of \( j \). Henceforth, whenever we refer to the actual motion of any agent \( j \) with respect to \( i \), we will be referring to the estimate in the set \( P_j(x) \) that minimizes the contribution to safety, i.e.,

\[
    \dot{\alpha}_j = \arg \min_{\dot{x}_j \in P_j} \left\{ \frac{\partial h_{ij}}{\partial x_j} \right\}. \quad (15)
\]

In the following sections, we first propose the Tunable-CBF as a method that allows us to relax or tighten the CBF constraints while still ensuring safety. Then, we design the trust metric to adjust \( \alpha_{ij} \) and tune the controller response.

A. Rate-Tunable Control Barrier Functions

In contrast to the standard CBF where the parameter \( \alpha \) in (10) is fixed, we aim to design a method to adapt the values of \( \alpha \) depending on the trust metric that will be designed in Section IV-B. In this section, we show that treating \( \alpha \) as a state with Lipschitz continuous dynamics can still ensure forward invariance of the safe set.

**Definition 7.** (Rate-Tunable CBF) Consider the system dynamics in (1), augmented with the state \( \alpha \in \mathbb{R}^+ \) that obeys the dynamics

\[
    \begin{bmatrix}
        \dot{x} \\
        \dot{\alpha}
    \end{bmatrix} =
    \begin{bmatrix}
        f(x) + g(x)u \\
        f'(x, \alpha)
    \end{bmatrix}, \quad (16)
\]

where \( f' : \mathcal{X} \times \mathbb{R} \to \mathbb{R} \) is a locally Lipschitz continuous function w.r.t \( x, \alpha \). Let \( S \) be the safe set defined as in (4). The function \( h : \mathcal{X} \to \mathbb{R} \) is a Rate-Tunable CBF for the augmented system (16) on the set \( S \) if there exists a compact set \( \mathcal{A} \subset \mathbb{R}^+ \) such that

\[
    \sup_{\alpha \in \mathcal{A}} \left[ \frac{\partial h}{\partial x}(f(x) + g(x)u) \right] \geq -\alpha h(x) \quad \forall x \in \mathcal{X}, \alpha \in \mathcal{A} \quad (17)
\]

**Theorem 1.** Consider the augmented system (16), and a safe set \( S \) in (4) be defined by a Rate-Tunable CBF \( h \). For a locally Lipschitz continuous reference controller \( u^r : \mathcal{X} \to \mathbb{R}^m \), let the controller \( u = \pi(x, \alpha) \), where \( \pi : \mathcal{X} \times \mathbb{R}^+ \to \mathcal{U} \) be formulated as

\[
    \pi(x, \alpha) = \arg \min_{u \in \mathcal{U}} ||u - u^r(x)||^2 \quad (18a)
\]

subject to

\[
    \frac{\partial h}{\partial x}(f(x) + g(x)u) \geq -\alpha h(x) \quad (18b)
\]

where \( \frac{\partial h}{\partial x} \) is also locally Lipschitz continuous. Suppose there exists \( \bar{h} > 0 \) such that \( h(x) < \bar{h} \) \( \forall x \in S \). Then, the set \( S \) is forward invariant.

**Proof.** Step 1: Since \( h \) is a Rate-Tunable CBF, according to (17), there always exists a \( u \in \mathbb{R}^m \) that satisfies the constraint of the QP (18b). The solution \( \pi(x, \alpha) \) is also unique as the optimization problem is strictly convex. Since \( f, g, h, \frac{\partial h}{\partial x} \) are locally Lipschitz, by composition, \( \frac{\partial h}{\partial x} f(x) \) and \( \frac{\partial h}{\partial x} g(x) \) are also locally Lipschitz. By [16, Thm. 3], it follows that \( \pi \) is locally Lipschitz continuous.

Step 2: Since \( \pi \), and hence the closed-loop dynamics, is locally Lipschitz continuous, (16) admits a unique solution over its interval of existence. Let us consider the evolution of \( h \) trajectory for this system now. Since the set \( \mathcal{A} \) is compact, \( \exists \gamma \in \mathbb{R}^+ \) such that \( -\alpha h \geq -\gamma h \), \( \forall \alpha \in \mathcal{A} \). Consider the system governed by \( \bar{h} = -\gamma h \). From [23, Lemma 4.4], this systems admits a unique solution for all \( t > 0 \) and is lower bounded by 0. Therefore, using comparison lemma [23, Lemma 3.4], the systems governed by \( h \geq -\alpha h \) and \( h \geq -\alpha h \), over their respective interval of existence, are also lower bounded by 0. Since \( h \) is also upper bounded, it evolves on a compact set \( \mathcal{S} = \{ h \ | 0 \leq h \leq \bar{h} \} \). Therefore, from [23, Thm 3.2] the system \( h \geq -\alpha h \) admits a solution globally, i.e., \( \forall t > 0 \). Hence the set \( S \) is forward invariant.

□

B. Design of Trust Metric

The margin by which the CBF condition (10) is satisfied, i.e., the value of \( \dot{h}_{ij} + \alpha_{ij} h_{ij} \), as well as the best action that agent \( i \) can implement to increase this margin for a given motion \( \dot{x}_j \) of agent \( j \), are both important cues that help infer the nature of agent \( j \). The adaptation of the parameters \( \alpha_{ij} \) by agent \( i \) depends on the following:

1) **Allowed motions of agent \( j \):** The worst that an agent \( j \) is allowed to perform in agent \( i \)'s perspective is the critical point beyond which agent \( i \) cannot find a feasible solution...
to (14). This is formulated as follows:

\[
\frac{\partial h_{ij}}{\partial x_j} \dot{x}_j \geq -\alpha_{ij} - \frac{\partial h_{ij}}{\partial x_i} \dot{x}_i, \tag{19}
\]

\[
\geq -\alpha h - \max_{\hat{n}_i} \left\{ \frac{\partial h_{ij}}{\partial x_i} \dot{x}_i \right\}. \tag{20}
\]

This gives a lower bound that, if violated by agent \( j \), will render agent \( i \) unable to find a feasible solution to (14). The required maximum value in Eq. (20) can be obtained from the following Linear Program (LP):

\[
\max_{\hat{n}_i} \frac{\partial h_{ij}}{\partial x_i} \dot{x}_i \tag{21a}
\]

s.t. \( \hat{h}_{ik} \geq -\alpha_{ik} h_{ik}, \ \forall k \in \mathcal{V} \setminus i,j. \tag{21b} \]

Equation (20) represents a halfspace whose separating hyperplane has the normal direction \( \hat{s} = \frac{\partial h_{ij}}{\partial x_j} \), and has been visualized in Fig. 2.

(2) Actual motion of agent \( j \): This is given by Eq. (15).

(3) Nominal behavior of agent \( j \): Suppose that agent \( i \) has knowledge of agent \( j \)'s target state. A nominal direction of motion \( \hat{n}_j \) of agent \( j \) in agent \( i \)'s perspective can be obtained using the Lyapunov function \( V_i^{(j)} = ||x_j - x_j^{(0)}||^2 \) as:

\[
\hat{n}_j = -\frac{\partial V_i^{(j)}}{\partial x_j} / ||\partial V_i^{(j)}||. \tag{22}
\]

Knowledge of nominal direction plays an important role in shaping belief as it helps to distinguish between uncooperative and adversarial agents. Note that infinitely many values of \( \dot{x}_j \) may satisfy the CBF derivative condition in the same way (i.e., will have the same \( d_{a_j}^\epsilon \) in Fig. 2 as explained later). This gives an additional degree of freedom for deciding which possible motion is more trustworthy. If \( \dot{x}_j \) is behaving worse than the nominal direction, then it is a cause of concern as it is not consistent with the belief of agent \( i \) for agent \( j \). Note that \( x_j^{(0)} \) need not be the true target state of agent \( j \) for our algorithm to work since if \( x_j^{(0)} \) is not known, we can consider the worst-case scenario where \( x_j^{(0)} = x_i \), i.e., that \( j \) is an adversary and targets to collide with agent \( i \); over time, our algorithm will learn that \( j \) is not moving along \( \dot{x}_j^{(0)} \), and therefore increases \( i \)'s trust of \( j \).

The two quantities of interest from Fig. 2 are the distance of \( a_j \) from hyperplane, which is the margin by which the CBF condition (10) is satisfied, and the deviation of the actual direction \( a_j^\epsilon \) of agent \( j \) with respect to the nominal direction \( \hat{n}_j \).

1) Distance-based Trust Score: Let the half-space in (20) be represented in the form \( Av \geq b \) with \( v = \dot{x}_j \) and \( A,b \) defined accordingly. The distance \( d_{n}^\epsilon \) of a vector \( v \) from the dividing hyperplane (see Fig. 2) is given by

\[
d_{n}^\epsilon = Av - b \tag{23}
\]

where \( d_{n}^\epsilon < 0 \) is an incompatible vector, corresponding to scenarios that should never happen. For \( d_{n}^\epsilon > 0 \), its numerical value expresses the margin by which the CBF constraint is satisfied. Therefore, the distance-based trust score is designed as

\[
\rho_d = f_d(d_{n}^\epsilon), \tag{24}
\]

where \( f_d : \mathbb{R}^+ \rightarrow [0,1] \) is a monotonically-increasing Lipschitz continuous function, such that \( f_d(0) = 0 \). A possible choice would be \( f_d(d) = \tanh(\beta d) \), with \( \beta \) being a scaling parameter.

2) Direction-based Trust Score: Suppose the angle between the vectors \( \hat{n} \) and \( \hat{s} \) is denoted by \( \theta_{s}^{n} \), and the angle between \( a_j^{\epsilon} \) and \( \hat{n} \) is denoted by \( \theta_{s}^{\epsilon} \). The direction-based trust is designed as

\[
\rho_{\theta} = f_{\theta}(\theta_s^{\epsilon}/\theta_s^{n}), \tag{25}
\]

where \( f_{\theta} : \mathbb{R}^+ \rightarrow [0,1] \) is again a monotonically-increasing Lipschitz continuous function with \( f_{\theta}(0) = 0 \). Note that even if \( \theta_s^{\epsilon} = \theta_s^{n} \), i.e., \( j \) is perfectly following its nominal direction, we do not design \( f_{\theta} \) to be 1 as the robot might be uncooperative. We give \( f_{\theta} \) higher values when \( \theta_s^{\epsilon} < \theta_s^{n} \), as with \( a_j^{\epsilon} \) in Fig. 2, \( j \)'s motion is performing worse for inter-robot safety than its nominal movement for improved safety, thus leading to a higher score. Finally, when \( \theta_s^{\epsilon} < \theta_s^{n} \), as with \( a_{j_1} \) in Fig. 2, then \( j \)'s current motion is performing worse for inter-robot safety than its nominal motion and is therefore either uncooperative/adversarial or under the influence of other robots, both of which lead to lower trust.

3) Final Trust Score: The trust metric is now designed based on \( \rho_d \) and \( \rho_{\theta} \). Let \( \bar{\rho}_d \in (0,1) \) be the desired minimum robustness in satisfying the CBF condition. Then, the trust metric \( \rho \in [-1,1] \) is designed as follows:

\[
\rho = \begin{cases} 
(\rho_d - \bar{\rho}_d)\rho_{\theta}, & \text{if } \rho_d \geq \bar{\rho}_d, \\
(\rho_d - \bar{\rho}_d)(1-\rho_{\theta}), & \text{if } \rho_d < \bar{\rho}_d. 
\end{cases} \tag{26}
\]

Here, \( \rho = 1,-1 \) represent the absolute belief in another agent being cooperative and adversary, respectively. If \( \rho_d > \bar{\rho}_d \),
then we would like to have $\rho > 0$, and its magnitude is
calculated by $\rho \theta$ with smaller values of $\rho \theta$ conveying low trust.
Whereas, if $\rho_d < \rho_d$, then we would like the trust factor to be
negative. A smaller $\rho \theta$ in this case implies more distrust and
should make the magnitude larger, hence the term $1 - \rho \theta$.

The trust $\rho$ is now used to adapt $\alpha$ with following equation

$$\dot{\alpha}_{ij} = f_{\alpha_{ij}}(\rho), \tag{27}$$

where $f_{\alpha_{ij}} : [-1, 1] \rightarrow \mathbb{R}$ is a monotonically increasing
function. A positive value of $f_{\alpha_{ij}}$ relaxes the CBF condition by
increasing $\alpha$, and a negative value decreases $\alpha$. The framework that each agent $i$ implements at every time $t$ is
detailed in Algorithm 1.

\begin{algorithm}
\caption{Trust-based Multi-agent CBF}
\begin{algorithmic}[1]
\REQUIRE $x, i, f_0, f', \rho_d$
\STATE $\alpha_{ij} \leftarrow$ Current value of CBF Parameters
\FORALL time \DO
\FOR $j \in \mathcal{V} \setminus i$ \DO
\STATE Predict $\alpha_j$ \Comment{Actual movement of $j$}
\STATE Compute $\rho_0, \rho_d$ with Eqns. (25), (24)
\STATE Compute $\rho$ with (26) \Comment{Trust score}
\STATE Calculate nominal input using (13).
\STATE Implement $u_i$ from (14).
\STATE Update $\alpha_{ij}$ with update rule (27)
\ENDFOR
\ENDFOR
\end{algorithmic}
\end{algorithm}

\textbf{Remark 2.} Note that equation (26) does not represent a
locally Lipschitz continuous function of $\rho_0, \rho_d$. This poses
theoretical issues as having a non-Lipschitz gradient flow in
(27) would warrant further analysis concerning existence of
unique solution for the combined dynamical involving $x$ and $\alpha$.
Hence, while (26) represents our desired characteristics,
we can use a sigmoid function to switch across the boundary
$\rho_d = \rho_d$ thereby holding the assumptions in Theorem 1 true.

\section{C. Sufficient Conditions for a Valid trust Metric}

In the presence of multiple constraints, feasibility of the
QP (14) might not be guaranteed even if the control input is
unbounded and the states are far away from boundary, i.e.,
h_{ij} >> 0. In this section, we derive conditions that guarantee
that the QP does not become infeasible.

\textbf{Assumption 3.} Suppose the initial state satisfies the barrier
constraints strictly, i.e., $h_{ij}(x_i(0), x_j(0)) > 0$, $j \in \mathcal{V} \setminus i$. Now
suppose at each time $t$, there always exist $\alpha_{ij} > 0$ such that
the QP (14) is feasible, then $h_{ij}(x_i(t), x_j(t)) > 0 \ \forall t > 0 .

\textbf{Theorem 2.} (Compatibility of Multiple CBF Constraints)
Under Assumptions 1-3, consider the system dynamics in (1)
subject to multiple barrier functions $h_j, j \in \mathcal{V}_M = 1, 2, \ldots, M$.
Let the controller $u(x, \alpha)$, with $\alpha = \{\alpha_j | j \in \mathcal{V}_M\}$, be
formulated as

$$u(x, \alpha) = \arg \min_u ||u - u^*||^2 \tag{28a}$$

s.t. $\dot{h}_j(x, u) \geq -\alpha_j h_j(x), \forall j \in \mathcal{V}_M$. \tag{28b}

with corresponding parameters $\alpha_j$. Now suppose $\alpha_j$, as an
augmented state, has following dynamics

$$\dot{\alpha}_j = f_j(x, \alpha_1, \alpha_2, \ldots, \alpha_M). \tag{29}$$

Suppose the solution of QP in (28), the control $u(x, \alpha) \in \mathcal{U}$,
is always locally Lipschitz continuous in $x$ and $\alpha$ so that the
closed-loop dynamics of $x$ be denoted as $\dot{x} = F(x)$
is locally Lipschitz continuous. Let $L_{h_j}, L_{\dot{h}_j}$, and $L_F$ be
the Lipschitz constants of $h_j(x), \dot{h}_j(x, \dot{x})$ and $F(x)$. In case of
imperfect information, we assume an estimate $\hat{F}(x)$ is available
such that $||F(x) - \hat{F}(x)|| \leq b_{\hat{F}}(x)$. Further, assume that
solution of Suppose a solution to (28) exists at initial
time $t = 0$ and $\dot{\alpha}_j$ satisfies the following condition with
$B(x) = ||\dot{F}(x)|| + b_{\hat{F}}(x)$ (see Assumption 2)

$$\dot{\alpha}_j \geq \frac{-(\delta x_j + L_{\dot{h}_j} B(x)(L_F + 1) + \alpha_j L_{h_j} B(x))}{L_{h_j}}. \tag{30}$$

Then, a solution to (28) continues to exist until $h_j = 0$ for
any $j \in \mathcal{V}_M$.

\textbf{Proof.} Based on Lipschitz property, we have $\delta \dot{x} \leq L_F ||\delta x||$
where $\delta y$ denotes infinitesimal variation in $y$. In order to have
a feasible solution for QP in (28), it should hold that

$$d_j(x, \dot{x}) = \dot{h}_j(x, \dot{x}) + \alpha_j h_j(x) \geq 0. \tag{31}$$

The variation in $d_j(x, \dot{x})$ is given by

$$\delta d_j = \delta \dot{h}_j + h_j \delta \alpha_j + \alpha_j \delta h_j. \tag{32}$$

For a feasible solution to QP in (28) with variation in $\delta d_j$,
one needs $d_j + \delta d_j \geq 0$. Thus, the following condition holds

$$\delta \alpha_j \geq \frac{-(\delta d_j + \delta h_j + \alpha_j \delta h_j)}{h_j}. \tag{33}$$

In the multi-agent setting, since each constraint expresses
pairwise safety among the robots $j_1, j_2 \in \{1, 2, \ldots, N\}$, it
follows that $h_j$ depends only on a subset of the state vector $x$
only on $x_j, x_{j_2}$. However, the evolution of state dynamics $\dot{x}$
depends on the control input $u$ computed based on the
QP formulation in (28), where $u$ depends on all constraints
$h_j$ and states $x_j$. Thus, $\delta h_j$ and $\delta \dot{x}$ are dependent on
the states of all agents and not only on $x_j, x_{j_2}$. This introduces
coupling between the constraints, which is hard to solve for
analytically. However, given that we assume that $\dot{x} = F(x)$
with a known Lipschitz constant $L_F$, we can decouple all
constraints. Using $||\delta \dot{x}|| \leq L_{\hat{F}} ||\delta x||, ||\delta h_j(x)|| \leq L_{h_j} ||\delta x||,$
and

$$||\delta h_j(x, \dot{x})|| \leq L_{h_j} \left[ ||\delta x|| + ||\delta \dot{x}|| \right] \leq L_{\hat{h}_j} (||\delta x|| + ||\delta \dot{x}||),$$

we can write (33) as
δα_j ≥ \frac{-(d_j + Lh_j(\|\delta x\| + ||\delta x||) + α_jLh_j ||\delta x||)}{h_j}.

Finally, since we are interested in infinitesimal variations with time, the change \( \delta x \) is specifically given by the dynamics as \( \delta x = F(x) \). From Assumption 2, we further have a bound available on \( ||F(x)|| \) which is \( B(x) = ||\delta F(x)|| + b_F(x) \). Therefore, if \( \alpha \) satisfies (30), we have that \( d_j + βd_j \geq 0 \), \( ∀j \) and hence a solution for QP in (28) exists. This completes the proof.

**Remark 3.** Consider a boundary condition scenario, i.e., when \( h_j \to 0 \) for which \( α_j \to ∞ \). Such scenario can happen, for example, when an agent is surrounded by adversaries and has no escape path. In order to keep the QP feasible, the agent will have to keep increasing \( α_j \) until \( h_j = 0 \) and then collision is unavoidable. We would expect that if there were an escape direction, then at least one of \( α_j \) would have lower bound \( ≤ 0 \) before \( h_j = 0 \) is reached. Note that in case of \( α_j \to ∞ \) in finite time, the assumption on local Lipschitzness of \( u \) and \( h_j \) might no longer hold. Such an analysis will be formally addressed in future work.

V. Simulation Results

The proposed Algorithm 1 is implemented to design controllers for waypoint navigation of a group of three intact robots while satisfying desired constraints. The intact agents are modeled as unicycles with states given by the position coordinates \( p_x, p_y \) and the heading angle \( ψ \) w.r.t. a global reference frame, \( \dot{p}_x = v \cos θ, \dot{p}_y = v \sin θ, ψ = ω \), where \( v, ω \) are linear and angular velocity expressed in the body-fixed frame, and act as control inputs. The adversarial and uncooperative agents are modeled as single integrators with dynamics \( \dot{p}_x = v_x, \dot{p}_y = v_y \), where \( v_x, v_y \) are velocity control inputs. Fig. 3 shows a scenario where an adversarial robot chases one of the intact agents, and two other uncooperative agents move along horizontal paths without regard to any other agent. The nominal trajectories in this case are straight lines from initial to target locations. None of the intact robots know the identity of any other robot in the system, and initialize \( α_{ij} \to 0.8 \) uniformly. It can be seen that intact agents are successfully able to remain close to the nominal paths and reach their target location in given time. For comparison, we also show trajectories for fixed \( α \) (non-adaptive) case. For \( α = 0.8 \), no solution exists to the QP (14) after some time, and the simulation fails. Therefore, we simulate another scenario where we assume to know identities of agents beforehand. \( α \) for adversaries is chosen to be \( 2.0 \) for intact (cooperative) agents, and for \( 0.8 \) for others. Even with these choices for \( α \), the intact agents fail to reach the goal and diverge away from their nominal paths. Note that our method saves us the effort to identify agents and then choose proper \( α \) for them. Fig. 4, 5, and 6 illustrate the variation of trust metric, \( α_{ij}s \), and CBFs with time. The adversarial agent uses an exponentially stabilizing CLF to chase agent 1. The reference velocity for unicycles using was computed with a controller of the form: \( v = k_xe_x, ω = k_ω(\arctan(e_y/e_x) - ψ) \), where \( k_x, k_ω > 0 \) are gains, both taken as 2.0 in simulations, \( e_x, e_y \) are position errors of the agent to its nominal trajectory in X,Y coordinates respectively. The trust metric was computed using (26) with \( \bar{ρ}_d = 1.0 \). We use first-order barrier functions for the unicycle model [24] to enforce collision avoidance, and \( f_d = \tanh(d), f_θ = \tanh(θ^a/θ^p) \) in Eqns. (24)-(25). The simulation video and code can be found at https://github.com/hardikparwana/Adversary-CBF.

Assumption 2 was realized by assuming (and verifying through observations) that the maximum normed difference between \( \dot{x}_i(t) \), which is to be predicted, and \( \dot{x}_i(t - Δt) \), is 10% of the norm of \( \dot{x}_i(t - Δt) \). This is reasonable as the simulation time step \( Δt \) is 0.05 sec, and because the adversaries and intact agents all use Lipschitz continuous controllers. Note that the case with fixed \( α \) in Fig. 3 uses the same assumption to design a controller, but still fails to reach the goal.

Fig. 3: Intact agents in formation navigating through an environment with non-cooperative agents. The timestamp of different points on the trajectory is given by the colormap. The bold colors show the path resulting from the proposed method. The paths with increased transparency result from application of CBFs with fixed \( α \) in (14).

A. Conclusion and Future Work

This paper introduces the notion of trust for multi-agent systems where the identity of robots is unknown. The trust metric is based on the ease of satisfaction of CBF constraints. It also provides a direct feedback to the low-level controller and help shape a less conservative response while ensuring safety. The effect of input constraints and the sensitivity of the algorithm to its parameters and function choices will be evaluated in future work.

REFERENCES

[1] Y. Rizk, M. Awad, and E. W. Tustel, “Cooperative heterogeneous multi-robot systems: A survey,” ACM Computing Surveys (CSUR), vol. 52, no. 2, pp. 1–31, 2019.
Fig. 4: Trust between different pairs of robots. Note that two agents need not have the same level of trust for each other. The initial trust for adversary is positive. This is because the initial value of $\alpha = 0.8$ is already a very conservative value and trust based relation allows agent 1 to not deviate too much from nominal trajectory but still avoids collision.

![Graph showing trust between different pairs of robots](image)

Fig. 5: Variation of the parameters $\alpha_{ij}$ of Robot 1 with respect to Robot $j=2,3$ and the adversarial agent. Trust-based adaptation allows robot 1’s $\alpha$ to increase, thus relaxing the constraints.

[2] L. Lindemann and D. V. Dimarogonas, “Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks,” IEEE control systems letters, vol. 3, no. 3, pp. 757–762, 2019.

[3] ——, “Barrier function based collaborative control of multiple robots under signal temporal logic tasks,” IEEE Transactions on Control of Network Systems, vol. 7, no. 4, pp. 1916–1928, 2020.

[4] I. Saha, R. Ramaihittama, V. Kumar, G. J. Pappas, and S. A. Seshia, “Automated composition of motion primitives for multi-robot systems from safe ltl specifications,” in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2014, pp. 1525–1532.

[5] J. R. Bruce and M. M. Veloso, “Safe multirobot navigation within dynamics constraints,” Proceedings of the IEEE, vol. 94, no. 7, pp. 1398–1411, 2006.

[6] F. Pasqualetti, A. Bicchi, and F. Bullo, “Consensus computation in unreliable networks: A system theoretic approach,” IEEE Transactions on Automatic Control, vol. 57, no. 1, pp. 90–104, 2011.

[7] S. Sundaram and C. N. Hadjicostis, “Distributed function calculation via linear iterative strategies in the presence of malicious agents,” IEEE Trans. on Automatic Control, vol. 56, no. 7, pp. 1495–1508, July 2011.

[8] J. Usevitch and D. Panagou, “Resilient leader-follower consensus to arbitrary reference values in time-varying graphs,” IEEE Transactions on Automatic Control, vol. 65, no. 4, pp. 1755–1762, 2019.

[9] K. Saultier, D. Saldana, P. Prorok, G. J. Pappas, and V. Kumar, “Resilient flocking for mobile robot teams,” IEEE Robotics and Automation letters, vol. 2, no. 2, pp. 1039–1046, 2017.

Fig. 6: Variation of barrier functions of Robot 1 with time. The plots represent the value of barrier function used by agent 1 for collision avoidance with agent 2,3 and the adversarial agent. The barriers with uncooperative agents are not shown but can be found along with our videos. Trust-based relaxation allows agents to go closer to the safe set boundary ($h_{1j} = 0$) compared to fixed $\alpha$ case, and hence leads to less conservative response while still guaranteeing safety.

[10] L. Zhou, V. Troumas, G. J. Pappas, and P. Tokekar, “Resilient active target tracking with multiple robots,” IEEE Robotics and Automation Letters, vol. 4, no. 1, pp. 129–136, 2018.

[11] L. Guerrero-Bonilla and V. Kumar, “Realization of $r$-robust formations in the plane using control barrier functions,” IEEE Control Systems Letters, vol. 4, no. 2, pp. 343–348, 2019.

[12] A. Mustafa and H. Modares, “Attack analysis and resilient control design for discrete-time distributed multi-agent systems,” IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 369–376, 2019.

[13] M. Pirani, E. Nekouei, S. M. Dibaji, H. Sandberg, and K. H. Johansson, “Design of attack-resilient consensus dynamics: a game-theoretic approach,” in 2019 18th European Control Conference (ECC). IEEE, 2019, pp. 2227–2232.

[14] U. Bormann, L. Wang, A. D. Ames, and M. Egerstedt, “Control barrier certificates for safe swarm behavior,” IFAC-PapersOnLine, vol. 48, no. 27, pp. 68–73, 2015.

[15] J. Usevitch and D. Panagou, “Adversarial resilience for sampled-data systems using control barrier function methods,” in 2021 American Control Conference (ACC). IEEE, 2021, pp. 758–763.

[16] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” IEEE Trans. on Automatic Control, vol. 62, no. 8, pp. 3861–3876, 2016.

[17] B. Brito, A. Agrawal, and J. Alonso-Mora, “Learning interaction-aware guidance policies for motion planning in dense traffic scenarios,” arXiv preprint arXiv:2107.04538, 2021.

[18] A. Valtazanos and S. Ramamoorthy, “Intent inference and strategic escape in multi-robot games with physical limitations and uncertainty,” in 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2011, pp. 3679–3685.

[19] M. Fooladi Mahani, L. Jiang, and Y. Wang, “A bayesian trust inference model for human-multi-robot teams,” International Journal of Social Robotics, pp. 1–15, 2020.

[20] C. Hu and J. Wang, “Trust-based and individualizable adaptive cruise control using control barrier function approach with prescribed performance,” IEEE Trans. on Intelligent Transportation Systems, 2021.

[21] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in 2019 European Control Conference. IEEE, 2019, pp. 3420–3431.

[22] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, “Information-theoretic regret bounds for gaussian process optimization in the bandit setting,” IEEE transactions on information theory, vol. 58, no. 5, pp. 3250–3265, 2012.

[23] H. K. Khalil, “Nonlinear systems third edition,” Prentice Hall, 2002.

[24] J. R. Bruce and M. M. Veloso, “Realization of $r$-robust formations in the plane using control barrier functions,” in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2014, pp. 68–73, 2015.