Safe Multi-Agent Collaborative Transport Without Communication

Albert Li, Bibit Bianchini, Lauren Luo

Abstract— **In order to achieve effective coordination between heterogeneous robot teams and humans, strategies for prediction, decentralized control, and safety must be studied together. This paper analyzes the multi-agent collaborative transport problem, wherein a team of agents must work together to safely move a load from one location to another without communication. In particular, this setup assumes the leader blindly tracks some desired trajectory without regard for safety and the followers must protect the load. The paper's contributions include a general mathematical framework for analyzing safety given decentralized dynamical predictions, a heuristically-motivated method for quantifying trust in a team, a robustness analysis of the resulting strategies with quantitative criteria for design parameter selection, and simulations validating the performance of the proposed method.**

I. Introduction

A. Motivation

In recent years, significant advances have allowed multi-robot systems to effectively automate tasks that are difficult, expensive, or time-consuming for human or single-robot agents. Among these tasks are area coverage for surveillance and emergency monitoring, mapping, and load transport. Novel mathematical tools have been developed to ensure the stability of these objectives, including the linear consensus protocol (LCP) for rendezvouz, LCP derivatives for formation control, weighted protocols for connectivity maintenance, and Voronoi partitioning methods [1], [2].

In parallel, the field of safe control has blossomed with new developments in *control barrier functions* (CBFs), which can formally guarantee safety in dynamical control systems [3]. However, while CBFs are effective when a system's dynamics are well-characterized, these guarantees erode as model quality and accuracy decreases. Additionally, many multi-agent control strategies assume the existence of a communication network which may not be present in teams with heterogeneous composition or with humans.

To that end, this paper studies the multi-agent *collaborative transport* task, wherein a team of agents composed of humans and robots work together to transport a load that may be cumbersome or impossible for a single agent to move alone. In particular, the analysis presumes that there is an explicitly designated *leader* who may guide the load freely and be assisted and/or protected by a group of *followers*. This setup is common in tasks where a human leader may have an intended objective that is difficult to explicitly communicate to robotic assistants in unstructured environments with obstacles or hazards.

The authors are with the Department of Mechanical Engineering at Stanford University, Stanford, CA, USA. {ahli,bibit,jlluo} @stanford.edu.

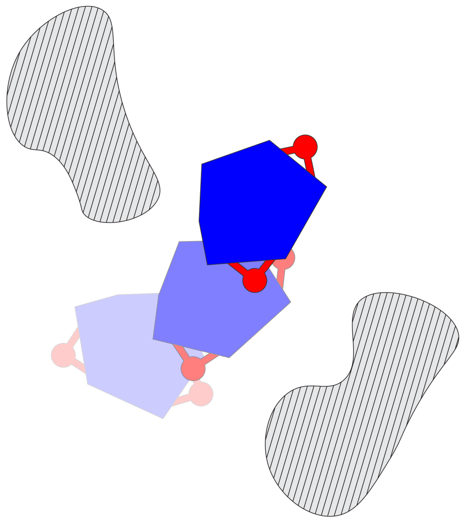

Fig. 1: A typical collaborative transport task. Two agents (red) must work together to transport a load (blue) in an environment with obstacles or hazards (shaded). Typically, the agents will also have asymmetric access to information due to factors like visual occlusions or sensor availability, so accurate predictions of the team behavior are necessary for success.

B. Related Work

CBFs have been used to control multi-robot systems featuring communication networks with great success. A collision avoidance algorithm that was minimally invasive with respect to a nominal control action was implemented in [4]. A minimally-invasive formation control algorithm with communication delays was implemented in [5]. An approach to encoding many objectives like collision avoidance and proximity maintenance in one function was presented in [6].

Multi-agent collaborative transport has also been previously studied, such as for the furniture-moving problem [7] or with the inclusion of human teammates using methods like programming by demonstration [8] and human intent modeling via neural networks [9]. On the other hand, many methods use no prediction model, instead assuming a designated leader always executes safe actions. For example, in [10], the followers apply a simple consensus-like control law that guarantees convergence to the leader's desired trajectory. Similarly, [11] implements an decentralized adaptive method that allows agents to learn the load properties and achieve trajectory tracking.

Despite these successes, ensuring formal safety guarantees given decentralized predictions remains an unsolved problem. This work therefore seeks to unify safe multi-robot approaches with predictive methods suitable for human-robot collaboration.

C. Contributions

The contributions of this paper with respect to the aforementioned prior work are

- a generalizable prediction-based framework for safe decentralized collaborative control,
- a heuristically-motivated method to dynamically update each agent's trust in the team,
- a robustness analysis of the proposed method including controller design heuristics, and
- simulations characterizing the strategy's effectiveness.

II. Problem Setup

A. Notation

The notation in the remainder of this paper abides by the conventions described here. The analysis is conducted in $SE(2)$, where (x, y) denote position and θ orientation. If not otherwise specified, these variables refer to the carried load's center of mass (COM) states. The load mass and moment of inertia about its COM are denoted m and I respectively. The total number of agents is denoted n. For continuous-time representations, the explicit dependence on time t is omitted. For clarity, discrete-time variables will show the dependence on time step k using brackets, e.g. $x[k]$. Components of vectors are written as superscripts while subscripts are reserved for designating the agent of interest. For example, F^x corresponds to the x component of force while F_i corresponds to the force exerted by agent i.

B. Joint Dynamical Model

The joint system consists of the load, which follows rigid body dynamics in the plane, and the leader and follower agents, which are infinitesimally small, massless, and rigidly attached to the load. Friction is ignored, as it does not significantly affect the analysis. External disturbances are also not modeled. Thus, the joint dynamics can be expressed:

$$m\ddot{x} = \sum_{i=1}^{n} F_i^x, \tag{1}$$

$$m\ddot{y} = \sum_{i=1}^{n} F_i^y, \tag{2}$$

$$I\ddot{\theta} = \sum_{i=1}^{n} r_i \times F_i. \tag{3}$$

The joint state refers to the following values with respect to the load COM:

$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix}, \tag{4}$$

which allows the following state-space representation of the joint dynamics:

$$\dot{z} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} z_2 \\ \frac{1}{m}\sum_{i=1}^{n} F_i^x \\ z_4 \\ \frac{1}{m}\sum_{i=1}^{n} F_i^y \\ z_6 \\ \frac{1}{I}\sum_{i=1}^{n} r_i \times F_i \end{bmatrix}. \tag{5}$$

C. Assumptions

Necessary assumptions are detailed in this section. First, each robot is equipped with the following sensors:

- inertial measurement units (IMUs),
- cameras, and
- laser rangefinders.

Each robot is assumed to possess the following capabilities and information:

- the ability to perform simultaneous localization and mapping for accurate position estimates,
- accurate estimates of the load's physical properties and COM position,
- accurate load velocity estimation techniques given local velocity measurements,
- the ability to independently apply a net torque by applying forces at two distinct locations, and
- knowledge of the number of total agents.

With the above sensors and capabilities, the robot can measure the joint state z. Though some of these assumptions are generous, recent results using methods like adaptive control detail procedures for decentralized estimation of some of the required information [11].

The assumption that each robot can apply two forces is summarized by the following equations:

$$F_i^x = F_{i,1}^x + F_{i,2}^x, \tag{6}$$

$$F_i^y = F_{i,1}^y + F_{i,2}^y, \tag{7}$$

$$M_i = r_{i,1} \times F_{i,1} + r_{i,2} \times F_{i,2}, \tag{8}$$

where $F_{i,1}$ and $F_{i,2}$ correspond to the forces applied by robot i at its first and second attachment points respectively and $r_{i,1}, r_{i,2}$ denote the locations of those attachment points relative to the load COM.

III. Mathematical Preliminaries

A. Definitions

This section will summarize the theory of control barrier functions. First, the following definitions will prove useful:

Definition 1 (Class-$\mathcal{K}$ Function)**.** $\alpha : [0, a) \to [0, \infty]$ is a *class-$\mathcal{K}$ function* if $\alpha(0) = 0$ and α is monotonically increasing. Further, $\alpha : (-c, d) \to (-\infty, \infty),\ c, d > 0$ is an *extended class-$\mathcal{K}_\infty$ function* if it satisfies the same properties as a class-$\mathcal{K}$ function over its extended domain.

Definition 2 (Lie Derivative)**.** The *Lie Derivative* evaluates the change of a tensor field along the flow of another vector field. For the purposes of this paper, it serves as a convenient notational shorthand relating two sufficiently smooth functions $h(z), f(z)$:

$$L_f h = \frac{\partial h}{\partial z} \cdot f(z). \tag{9}$$

This can be equivalently interpreted as a directional derivative, though the Lie derivative is more general.

Definition 3 (Forward Invariance)**.** A set $\mathcal{C}$ is *forward invariant* if for every $z_0 \in \mathcal{C}$, $z(t) \in \mathcal{C}$ for $z(0) = z_0, \forall t \geq 0$.

B. Control Barrier Functions

The summary of control barrier functions in this section follows from [3]. Safety can be interpreted as the enforcement of the invariance of a *safe set*

$$\mathcal{C} = \{z \in \mathcal{D} \mid h(z) \geq 0\}. \tag{10}$$

By convention, the safe set is the 0-superlevel set of a continuously differentiable function $h : \mathcal{D} \to \mathbb{R}$ which maps states in the domain $\mathcal{D}$ to a scalar. The dynamical control system $\dot{z} = f(z, u)$ is *safe* if $\mathcal{C}$ is forward invariant. This allows the definition:

Definition 4 (Control Barrier Function)**.** Let $\mathcal{C} \subset \mathcal{D} \subseteq \mathbb{R}^n$ be the 0-superlevel set of a continuously differentiable function $h : \mathcal{D} \to \mathbb{R}$. Then, h is a *control barrier function* if there exists an extended class-$\mathcal{K}_\infty$ function α such that for the control-affine dynamics $\dot{z} = f(z) + g(z)u$,

$$\sup_u \{L_f h(z) + L_g h(z)u \geq -\alpha(h(z))\} \tag{11}$$

for all $z \in \mathcal{D}$.

The intuition for this definition is as follows: if the system is unsafe ($h(z) < 0$), then the safety dynamics are constrained to be evolving towards the safe set ($\dot{h}(z) > 0$). Further, the rate at which the value of h approaches the safe set increases with a decrease in h due to the monotonicity of α. If the system is safe ($h(z) \geq 0$), then the safety dynamics are free to evolve to become more or less safe, which allows the system to approach the boundary of the safe set if needed.

For continuous-time systems and control-affine dynamics, the following optimization problem will always be a quadratic program in u:

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & u^\top u \\ \text{subject to} \quad & L_f h + L_g h u \geq -\alpha(h(z)). \end{aligned} \tag{12}$$

If there are no constraints on the magnitude of u, then the above problem is always feasible and formally guarantees the safe evolution of the system.

C. Exponential Control Barrier Functions

The theory presented in the prior section relies on the ability of the control input to directly affect $\dot{h}(z)$. However, in the case when the relative degree of the CBF (with respect to the input u) is greater than two, i.e. only higher-order time derivatives of h depend on the input, (12) does not provide a sufficient framework for ensuring safety.

The *exponential control barrier function* (EBCF) was established in [12] as a method for enforcing safety even for systems with arbitrarily high relative degree. The findings are summarized here. First, for a system with relative degree r, define the following:

$$\eta_b(z) = \begin{bmatrix} h(z) \\ \dot{h}(z) \\ \vdots \\ h^{(r-1)}(z) \end{bmatrix} = \begin{bmatrix} h(z) \\ L_f h(z) \\ \vdots \\ L_f^{r-1} h(z) \end{bmatrix}. \tag{13}$$

Then, the following linear system can be constructed:

$$\begin{aligned} \dot{\eta}_b(z) &= F\eta_b(z) + G\mu, \\ h(z) &= C\eta_b(z), \end{aligned} \tag{14}$$

where

$$\begin{aligned} F &= \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix}, \\ C &= \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}. \end{aligned} \tag{15}$$

For the selection $\mu \geq -K_\alpha \eta_b(z)$, it is guaranteed that $h(z(t)) \geq Ce^{(F - GK_\alpha)t}\eta_b(z_0)$. Formally:

Definition 5 (Exponential Control Barrier Function)**.** For $\mathcal{C} \subset \mathcal{D} \subseteq \mathbb{R}^n$ defined as the 0-superlevel set of an r-times continuously differentiable function $h : \mathcal{D} \to \mathbb{R}$, h is an *exponential control barrier function* if there exists a row vector $K_\alpha \in \mathbb{R}^r$ such that for control-affine dynamics $\dot{z} = f(z) + g(z)u$,

$$\sup_u \left\{ L_f^r h(z) + L_g L_f^{r-1} h(z)u \geq -K_\alpha \eta_b(z) \right\} \tag{16}$$

$\forall x \in \text{Int}(\mathcal{C})$ ensures that $h(z(t)) \geq Ce^{(F - GK_\alpha)t}\eta_b(z_0)$ if $h(z_0) \geq 0$.

The main result requires some additional restrictions on the closed-loop poles of $F - GK_\alpha$. First, the poles must be real and strictly negative. Second, define the recursive family of functions

$$\begin{aligned} y_0 &= h(z) \\ y_1 &= \dot{y}_0 + \lambda_1 y_0 \\ &\vdots \\ y_i &= \dot{y}_{i-1} + \lambda_i y_{i-1}. \end{aligned} \tag{17}$$

Then, the following must be satisfied:

$$-\lambda_i(F - GK_\alpha) \geq -\frac{\dot{y}_{i-1}(z_0)}{y_{i-1}(z_0)}. \tag{18}$$

In practice, the following selection rule works well and allows tuning of the strictness of the ECBF:

$$\lambda_i = \min\left(\frac{\dot{y}_{i-1}(z_0)}{y_{i-1}(z_0)}, -c_i\right), \tag{19}$$

where $c_i > 0$ is an "alternate" gain for the i^{th} pole of $F - GK_\alpha$ and greater values yield greater responsiveness to CBF violations. However, choosing over-aggressive values for c_i may produce instability.

IV. Control for Collaborative Transport

Having laid the mathematical groundwork for general safety theory, the following section presents the control strategy for safe decentralized collaborative transport.

A. Nominal Control Strategy

In explicit-leader schemes, it is reasonable to assume that the leader directs the load according to some higher-level planning algorithm and the followers assist the leader by applying an assistive force. In this paper, the leader blindly uses proportional-derivative control laws to track some precomputed desired trajectory without regard for safety. The desired net applied force is therefore computed as

$$F_d = -k_p(p - p_d) - k_d(\dot{p} - \dot{p_d}), \tag{20}$$

where p is the vector of the load's position states $\begin{bmatrix} x & y & \theta \end{bmatrix}^\top$; $\dot{p}$ is the vector of the load's velocity states $\begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^\top$; and p_d, $\dot{p_d}$ are the desired position and velocity states respectively.

The leader also tracks a desired orientation at which the vector from the load COM to the leader r_l is tangent to the trajectory. This yields the following expression for the desired moment:

$$M_d = -k_\theta \text{sign}(\gamma)\,|\beta| - k_\omega \omega, \tag{21}$$

where

$$\gamma = \det\left(\begin{bmatrix} \frac{\dot{p_d}}{\|\dot{p_d}\|} & \frac{\dot{r_l}}{\|\dot{r_l}\|} \end{bmatrix}\right), \tag{22}$$

$$|\beta| = \cos^{-1}\left(\frac{\dot{p_d}}{\|\dot{p_d}\|} \cdot \frac{r_l}{\|r_l\|}\right). \tag{23}$$

Here, the value $|\beta|$ is the magnitude of the heading error of the load, while γ indicates the sign of β.

The following optimization problem computes a feasible control input for the leader $u_l = \begin{bmatrix} F^x_{l,1} & F^x_{l,2} & F^y_{l,1} & F^y_{l,2} \end{bmatrix}^\top$ that satisfies F_d and M_d so long as the two points of force application are not located identically in the plane:

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & \|u\|_2^2 \\ \text{s.t.} \quad & \begin{bmatrix} F^x_d \\ F^y_d \\ M_d \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ -r^y_{l,1} & -r^y_{l,2} & r^x_{l,1} & r^x_{l,2} \end{bmatrix} u \end{aligned} \tag{24}$$

The followers nominally apply the control law from [10] to assist the leader by pushing the load in the direction it is already traveling:

$$F_d = k_a \frac{v}{\|v\|}, \tag{25}$$

$$M_d = 0, \tag{26}$$

where v is the current load positional velocity $\begin{bmatrix} \dot{p^x} & \dot{p^y} \end{bmatrix}^\top$. Note that the desired net moment is zero, since the goal is for the leader to direct the load's travel. The corresponding nominal follower inputs are similarly computed using (24).

B. Safety Constraints

The safety concerns studied in this paper are collision avoidance and limits on the load's linear and angular velocities. For a desired lower bound on the distance to obstacles δ_p and desired upper bounds on the linear and angular velocity δ_v, δ_ω, the following barrier functions can be defined:

$$h_d = \|d\|^2 - \delta_p^2, \tag{27}$$

$$h_v = \delta_v^2 - \|v\|^2, \tag{28}$$

$$h_\omega = \delta_\omega^2 - \|\omega\|^2. \tag{29}$$

Geometrically, $\{z \mid h_d(z) < 0\}$ can be interpreted as the set of states occupying a ball of radius δ_p about the obstacle, which forms the complement of the safe set for collisions. In contrast, the balls with radii δ_v, δ_ω are the safe sets for linear and angular velocities and their complements are dangerous.

C. Decentralized Dynamical Prediction

Having established a viable nominal strategy and associated safety constraints for the task at hand, a particular robot i in the team must predict the actions of the other team members in order to compute an appropriate control action. The joint dynamics from Section II-B can be expressed in the following control-affine form:

$$\begin{aligned} \dot{z} &= f(z) + g_1(z)u_1 + \cdots + g_i(z)u_i + \ldots g_n(z)u_n \\ &= f(z) + \sum_{j \neq i} g_j(z)u_j + g_i(z)u_i. \end{aligned} \tag{30}$$

Using this decomposition, the *predictive dynamics* for robot i can be written

$$\hat{\dot{z}}_i = \hat{f}_i(z, \hat{u}_{j\neq i}) + \hat{g}_i(z)u_i, \tag{31}$$

where

$$\hat{f}_i(z, \hat{u}_{j\neq i}) = f(z) + \sum_{j \neq i} g_j(z)\hat{u}_j, \tag{32}$$

$$\hat{g}_i(z) = g_i(z), \tag{33}$$

and predicted values are designated with a hat. If the predictive load dynamics are accurate, then a safe control action from the perspective of robot i can be derived using the methods presented in Sections III-B and III-C.

This paper considers the following crude zeroth-order prediction model, which assumes that each agent will apply the same forces and moments as in the previous time step:

$$\hat{F}_j[k] = F_j[k-1], \quad \hat{M}_j[k] = M_j[k-1], \ \forall j \neq i. \tag{34}$$

Remark 1. Though the dynamics and barrier function theory are derived in continuous time, the prediction model in (34) is naturally conceived of in discrete time, since in continuous time, there is not a well-defined notion of the "previous" inputs. In practice, with a sufficiently fine control frequency, this pseudo-continuous strategy performs adequately and is easily implemented on digital systems. Additionally, the designer is free to choose a continuous-time prediction law, as long as it is expressible in closed form. The remainder of this section uses discrete-time notation, but one can readily convert between discrete and continuous time using the relation $t = k\Delta t$, where $1/\Delta t$ is the control frequency.

By manipulating the dynamical model presented in Section II-B using a similar decomposition as in (30), one obtains the following expressions at any given time step k:

$$\sum_{j\neq i} F_j^x = m\ddot{x} - (F_{i,1}^x + F_{i,2}^x), \tag{35}$$

$$\sum_{j\neq i} F_j^y = m\ddot{y} - (F_{i,1}^y + F_{i,2}^y), \tag{36}$$

$$\begin{aligned}\sum_{j\neq i} M_j = I\ddot{\theta} - (&-r_{i,1}^y F_{i,1}^x - r_{i,2}^y F_{i,2}^x \\ &+ r_{i,1}^x F_{i,1}^y + r_{i,2}^x F_{i,2}^y).\end{aligned} \tag{37}$$

The prediction model then gives

$$\begin{aligned}\sum_{j\neq i} \hat{F}_j^x[k] &= m\ddot{x}[k-1] - (F_{i,1}^x[k-1] + F_{i,2}^x[k-1]), \\ \sum_{j\neq i} \hat{F}_j^y[k] &= m\ddot{y}[k-1] - (F_{i,1}^y[k-1] + F_{i,2}^y[k-1]), \\ \sum_{j\neq i} \hat{M}_j[k] &= I\ddot{\theta}[k-1] - (r_{i,1}[k-1] \times F_{i,1}[k-1] \\ &\qquad + r_{i,2}[k-1] \times F_{i,2}[k-1]).\end{aligned} \tag{38}$$

Therefore, if the input of robot i is written

$$u_i = \begin{bmatrix} F_{i,1}^x & F_{i,2}^x & F_{i,1}^y & F_{i,2}^y \end{bmatrix}^\top, \tag{39}$$

the predictive dynamics at step k can be expressed as

$$\hat{f}_i = \begin{bmatrix} z_2[k] \\ \frac{1}{m}\sum_{j\neq i} \hat{F}_j^x[k] \\ z_4[k] \\ \frac{1}{m}\sum_{j\neq i} \hat{F}_j^y[k] \\ z_6[k] \\ \frac{1}{I}\sum_{j\neq i} \hat{M}_j[k] \end{bmatrix}, \tag{40}$$

$$\hat{g}_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ -r_{i,1}^y[k] & -r_{i,2}^y[k] & r_{i,1}^x[k] & r_{i,2}^x[k] \end{bmatrix}. \tag{41}$$

To compute the aggregate force and moment predictions, robot i requires knowledge of the linear and angular accelerations of the load's center of mass. When these values cannot be directly measured, they can be estimated using, for example, first-order Euler estimates and velocity data (which may in turn be estimated from position data).

D. Optimal Evasive Maneuvers

With the nominal controller and predictive dynamics established, a decentralized and minimally-invasive control input can be computed for robot i. Let $\mathcal{L}_i$ be the set of indices corresponding to detected obstacles in the environment. Further, let $\mathcal{V}_i$ be the set of indices corresponding to the points on the load which robot i tries to protect from collision. In other words, if $j \in \mathcal{L}_i$ and $k \in \mathcal{V}_i$, then h_{ijk} is the CBF relating the distance between the j^{th} obstacle that robot i observes and the k^{th} protected point on the load. Borrowing the notation from [12], let

$$\eta_{ijk} = \begin{bmatrix} h_{ijk} \\ \dot{h}_{ijk} \end{bmatrix}. \tag{42}$$

With this notation, the solution to the following quadratic program yields a minimally-invasive safe control input for robot i with respect to the predictive dynamics:

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & \|u - u_{i,\text{nom}}\|^2 \\ \text{subject to} \quad & L_{\hat{f}_i} h_v + L_{\hat{g}_i} h_v u \geq -\alpha_v(h_v) \\ & L_{\hat{f}_i} h_\omega + L_{\hat{g}_i} h_\omega u \geq -\alpha_\omega(h_\omega) \\ & L_{\hat{f}_i}^2 h_{ijk} + L_{\hat{g}_i} L_{\hat{f}_i} h_{ijk} u \geq \\ & - K_{ijk}\eta_{ijk}, \ \forall j \in \mathcal{L}_i, \ \forall k \in \mathcal{V}_i. \end{aligned} \tag{43}$$

Using the design protocol presented in Section III-C for designing ECBFs, one can first select the desired eigenvalues of the closed-loop linear system (14) and then use standard pole placement techniques to select the feedback gains K_{ijk} corresponding to barrier function h_{ijk}. Let λ_1, λ_2 be the two closed-loop poles. Then, applying (19) gives

$$\begin{aligned} \lambda_1 &= \min\left(\frac{\dot{h}_{ijk}(z_0)}{h_{ijk}(z_0)}, -c_1\right), \\ \lambda_2 &= \min\left(\frac{\ddot{h}_{ijk}(z_0, u_{i,\text{nom}})}{\dot{h}_{ijk}(z_0)}, -c_2\right), \end{aligned} \tag{44}$$

where $c_1, c_2 > 0$ are tunable constants and z_0 is the value of the load state when CBF h_{ijk} is instantiated. For example, if any obstacle leaves the view of robot i and is detected again at a later time, then the pole placement procedure must be repeated since the barrier function will have been inactive for some period of time and must be re-instantiated.

Remark 2. Recall that because h_{ijk} will have relative degree 2, h_{ijk} and $\dot{h}_{ijk}$ do not depend on the input u_i. However, since the selection criteria for λ_2 will in general depend on the highest-order time derivative of h_{ijk}—which in turn depends on the input—a sound choice is to compute the second derivative as a function of the nominal input, since one would hope for the nominal input to be close in magnitude to any input resulting in evasive maneuvers.

Remark 3. The exact class-$\mathcal{K}_\infty$ functions α_v and α_ω as well as the values of feedback gain matrix K_{ijk} can be selected through empirical testing or simulation. Depending on the magnitude of prediction, discretization, or other modeling error, parameter values which may be appropriate for some scenarios may be unsafe or destabilizing in others. Some intuition can be borrowed from classical gain tuning in this regard. For example, if α_v or α_ω are chosen with large growth rates, then the convergence back to safe linear and angular velocities will be rapid, but if modeling errors result in a large violation of either constraint, then the resulting input may induce oscillations and become unbounded, as in the use of overly-aggressive proportional gains. In practice, selecting linear functions for $\alpha(\cdot)$ seems to work well.

E. Dynamic Trust

The simplicity of the prediction model may yield poor performance when many robots simultaneously make inaccurate predictions. Consider a scenario where the load approaches an obstacle and all followers observe the obstacle at the same instant. Based on the zeroth-order prediction dynamics, no robot will expect any other to react to the obstacle, and it is likely that the aggregate action will result in an overreaction that can destabilize the system.

Define $\psi_i(h)$ as the *trust factor* that robot i has in the team to enforce barrier function h. Further, if robot i is actively enforcing m_i different barrier functions, define the *aggregate trust factor* of robot i as

$$\psi_i = \frac{1}{m_i}\sum_{j=1}^{m_i}\psi_i(h_j). \tag{45}$$

A heuristically-motivated strategy for acting appropriately is dividing the safe input from the optimization problem in (43) by the aggregate trust factor. Consider some value of a particular trust factor at step k, $\psi_i(h)[k]$. Let the *predictive barrier value* for the CBF value at step k be one of

$$\hat{h}[k] = h[k-1] + (\Delta t)\dot{h}[k-1], \tag{46}$$

$$\hat{h}[k] = h[k-1] + (\Delta t)\dot{h}[k-1] + \frac{1}{2}(\Delta t)^2\ddot{h}[k-1], \tag{47}$$

depending on whether h has relative degree one or two, where the highest-order time derivative of h should depend on the input $u_i[k-1]$.

Then, an update rule for the trust factor can be written:

$$\psi_i(h)[k+1] = \left(1 + \tau\left(\hat{h}[k] - h[k]\right)\right)\cdot\psi_i(h)[k], \tag{48}$$

where τ is some function of the predictive barrier value error that satisfies the property that it is non-decreasing over the domain. Further, it should be negative when the error is positive and positive when the error is negative, which translates to an increase in trust when the system evolves to be safer than predicted and vice versa.

This trust update is performed over all active barrier functions, which are then used to compute the aggregate trust at this step $\psi_i[k]$. Finally, if $\bar{u}_i$ is the result of solving (43), then let the input of robot i at this step be

$$u_i[k] = \frac{\bar{u}_i[k]}{\psi_i[k]}. \tag{49}$$

An optimistic initial value of the trust factor is $n-1$, which implies that every robot believes that the control effort for enforcing all barrier functions will be uniform for all followers on the team. The selection of this value in general, of course, depends highly on the problem setup and is left to the designer.

Remark 4. The intuition behind the trust factor is as follows: if robot i trusts the team more, then its individual control action required to enforce safety should be smaller, since other teammates will react. Conversely, lower trust means robot i will apply a larger input to yield stronger evasive actions. Naturally, trust should be related to the relative sensor coverage of the team members: coverage overlaps should increase trust while compartmentalized coverage should decrease it. Further, the trust factor should be dynamic, since factors like visual occlusions, noise, and more may cause rapid changes to a robot's awareness, and therefore, the trust dynamics, which yields an update law resembling integral control. While the presented heuristic works well in practice, many choices of τ result in unstable outcomes. A rigorous analysis of the trust factor is left for future work.

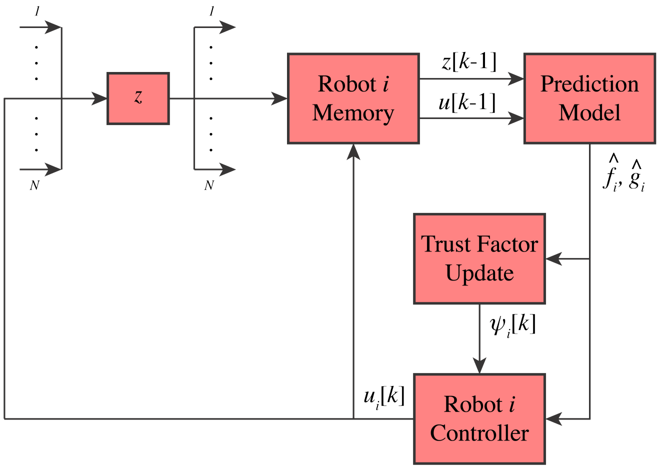

Fig. 2: A block diagram detailing the zeroth-order prediction framework as it relates to the controller. Note the completely decentralized flow of information: robot i only requires memory of its own inputs and observations.

V. Simulations

A. General Results and a Representative Example

A series of simulations were conducted to assess the viability of the proposed method. Generally, the results were favorable and the system was shown to safely navigate around obstacles placed directly in the path of the load as well as moving obstacles. The controller was able to operate under a wide range of control frequencies, design parameter choices, and number of robots on the team, though there is a floor for how low control frequencies may become. Throughout these simulations, there were no limits on the control inputs, as this may yield infeasible solutions to (43).

A representative example for demonstration is depicted in Figure 3. The system consists of a rectangular load, one leader, and three followers. The system is initialized in the center of the environment and tracks a circular trajectory while avoiding static and moving obstacles. In this example, all gains were well-tuned (c_1, c_2 for collision avoidance, α_v, α_ω for linear and angular velocity constraints) and the robot successfully maintains a wide berth around obstacles while regulating all three CBFs reasonably, with only mild CBF violations and fast convergence back to safety.

It is evident from Figures 3(d)-(f) that safety violations often occur hand-in-hand. This is an intuitive result given the

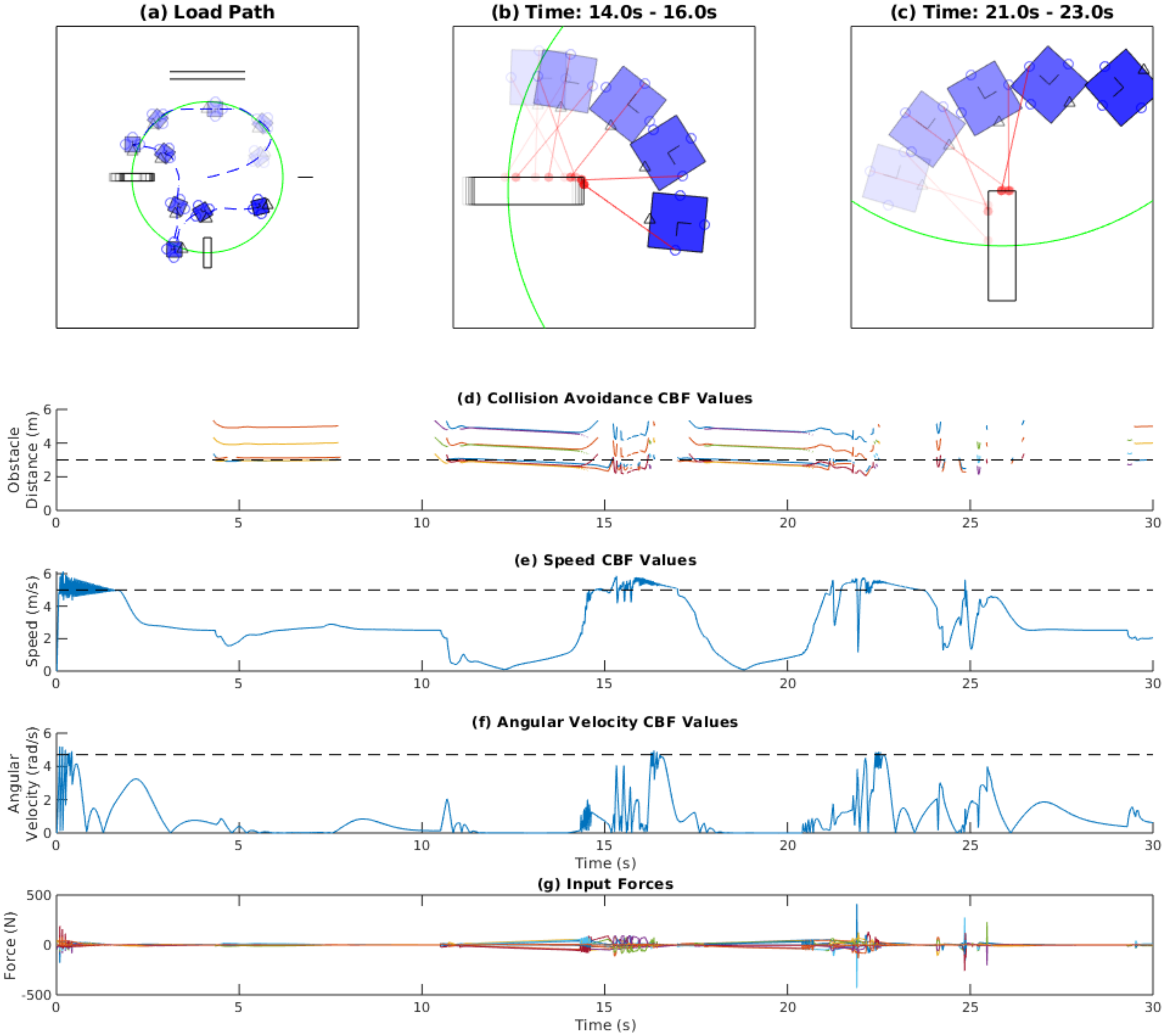

Fig. 3: Simulation results from a well-tuned 30 second run. Obstacles appear as black lines while the load is the blue square. Lasers are shown in red and only lasers with detections are shown. The leader robot is denoted as a triangle and the followers circles. The desired trajectory is shown in green. (a)-(c) show the overall path of the load, with (b) and (c) showing magnified views of the load near obstacles. (d)-(f) show the values of the various barrier functions over the run. In (d), the empty sections correspond to inactive barriers (no laser detections). The safety margins are denoted with the black dotted line. (g) shows all input forces over the run. The peak force is 412N.

nature of the desired safety constraints: evasive maneuvers typically involve high-speed movements in order to avoid collisions, especially when the obstacle is moving as in the one shown in 3(b). While there are naturally tradeoffs in enforcing each barrier function, these results suggest there may exist some tuned parameter values that are optimal for multiple safety objectives depending on the choice of CBFs. However, during evasive maneuvers, forces may increase significantly (Figure 3(g)), and if input bounds are enforced, infeasibilities may result or large safety violations may occur.

B. Failure Modes

Many common failure modes were also observed in simulated trials. Two such cases are depicted in Figure 4. In Figure 4(a), the inputs become unbounded with aggressively-tuned gains and no trust factor, destabilizing the system within 0.2 seconds. Similar and often oscillatory failure modes were observed even when trust factors were activated given sufficiently aggressive gains. Additionally, high prediction error resulting from a low control frequency yielded similar behavior.

Figure 4(b) demonstrates the relationship between aggressive regulation of linear and angular velocity constraints when the collision avoidance gains are poorly-tuned. Low

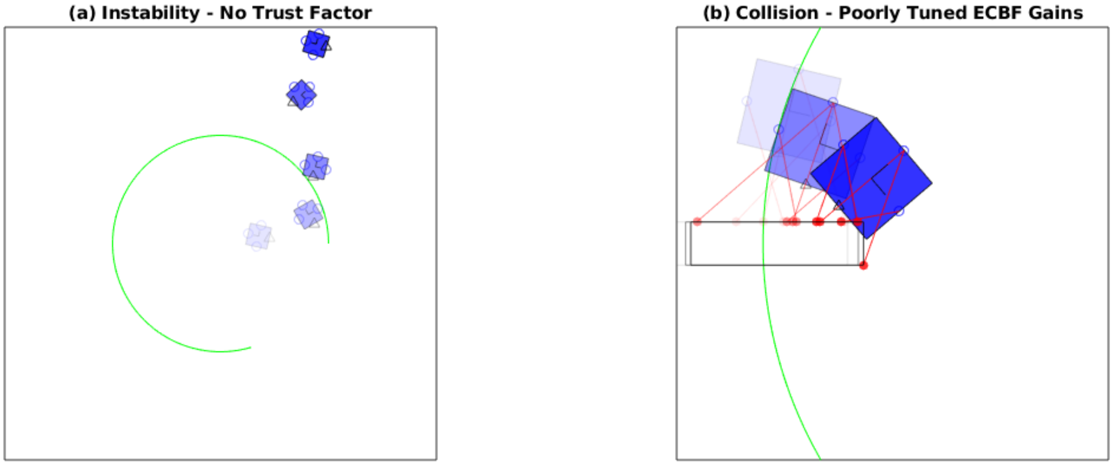
(a) Instability - No Trust Factor
(b) Collision - Poorly Tuned ECBF Gains

Fig. 4: Two cases of poor gain tuning. (a) shows instability resulting from the lack of trust factors, yielding unbounded inputs. Overly-aggressive class-$\mathcal{K}$ functions α_v and α_ω often induce similar behavior via oscillations. (b) shows an overly-relaxed selection for c_1, c_2 when tuning the feedback gain matrix for the collision avoidance constraint. Overly-timid tuning causes under-reactions near obstacles.

values of c_1, c_2 in (44) results in poor low responsiveness to obstacles, whereas higher values yielded "rigid" behavior with greater responsiveness at the cost of more severe linear and angular velocity safety violations. On the other hand, over-responsiveness was also observed to result in destabilizing oscillatory behavior.

VI. Conclusion

This paper offers a prediction-based framework for safe decentralized collaborative transport with an explicit team leader who is unaware of the safety constraints of other team members. Many qualitative insights into the team dynamics and behavior were provided in addition to a robustness analysis that provides criteria for the selection of design parameters. Simulations of the proposed method demonstrated success, with common failure modes being identified.

Some possible avenues for future research include

- more accurate prediction models derived from game theory or learning-based methods,
- rigorous analyses of the trust factor using concepts from adaptive control theory,
- the usage of discrete-time CBFs [13] (this paper did not explore discrete-time CBFs because guarantees on the convexity of the optimal control problem would vanish, and in general some scheme such as linearization of the CBF must take place. Even with linear CBF dynamics, the best one can hope for is a QCQP),
- the use of delay differential equation (DDE) theory to analyze the instability or oscillations induced by the delayed zeroth-order prediction model,
- improved perception methods that include semantic representations of the environment, and
- the establishment of more stringent mathematical characterizations of the method's robustness.

References

[1] J. Cortes and M. Egerstedt, "Coordinated control of multi-robot systems: A survey," *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 6, pp. 495–503, 2017.

[2] P. Ogren, M. Egerstedt, and X. Hu, "A control lyapunov function approach to multi-agent coordination," *Proceedings of the 40th IEEE Conference on Decision and Control*, pp. 1150–1155, December 2001.

[3] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, June 2019, pp. 3420–3431.

[4] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, June 2017.

[5] G. Notomista, X. Cai, J. Yamauchi, and M. Egerstedt, "Passivity-based decentralized control of multi-robot systems with delays using control barrier functions," *2019 International Symposium on Multi-Robot and Multi-Agent Systems*, September 2019.

[6] D. Panagou, D. M. Stipanović, and P. G. Voulgaris, "Distributed coordination control for multi-robot networks using lyapunov-like barrier functions," *IEEE Transactions on Automatic Control*, vol. 61, no. 3, pp. 617–632, March 2016.

[7] D. Rus, B. Donald, and J. Jennings, "Moving furniture with teams of autonomous robots," in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 1, Aug 1995, pp. 235–242 vol.1.

[8] L. Rozo, D. Bruno, S. Calinon, and D. G. Caldwell, "Learning optimal controllers in human-robot cooperative transportation tasks with position and force constraints," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 1024–1030.

[9] M. D. Kennedy, "Modeling and control for robotic assistants: Single and multi-robot manipulation," Ph.D. dissertation, University of Pennsylvania, 2019.

[10] Z. Wang and M. Schwager, "Force-amplifying N-robot transport system (force-ANTS) for cooperative planar manipulation without communication," *International Journal of Robotics Research*, vol. 35, no. 13, pp. 1564–1586, 2016.

[11] P. Culbertson and M. Schwager, "Decentralized adaptive control for collaborative manipulation," in *Proc. of the International Conference on Robotics and Automation (ICRA)*, 5 2018, pp. 278–285.

[12] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*, July 2016, pp. 322–328.

[13] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation," in *Robotics: Science and Systems*, 2017.