Decentralized Safety for Aggressively Maneuvering Multi-Robot Interactions

Phillip M. Rivera-Ortiz,[1,2] Marin Kobilarov[2]

***Abstract*— This work provides a decentralized approach to safety by combining tools from control barrier functions (CBF) and nonlinear model predictive control (NMPC). It is shown how leveraging backup safety controllers allows for the robust implementation of CBF over the NMPC computation horizon, ensuring safety in nonlinear systems with actuation constraints. A leader-follower approach to control barrier functions (LF-CBF) enforcement will be introduced as a strategy to enable a robot leader, in a multi-robot interactions, to complete its task in minimum time, hence aggressively maneuvering. An algorithmic implementation of the proposed solution is provided and safety is verified via simulation.**

I. Introduction

The area of safety-critical systems has been extensively researched in recent years. The main focus of this work is on single-robot or multi-robot teams that operate in contested environments with time-critical objectives. Examples include search and rescue operations [1] where immediate assistance might be needed, multi-player capture problems as in pursuit-evasion games [2], [3], or mixed autonomous and human-operated robot scenarios where cooperation cannot be expected to preserve safety from other robots in the environment as shown in Fig. 1. The desired functionality is to enable safety in multi-robot environments, while the robots execute aggressive trajectories. In Fig. 1 the provided approach is demonstrated in an autonomous driving scenario where a human-operated vehicle (blue) invades the lane of the autonomous vehicle (orange). The autonomous vehicle preserves safety by first deaccelerating, then changes lanes to ensure the desired set speed is maintained.

This work leverages recent contributions in Control Barrier Functions (CBF) to safety in a minimally invasive way. CBF research often describes a system objective as the preservation of two properties, a *liveliness* property encoding goal satisfaction, and a *safety* property ensuring forward invariance of a safe set [4], [5]. A common way of selecting input actions for robotic systems in dynamically changing environments is through Nonlinear Model Predictive Control (NMPC) [6]. NMPC offers a framework under which the desired liveliness property can be attained in real-time. Thus, the proposed approach for this work is to combine elements of CBF with NMPC in the context of multi-robot interactions.

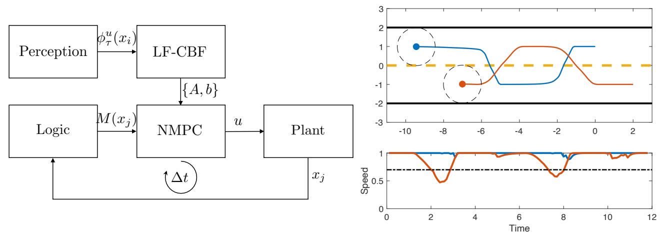


Fig. 1. Autonomous driving example. Blue robot-i is selected as the leader in the LF-CBF allowing the orange robot-j to preserve safety. The block diagram shows the mechanization of the proposed approach for autonomous driving where robot-j tries to preserve a set speed while ensuring safety. The Logic block induces a lane change when the speed falls below a certain threshold, provided by the dashed line in the speed subplot. The perception block provides the flow of robot-i.

The addressed technical challenge is scalability. Even if objective of every robot is known to the entire team, simultaneous computation of all optimal trajectories might be computationally prohibitive, especially in real-time scenarios with complex nonlinear plants. Regarding safety, this implies it is not possible to verify the safety condition at all times. Thus, the focus is on NMPC problems whose multi-robot coupling enters as a constraint to ensure safety. Through the use of CBFs, decentralized constraint enforcement will be shown possible.

The contributions of the proposed approach are twofold:

- Temporally local approach to safety for optimal trajectories of actuator constrained nonlinear plants,
- Temporally global assurance of individual task satisfaction and trajectory feasibility for multi-robot teams.

The proposed approach leverages work on active set invariance for CBF to provide temporally local input constraints to NMPC algorithms, thus combining safety and local trajectory optimality. This contrasts with the work that focuses on CFBs in combination with Control Lyapunov Functions (CLF) because there is no notion of optimality or organic way of embedding final time constraints [7]. It also contrasts with the use of CBFs as a safety filter after a control action is selected resulting in non-optimal trajectories [8]. The advantage of the developed approach is that safety can be readily incorporated in the NMPC problem by the addition of affine constraints to the input at the current time through the evaluation of safe backup strategies. These affine constraints can be readily decentralized as shown in this work.

This paper is organized as follows. Section II provides the

*Work sponsored in part by the Department of the Navy, Office of Naval Research under ONR award number: N00014-21-1-2415, and by a Johns Hopkins University Applied Research Laboratory Internal Research and Development grant

[1]JHU-Applied Physics Laboratory, Laurel, MD 20723, USA phillip.rivera@jhuapl.edu

[2]Department of Mechanical Engineering, John Hopkins University, Baltimore, MD 21218, USA

problem description, Section III contains the mathematical preliminaries expanded on for this work, and Section IV provides main results, including an algorithmic implementation. Simulation examples and analysis are provided in Section V and Section VI respectively. Section VII provides the conclusion.

II. Problem Description

The objective is the design of locally optimal trajectories that ensure objective satisfaction while guaranteeing safety in multi-robot tasks. To that end, consider a team of N robots with Lipschitz continuous nonlinear dynamics

$$\begin{aligned} \dot{x}_i &= f(x_i) + g(x_i)u \\ u &\in \mathcal{U}_i \subseteq \mathbb{R}^m, \quad i \in \mathcal{I} = \{1, \ldots, N\} \end{aligned} \tag{1}$$

where $x_i \in \mathbb{R}^n$, and $\mathcal{U}_i$ encodes actuator limits of robot-i. Safety of the dynamical system in (1) can be encoded as the satisfaction of internal robot, or robot to environment inequality $h_i(x_i) \geq 0, \ \forall i \in \mathcal{I}$, and robot-team-wise inequality $h_{i,j}(x_i, x_j) \geq 0 \ \forall i \in \mathcal{I}, \forall j \neq i$. Thus, defining the super level set

$$\begin{aligned} \mathcal{C} = \{(x_i, x_j) \, | h_i(x_i) \geq 0, \ \forall i \in \mathcal{I}, \\ h_{i,j}(x_i, x_j) \geq 0, \ \forall i \in \mathcal{I}, \forall j \neq i\}, \end{aligned} \tag{2}$$

safety is guaranteed by ensuring forward invariance of the safe set $\mathcal{C}$, i.e., if the state start in the set, it remains in the set for a prescribed finite time.

Optimal trajectories of system (1) are obtained through the Optimal Control Problem (OCP)

$$(u_i^\star, T_i^\star) = \underset{u_i, t_f}{\arg\min} \ M(x_i(t_f)) + \int_{t_0}^{t_f} L(x_i, u_i, t) dt \tag{3}$$

subject to the system dynamics in (1). In this OCP, the Mayer Term $M(x)$ is used to enforce task satisfaction. Both the Mayer term and the Lagrange term $L(x, u, t)$ are task specific.

The main challenges in guaranteeing safety for the multi-robot interactions arise from the time-varying nature of the environment, and the fact that objective satisfaction and safety may be at odds. By the time-varying nature of the environment, we imply that it might not be known to robot-i what the task of robot-j is ahead of time. Therefore it is not possible to evaluate $h_{i,j}(x_i, x_j)$ at all times. Furthermore, even if robot-i is aware of robot-j's task, it might be that $h(x_i^\star, x_j^\star) \leq 0$. Thus, we seek to develop a decentralized trajectory optimization framework that is flexible enough to adapt to time-varying changes in the environment and ensures the feasibility of the objective satisfaction for all agents in the presence of actuator limits.

This work assumes the multi-robot systems have model-based knowledge of all the, potentially heterogeneous, constituents. Assurance of safe task satisfaction will be developed imposing a leader-follower topology on safety, in which a selected leader is assumed to share a parameterization of its optimal input sequence. Furthermore, the only additional information needed for decentralized safety of the multi-robot team is knowledge of the neighbors backup strategy as

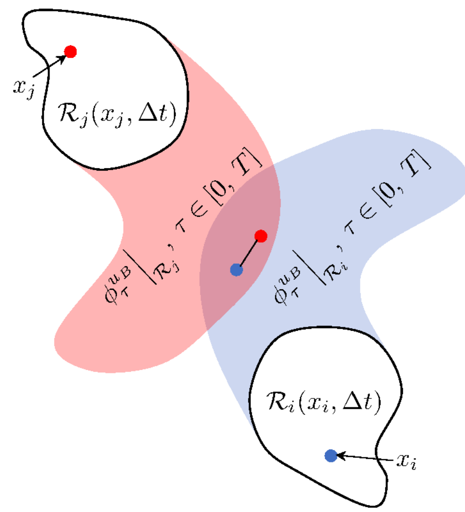


Fig. 2. Safety is evaluated over the flow of the two systems (i, j) starting from their individual reachable sets under time Δt. An example of maintaining a safety distance is provided by the two colored circles connected by the black line, which is considered safe if $h_{i,j}(x_i, x_j) \geq 0$.

introduced in [8], assuming instantaneous state information can be measured.

III. NMPC and CBF Preliminaries

A. Nonlinear MPC

Direct Transcription provides a commonly used approach to solving OCP of nonlinear systems [9]. Each robot-i can formulate the trajectory optimization problem as

$$\min_{u_i \in \mathcal{U}_i, t_f} \quad M(x_i(t_f)) + \int_{t_0}^{t_f} L(x_i, u_i, t) dt \tag{4}$$

$$\text{s.t.} \quad \dot{x}_i = f(x_i) + g(x_i) u_i,$$

$$h_i(x_i(t)) \geq 0, \ \forall t \in [t_0, t_f] \tag{4.a}$$

$$h_{i,j}(x_i(t), x_j(t)) \geq 0, \ \forall t \in [t_0, t_f]. \tag{4.b}$$

Note that problem (4)-(4.b) is considered a centralized trajectory optimization algorithm because the trajectories of the multi-robot team are needed to evaluate (4.b). The considered NMPC solves an OPC and executes the input commands over a determined time interval Δt, then recomputes the optimal trajectory, discarding the remaining input trajectory. This Δt accounts for the OCP computation time. These algorithms can be solved through Sequential Quadratic Programming, whose convergence is affected by non-convex state feasible sets [9], [10].

Thus, the proposed approach looks into developing a decentralized NMPC algorithm by converting the constrains (4.a)-(4.b) into an affine constraint *only on the input* over a fixed time interval, e.g., $A_i u_i \leq b_i \ \forall t \in [t_0, t_0 + \Delta t]$ to ensure safety. The general approach will be to construct CBFs considering the individual robot state flows using the selected backup controller as shown in Fig. 2. To ensure safety over the NMPC horizon Δt, work from CBF over sampled data systems [11] will be leveraged. It will be shown that feasibility of the NMPC problem is ensured by the existence of a safe trajectory from a backup controller.

B. Main Results in Control Barrier Functions

Let us now recall some relevant results from CBFs that will enable the extension to the NMPC context.

Definition 1 (Control Barrier Function [4]): Let $\mathcal{C} \subset D \subset \mathbb{R}^n$ be the superlevel set of a continously differentiable function $h : D \to \mathbb{R}$, then h is a control barrier function (CBF) if there exists an extended class-$\mathcal{K}$ function α such that for control system (1):

$$\sup_{u \in \mathcal{U}} \left[\dot{h}(x, u)\right] + \alpha(h(x)) \geq 0 \tag{5}$$

for all $x \in D$.

From this definition, safety is guaranteed by selecting inputs from the set that render $\mathcal{C}$ safe

$$K_{cbf}(x) = \{u \in \mathcal{U} \mid \dot{h}(x, u) + \alpha(h(x)) \geq 0\}. \tag{6}$$

Note that this approach provides safety conditions point-wise for every $x \in D$. A known issue with extending this to systems with actuation limits is that when considering trajectories of system (1), nothing prevents the safe input set (6) from becoming empty. Approaches have been developed for jointly optimizing trajectories and parameterized CBFs to ensure the feasibility of the set [12], [13]. Our approach instead builds on the work of Active Set Invariance [14], which relies on evaluating the flow of the system using pre-designed backup safety controllers, and ensuring the system remains within reach of the backup controller.

Before presenting results on the active set invariance, one of the main contributions on CBFs, leveraged in this work, is provided by the following theorem.

Theorem 1 (Necessity for Safety[4]): Let $\mathcal{C}$ be a compact set that is the superlevel set of a continously differentiable function $h : D \to \mathbb{R}$ with the property that $\frac{\partial h}{\partial x}(x) \neq 0$ for all $x \in \partial\mathcal{C}$. If there exists a control law $u = k(x)$ that renders $\mathcal{C}$ safe, then $h : \mathcal{C} \to \mathbb{R}$ is a CBF on $\mathcal{C}$.

C. Active Set Invariance

To actively ensure the feasability of the CBF condition, denoted by $K_{cbf}(x) \neq \emptyset$ in (6), the use of safety backup controllers as suggested in [8], [11], [14] will be leveraged. The main idea is as follows. If we can ensure that over a fixed interval of time T, the dynamical system is backward reachable from a smaller, but known, safe set using the backup controller $u_B(x)$, then conditions for forward invariance can be obtained relative to a single safe trajectory of the system.

Let us define $\mathcal{B}(x) = \{x \in \mathbb{R}^n \mid h^B(x) \geq 0\}$, where $\mathcal{B} \subseteq \mathcal{C}$ is a smaller control invariant set for which a backup controller $u_B(x)$ can guarantee forward invariance. Furthermore, let us denote h^C as the safety conditions in (2). One can obtain a new CBF, $h^S(x)$ with $\mathcal{S}$ as a control invariant set, such that $\mathcal{B} \subseteq \mathcal{S} \subseteq \mathcal{C}$, from

$$h(t) = \min\left\{\min_{t' \in [t,\, t+T]} h^C(\phi^{u_B}_{t'-t}(x_0)), h^B(\phi^{u_B}_T(x_0))\right\}. \tag{7}$$

A sufficient condition for enforcing $\dot{h} + \alpha(h) \geq 0$ in (7) is given by [14]

$$\begin{aligned} \nabla h^B\left(\phi^{u_B}_T(x_0)\right)\left(\nabla\phi^{u_B}_T(x_0)\tilde{f}(x_0, u)\right)& \\ +\alpha\left(h^B\left(\phi^{u_B}_T(x_0)\right)\right) &\geq 0 \\ \nabla h^C\left(\phi^{u_B}_\tau(x_0)\right)\left(\nabla\phi^{u_B}_\tau(x_0)\tilde{f}(x_0, u) + \tfrac{\partial}{\partial t}\phi^{u_B}_\tau(x_0)\right)& \\ +\alpha\left(h^C\left(\phi^{u_B}_\tau(x_0)\right)\right) &\geq 0 \end{aligned} \tag{8}$$

for $\tau \triangleq t' - t$ such that $\tau \in [0, T]$, where $\tilde{f}(x_0, u) \triangleq f(x_0) + g(x_0)u$, and feedback control law $u_B(x)$ is fixed. The expression $\phi^{u_B}_T(x_0)$ corresponds to the flow of the system for a time period T, under the fixed control law $u_B(x)$ with initial conditions x_0. Note that $\frac{\partial}{\partial\tau}(\cdot) = -\frac{\partial}{\partial t}(\cdot)$. The Jacobian $\nabla(\phi^u_T(x_0))$ can be obtained by the forward integration of $\dot{Q} = \nabla(f(x) + g(x)u(x))Q$ where setting $x(0) = x_0$ and $Q(0) = I$, yields $Q(T) = \nabla\phi^u_T(x_0)$.

D. CBFs for Sampled Data Systems

The mechanization of the NMPC requires a computation time $\Delta t > 0$. To ensure safety over the computation interval, considerations from sampled-data systems when applying CBF as described in [11] will be leveraged. The main idea is that safety verification in the context of a delayed input can be verified at a finite set of reachable states by the system. In this work, a continous form of the safety backup controller is considered. Thus $h^S(x)$ in (8) can be enforced from

$$\begin{aligned} \nabla h^B\left(\phi^{u_B}_T|_{\mathbf{x_0}}\right)\left(\nabla\phi^{u_B}_T|_{\mathbf{x_0}}\tilde{f}(\mathbf{x_0}, u)\right)& \\ +\alpha\left(h^B\left(\phi^{u_B}_T|_{\mathbf{x_0}}\right)\right) &\geq 0 \\ \nabla h^C\left(\phi^{u_B}_\tau|_{\mathbf{x_0}}\right)\left(\nabla\phi^{u_B}_\tau|_{\mathbf{x_0}}\tilde{f}(\mathbf{x_0}, u) + \tfrac{\partial}{\partial t}\phi^{u_B}_\tau|_{\mathbf{x_0}}\right)& \\ +\alpha\left(h^C\left(\phi^{u_B}_\tau|_{\mathbf{x_0}}\right)\right) &\geq 0 \end{aligned} \tag{9}$$

$\forall\tau \in [0, T]$, where $\mathbf{x_0} \triangleq \mathcal{R}(x_0, \Delta t) \approx x_0 + \mathbf{\Delta}_x$, and $\mathbf{\Delta}_x \subset \mathbb{R}^n$. The set $\mathbf{x_0}$ encodes the reachable set of states of the system over the Δt time horizon. It can also be used to accommodate input delays, and state uncertainties.

Remark 1: The significance of result (9) is that it removes the need for integration over sets and allows for the discrete evaluation of the CBF at the boundary of the reachable set, provided the system is incrementally stable [11].

For the remainder of this work, it will be assumed that the considered backup controllers render the dynamics incrementally stable, such that (9) can be evaluated from a finite set of points at the reachable set boundary. Furthermore, as suggested in [8], the conditions in (9) are evaluated at a finite set of times $\tau_k \in [0,\ T]$.

E. Decentralized Multi-Robot CBF

Let us regard ∇_i as the gradient with respect to the state of robot-i. As leveraged in previous work [8], [3], a decentralized implementation of the $h^S(x)$ condition (9) for the pair (i,j) can be implemented by robot-i as

$$\begin{aligned}\nabla_i h^B_{i,j}\left(\phi^{u_B}_T|_{\mathbf{x_0}}\right)\left(\nabla\phi^{u_B}_T|_{\mathbf{x_0}}\tilde{f}(\mathbf{x_0},u)\right)\\ +\frac{1}{2}\alpha\left(h^B_{i,j}\left(\phi^{u_B}_T|_{\mathbf{x_0}}\right)\right)\geq 0\\ \nabla_i h^C_{i,j}\left(\phi^{u_B}_\tau|_{\mathbf{x_0}}\right)\left(\nabla\phi^{u_B}_\tau|_{\mathbf{x_0}}\tilde{f}(\mathbf{x_0},u)+\tfrac{\partial}{\partial t}\phi^{u_B}_\tau|_{\mathbf{x_0}}\right)\\ +\frac{1}{2}\alpha\left(h^C_{i,j}\left(\phi^{u_B}_\tau|_{\mathbf{x_0}}\right)\right)\geq 0\end{aligned}\tag{10}$$

$\forall\tau\in[0,T]$. Here, $\mathbf{x_0}\subset\mathbb{R}^{2\cdot n}$ is sampled from the reachable set of both robot-i and robot-j, but the gradient is only relative to states of robot-i. Given the control-affine form of the dynamics (1), the inequalities in (10) will have the form $A_i u_i \leq b_i$.

IV. Safe Task Satisfaction

For each robot-$i\in\mathcal{I}$, we are interested in tasks that can be encoded in the Mayer term of the OCP in (4), which include terminal constraints or cost on the final state. To capture agressively maneuvering interactions, this work focuses on minimum-time problems, where the Mayer term in (4) specify a desired state of the system to be reached, and $L(x,u,t)=1$. The solution to this optimization problem provides an optimal input trajectory as well as a minimum time $(u^\star(t),T^\star)$.

The main technical challenge with decentralized multi-agent problems is that the optimal flows $\phi^{u^\star}_\tau(x_i)\ \forall i\in\mathcal{I}$ might not be available *a priori* to all agents, especially in dynamic environments, to verify and actively enforce safety. The proposed solution is based on [7] where it is assumed that robot-i only has access to state information of its neighboring robots such that decentralized CBF in the form of (10) can be enforced. However, the proposed solution in that work relied on an optimization objective that combined liveliness property for all agents. Given this does not hold on the current work, decentralized enforcement of (10) can yield safe agent trajectories that never satisfy the task encoded in their Mayer term. Thus, we propose a new strategy that imposes a leader-follower topology on the CBF condition in (10) to ensure task completion of a selected leader robot.

A. Task Definitions

We now focus on defining the type of task we want to perform and conditions under which safe task satisfaction can be ensured in a decentralized fashion. In what follows, consider a team of N robots with dynamics of the form (1). For a robot-i, consider its neighbor set $\mathcal{N}$. This neighborhood set can either be $\mathcal{I}\setminus i$ or nearby robots for which safety needs to be actively preserved as described in [7].

Definition 2 (Minimum Mayer Value): Considering the OCP in (4) only subject to constraint (4.a). We define the Minimum Mayer Value as $M^\star=\min_{x\in\mathcal{X}}M(x)$ where $\mathcal{X}=\{x\in\mathbb{R}^n\mid h_i(x)\geq 0\}$.

Definition 3 (Minimum Time): The minimum time $T^\star$ is given by the solution of the OCP (4) only subject to constraint (4.a) when $L(x_i,u_i,t)=1$.

Definition 4 (Time-Critical Task): A task is Time-Critical if the robot needs to achieve the Minimum Mayer Value in Minimum Time, i.e., $M\left(\phi^{u^\star}_{T^\star}(x_0)\right)=M^\star$.

Definition 5 (Persistent Task): A task is persistent if the Minimum Mayer Value can be achieved at $t\in[T^\star\ \infty)$.

Definition 6 (Safe Task Completion): Both Time-Critical or Persistent Tasks are completed safely by robot-i if it can also satisfy constraint $h_{i,j}(x_i,x_j)$ (4.b) for all $j\in\mathcal{N}$ while achieving the Minimum Mayer Value.

Definition 7 (Leader-Follower CBF): Consider the team-wise safety objective $h^C_{i,j}(x_i,x_j)\geq 0$, for a leader robot-i and its follower neighbor set $j\in\mathcal{N}$. The Leader-Follower CBF is given by

$$\begin{aligned}\nabla_j h^B_{i,j}\left(\phi^u_T(x_i),\phi^{u_B}_T|_{\mathbf{x}_{0,j}}\right)\left(\nabla\phi^{u_B}_T|_{\mathbf{x}_{0,j}}\tilde{f}(\mathbf{x}_{0,j},u)\right)\\ +\alpha\left(h^B_{i,j}\left(\phi^u_T(x_i),\phi^{u_B}_T|_{\mathbf{x}_{0,j}}\right)\right)\geq 0\\ \nabla_j h^C_{i,j}\left(\phi^u_\tau(x_i),\phi^{u_B}_\tau|_{\mathbf{x}_{0,j}}\right)\Big(\nabla\phi^{u_B}_\tau|_{\mathbf{x}_{0,j}}\tilde{f}(\mathbf{x}_{0,j},u)\\ +\tfrac{\partial}{\partial t}\phi^{u_B}_\tau(\mathbf{x}_{0,j})|_{\mathbf{x}_{0,j}}\Big)+\alpha\left(h^C_{i,j}\left(\phi^u_\tau(x_i),\phi^{u_B}_\tau|_{\mathbf{x}_{0,j}}\right)\right)\geq 0\end{aligned}\tag{11}$$

$\forall\tau\in[0,T]$, where x_i is the state of robot-i at $\tau=0$, and $\mathbf{x}_{0,f}$ is the reachable set of robot-j for $\tau\in[0,\Delta t]$. It is assumed that the robot-i follows the controller selected for the evaluation of its flow. Thus the term $\nabla\phi^u_T(x_i)f(x_i,u)-\frac{\partial}{\partial t}\phi^u_\tau(x_i)=0$. The inequality in (11) is only imposed on robot-j, and given the control-affine form of (1), it will have the form $A_j u_j\leq b_j$.

Remark 2: Note that *Definition* 7 interprets the enforcement of the inequality (9) in a leader-follower sense because robot-i is free to select its input, then robot-j selects its input to enforce the inequality.

B. Decentralized Safety in Multi-Robot Teams

We now provide the main results of this work that capture sufficient conditions under which safe task satisfaction can be achieved by a multi-robot team in a decentralized environment for aggressively maneuvering trajectories.

Lemma 1: For every initial condition $x_0\in\mathbf{x}_0$ and $\tau\in[0,T]$, if the backup controller $u_B(x)\in\mathcal{U}$ provides a system flow $\phi^{u_B}_\tau(x)\in\mathcal{S}\subseteq\mathcal{C}$, then $K_{cbf}\left(\phi^{u_B}_\tau(x_0)\right)\neq\emptyset$ for all $\tau\in[0,T]$.

Proof: This follows from Theorem 1, which states that h^S is a control barrier function that renders $\mathcal{S}\subseteq\mathcal{C}$ forward invariant. Thus, at each $x\in\phi^{u_B}_\tau(x_0)$ and $\tau\in[0,T]$, there exists a extended class-$\mathcal{K}$ function that satisfies Definition 1. This renders the set in (6) not empty. ■

Lemma 2: For the Leader-Follower CBF in Definition 7, if for all $x_j\in\mathbf{x}_{0,j}$ and $\tau\in[0,T]$ the objective $h_{i,j}\geq 0$, then $K_{cbf}\left(\phi^u_\tau(x_i),\phi^{u_B(x)}_\tau(x_j)\right)\neq\emptyset$ for all $\tau\in[0,T]$.

Proof: This follows from defining a joint flow of robot-i and robot-j, then recalling Lemma 1. ■

Proposition 1: If for every $i\in\mathcal{I}\setminus i_L$ there exists a backup controller $u_B(x)$ with time horizon $T\geq\Delta t$, which simultaneously yields $h_i\geq 0$ and $h_{i,j}\geq 0$ for all $x_0\in\mathbf{x}_0$,

$\forall j \in \mathcal{N}$, $\forall \tau \in [0,T]$, then the decentralized OCP problem

$$\begin{aligned} \min_{u_i \in \mathcal{U}_i, t_f} \quad & M(x_i,T) + \int_0^{t_f} L(x_i,u_i,t)dt \\ \text{s.t.} \quad & \dot{x}_i = f(x_i) + g(x_i)u_i, \\ & A_i u_i \leq b_i, \ \forall t \in [0,\Delta t] \end{aligned} \tag{12}$$

where $A_i u_i \leq b_i$ arise from CBF conditions in (9), (10) and/or (11), is guaranteed to be feasible while rendering the safe set $\mathcal{S} \subseteq \mathcal{C}$ forward invariant $\forall t \in [0,\Delta t]$.

Proof: The existence of backup controllers for all $i \in \mathcal{I} \setminus i_L$, which satisfy inequalities $h_i \geq 0$ and $h_{i,j} \geq 0$ over the flow of all systems, imply $\mathcal{X}_i = \{x_i \in \mathbb{R}^n \mid \dot{x}_i = f(x_i) + g(x_i)u,\ h(x_i) \geq 0,\ h_{i,j}(x_i,x_j) \geq 0,\ \forall i \in \mathcal{I} \setminus i_L\}$ is not empty. By Lemma 1-2, the CBF inequalities can be constructed such that the input set is not empty over the backup controller time horizon. Forward invariance follows from recalling the conditions from Lemma 1 for individual objectives h_i or joint objectives of type (10), and Lemma 2 for Leader-Follower CBF with leader index i_L. ■

Theorem 2: Consider a NMPC approach that solves the trajectory optimization problem in (12) at every Δt. Also consider a time horizon T_f over which the NMPC problems will be executed for the robot team. If the conditions of Proposition 1 are satisfied at the beginning of every NMPC computation $\forall t \in [0,T_f]$, then the NMPC approach renders the safe set $\mathcal{S} \subseteq \mathcal{C}$ forward invariant over $[0,T_f]$.

Proof: Given the conditions of Proposition 1 are satisfied at every NMPC computation time, the safe set $\mathcal{B}$ is forward invariant $\forall t \in [(n-1)\Delta t, n\Delta t]$ where $n = \{1,\ldots,\lceil T_f/\Delta t \rceil\}$. Thus, the safe set is forward invariant $\forall t \in [0,T_f]$. ■

What Theorem 2 states is that by designing a backup safety controller that ensures safety over a time horizon greater than the NMPC routine, and constructing CBF inequalities based on this safe trajectory, we can ensure the NMPC solution is safe over the computation horizon. This allows the NMPC routine great flexibility in optimizing the individual robot trajectories while guaranteeing safety, but it does not ensure task completion. The next Theorem states that task completion is achieved by the leader robot.

Theorem 3: If at a time t_0, robot-i_L has a solution $(u^\star, T^\star)$ to OCP (4)-(4.a) that achieves the Minimum Mayer Value, and every $i \in \mathcal{I} \setminus i_L$ robot selects trajectories through the NMPC in Theorem 2 with i_L as the leader, then robot-i_L achieves safe time-critical task completion.

Proof: By the theorem statement, robot-i_L can achieve minimum-time task completion in the absence of robot-to-robot safety constraints. Given that $i \in \mathcal{I} \setminus i_L$ satisfy the conditions in Proposition 1, the Leader-Follower CBF can be enforced by the multi-robot team, rendering $\mathcal{S} \subseteq \mathcal{C}$ forward invariant. Thus, robot-i_L can complete the time critical task, while robots $i \in \mathcal{I} \setminus i_L$ ensure safety for the multi-robot team. ■

When robot-i_L is selected as leader, the following algorithm verifies if robot-i can simultaneously use the same backup controller to enforce the Leader-Follower CBF in (11) and (10) for robots $j \in \mathcal{N} \setminus i_L$. Let us define U_B as the set of backup controllers.

Algorithm 1 Leader Selection for LF-CBF

procedure Feasible-Leader
$LF \leftarrow 0,\ K \leftarrow \emptyset$
for $u_B^k \in U_B$ **do**
  C1: Verify (9) for individual safety
  C2: Verify (10) $\forall j \in \mathcal{N} \setminus i_L$ for robot-team safety
  C3: Verify (11) for robot leader i_L
  if C1 $\wedge$ C2 $\wedge$ C3 **then**
    $LF \leftarrow 1$
    $K \leftarrow K \cup \{k\}$
  end if
end for

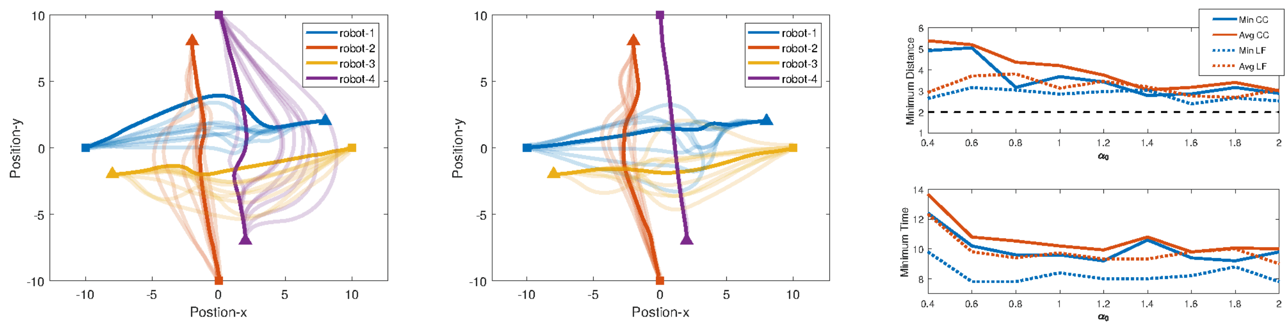
The rational for Algorithm 1 is as follows. In the fully cooperative setting, safety backup controllers that enforce simple maneuvers such as decelerating and turning have been shown effective to ensure safety of a robot team [8]. In the Leader-Follower CBF that is not the case, and the design of controllers with safety guarantees is beyond the scope of this work. The adopted approach verifies that at least one backup strategy is sufficient to guarantee safety against the leader for its entire task horizon, while still ensuring safety against all other robots in the team. Once a leader is selected, the Algorithm 2 can be executed in real time on all robots to ensure safety. If all robots contain backup strategies that render the problem feasible for the selected leader, then Proposition 1 guarantees they can solve the NMPC problem with the Leader-Follower CBF in (11). If the conditions of Proposition 1 are not verified by all robots, then they ensure safety through (10).

Algorithm 2 LF-CBF for Safe Task Satisfaction

procedure NMPC-CBF
Solve NMPC (4) without state constraints, obtain $u^\star$
$u_B \leftarrow \arg\min_{u_B^k | k \in K} \|u^\star - u_B^k\|$
if $LF_j = 1\ \forall j \in \mathcal{N}$ **then**
  Solve NMPC (12) subject to (9), (10), (11)
else
  Solve NMPC (12) subject to (9), (10)
end if

By Theorem 3, Algorithm 2 guarantees robot-i_L can complete its time critical task safely. It is thus expected for robot-i_L to select its optimal input command by solving (4) only subject to (4.a). It is important to note that the leader selection process is problem specific.

Theorem 4: Consider a team of N robots with persistent tasks. Assume for $i \in \mathcal{I} \setminus i_L$, a backup safety controller $u^B(x)$ can always be found for any leader $i_L \in \mathcal{I}$, and the conditions of Theorem 2 are verified. Let us adopt the notation of $i_L(t_0)$ for a leader that is selected at time t_0. Then selecting a new leader $i_L(t_n) \in \mathcal{I} \setminus i_L(t_{n-1}) \setminus \cdots \setminus i_L(t_0)$



(a) Cooperative CBF (CC) Implementation (b) Leader-Follower CBF (LF) Implementation (c) Performance Metrics

Fig. 3. Four robot scenario. (a) Fully cooperative implementation enabled by setting LF = 0 in Algorithm 1. (b) LF implementation selects the robot with the smallest t_f value from (15) as a leader. Due to the initialization, robot-4 was always selected as the first leader. (a)-(b) Triangles provide initial position, and squares the final position. Shaded trajectories correspond to varying $\alpha_0 \in [0.4,\ 2]$, with the solid line corresponding to $\alpha_0 = 2$. (c) Minimum distance provided by $\min_t \|c(x_i(t) - x_j(t))\|$ for every (i,j) robot pair. The minimum task completion time was collected for each scenario only for robots 1 through 3

after a previous leader $i_L(t_{n-1})$ achieves task satisfaction, in Algorithm 1-2 leads to safe task satisfaction by the entire multi-robot team.

Proof: At time t_0 selecting the leader $i_L(t_0)$, by the theorem statement we have that all robots $i \in \mathcal{I} \setminus i_L$ will find i_L feasible in Algorithm 1. Then solving the OCP in (12) with Leader-Follower CBF for $i_L(t_0)$ in Algorithm 2 will lead to safe Task Satisfaction by robot-$i_L(t_0)$ according to Theorem 3. After $i_L(t_0)$ achieves safe task satisfaction, select a new leader $i_L(t_1) \in \mathcal{I} \setminus i_L(t_0)$. By the theorem conditions, Algorithm 1-2 will again lead to safe Task Satisfaction by robot-$i_L(t_1)$. Applying this rule for every new leader $i_L(t_n) \in \mathcal{I} \setminus i_L(t_{n-1}) \setminus \cdots \setminus i_L(t_0)$ leads to safe task satisfaction by the multi-robot team. ■

Theorem 4 provides sufficient conditions for multi-robot task satisfaction by leveraging a backup saftety controller, which allows for the Leader-Follower CBF formulation in (11). The significance of this result is that, even in the absence of shared knowledge of every task for a multi-robot team, safe task satisfaction is still enabled in a decentralized fashion. The OCP problem formulation allows for instances where task satisfaction requires robot trajectories with aggressive maneuvers as shown in the Section V.

V. Implementation Example

For all scenarios, the considered dynamics are of the form

$$\begin{aligned} \dot{x}_p &= v\cos\theta \\ \dot{y}_p &= v\sin\theta \\ \dot{v} &= a \\ \dot{\theta} &= \omega \end{aligned}, \quad f(x) = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ 0 \\ 0 \end{bmatrix}, \quad g(x) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (13)$$

with states $x_i^\top = \begin{bmatrix} x_p & y_p & v & \theta \end{bmatrix}$, and input $u_i^\top = \begin{bmatrix} a & \omega \end{bmatrix}$. The input set $\mathcal{U}_i$ is constructed by imposing box constraints on the input $\text{lb} \le u_i \le \text{ub}$. The safety feature of interest is collision avoidance

$$h_{i,j}(x_i, x_j) = \|c(x_i - x_j)\| - d_s,\ c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (14)$$

where d_s is a safety distance.

The OCP problem solved is given by

$$\begin{aligned} \min_{u,\delta,t_f} \quad & \delta^\top Q_\delta \delta + \int_{t_0}^{t_f} 1 dt \\ \text{s.t. } & \dot{x} = f(x) + g(x)u \\ & M(x) = \begin{bmatrix} x_p & y_p \end{bmatrix}^\top - p_d + \delta = 0 \\ & Au \le b,\ \forall \tau \in [t_0,\ t_0 + \Delta t] \\ & \text{lb} \le u \le \text{ub},\ \forall \tau \in [t_0, t_f] \end{aligned} \quad (15)$$

where the inequity constraints $Au \le b$ are constructed from the conditions in Algorithms 1-2. For all robots, $\text{ub} = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top$, $\text{lb} = -\text{ub}$. The NMPC horizon $\Delta t = 0.2[s]$. The OCP in (15) was solved in ACADO [9], and all simulation results developed in MATLAB 2020a.

The selected backup controllers where breaking and break while turning actions as described by

$$u_B^1(x) = \begin{bmatrix} \sigma(-k_v v) \\ 0 \end{bmatrix},\ u_B^{(2,3)}(x) = \begin{bmatrix} \sigma(-k_v v) \\ \sigma(k_\theta(\theta_d - \theta)) \end{bmatrix}, \quad (16)$$

where $\sigma(x) = 2/(1+\exp(-2x)) - 1$ is a saturation function to enforce the maximum input constraint. The backup controllers look to bring the dynamics to an equilibrium point given by $v = 0$ for any position and angle. Thus, as suggested in [8], the backup safety set was taken as the intersection between the equilibrium points in the system and the safe set induced by (14). The parameters in (16) were selected to ensure an equilibrium point could be reached by $T = 2$. In all CBF realizations, the selected class-$\mathcal{K}$ function is given by $\alpha(x) = \alpha_0 x$.

VI. Analysis

The first two-robot scenario is presented in Fig. 1 (first page), where the blue robot is a human-controlled robot, and the orange robot is autonomous. In this scenario, robot-i is always selected as the leader in Algorithm 1, and Algorithm 2 includes additional constraints for lane keeping. The perception block provides the expected leader flow $\phi_T^u(x_i)$, and the Mayer term in (15) is selected by the lane keeping logic. For the presented scenario, the logic includes a simple lane change if the speed of the robot falls below a certain threshold, and the LF-CBF approach ensures safety of the multi-robot interaction. Notice how in these type of scenarios a cooperative approach to safety is not expected, thus the LF-CBF approach allows for robot-j to ensure safety in the presence of actuator constraints.

A more in depth experiment is provided in Fig. 3, where four robots are attempting to reach a target position across the 2D arena. All robots are initialized at equal length to their target position, except robot-4 which is 1 distance unit closer. Fig. 3(a) shows a fully cooperative implementation where the LF flag in Algorithm 1 is always set to false; hence, no leader is selected. Fig. 3(b) shows the Leader-Follower implementation, where the leader is selected as the agent with the smallest t_f from (15). The different traces correspond to different values $0.4 \leq \alpha_0 \leq 2$, the solid line corresponding to $\alpha_0 = 2$ in both plots. First, notice how even though at each NMPC iteration a minimum time problem is solved, depending on the conservatism of the CBF, the resulting trajectory can deviate largely from the minimum time one. This is especially noticeable in Fig. 3(a) where robot-4 deviates largely for smaller values of α_0. In contrast, the Leader-Follower implementation shown in Fig. 3(b) enables safe task completion on robot-4, and of every subsequent leader.

Fig. 3(c) show the minimum and average values of the performance metrics for this scenario under varying values of α_0. First note that the minimum distance $d_s = 2$ was not violated on any experiment, with the most amount of conservatism provided for the smaller values of α_0. The minimum time metric excludes the performance of robot-4 to make a relevant comparison between the (CC) and (LF) implementations. This is because robot-4 was always selected the first leader in the (LF) implementation, leading to the same smallest task completion. Note that even though the leader is selected one at a time, the minimum task completion time of the remaining robots was always smaller in the (LF) implementation.

VII. Conclusions

This work presented an approach to ensuring safety in aggressively maneuvering robot interactions by combining tools from nonlinear model predictive control and control barrier functions. Control barrier functions allow for a scalable way of ensuring safety by providing temporally local constraints on the individual robot inputs to preserve the existence of a safe trajectory. This approach was extended to NMPC by accounting for the numerical solution computation time as an input delay of a continuous time system. Implementing safety in a leader-follower sense, where a leader is selected to complete its minimum time task independent of team safety constraints, allows for individual minimum time task satisfaction. The approach does not required a common shared objective for the multi-robot task as in the CLF-based cooperation, thus extending the NMPC implementation to hierarchical approaches to autonomy where a path planning layer might provide a set of way-points, and the path following layer tracks them in real-time. Future work includes the exploration of strategies that allow for multiple leader selection while ensuring safety, and accounting for perception estimation errors. This will extend the proposed approach to environments where no cooperation is expected in multi-robot interactions.

References

[1] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191617–191643, 2020.

[2] P. Rivera-Ortiz, Y. Diaz-Mercado, and M. Kobilarov, "Multi-player pursuer coordination for nonlinear reach-avoid games in arbitrary dimensions via coverage control," in *2020 American Control Conference (ACC)*, pp. 2747–2753, IEEE, 2020.

[3] A. Davydov, P. Rivera-Ortiz, and Y. Diaz-Mercado, "Pursuer coordination in multi-player reach-avoid games through control barrier functions," *IEEE Control Systems Letters*, 2020.

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

[5] 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)*, pp. 3420–3431, IEEE, 2019.

[6] W. S. Levine, L. Grüne, R. Goebel, S. V. Rakovic, A. Mesbah, I. Kolmanovsky, S. Di Cairano, D. A. Allan, J. B. Rawlings, M. A. Sehr, *et al.*, "Handbook of model predictive control," 2018.

[7] G. Notomista and M. Egerstedt, "Persistification of robotic tasks," *IEEE Transactions on Control Systems Technology*, 2020.

[8] Y. Chen, A. Singletary, and A. D. Ames, "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: a control barrier function approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 127–132, 2020.

[9] B. Houska, H. J. Ferreau, and M. Diehl, "Acado toolkit—an open-source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[10] P. E. Gill, W. Murray, and M. H. Wright, *Practical optimization*. SIAM, 2019.

[11] A. Singletary, Y. Chen, and A. D. Ames, "Control barrier functions for sampled-data systems with input delays," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 804–809, IEEE, 2020.

[12] E. Squires, P. Pierpaoli, and M. Egerstedt, "Constructive barrier certificates with applications to fixed-wing aircraft collision avoidance," in *2018 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 1656–1661, IEEE, 2018.

[13] L. Wang, D. Han, and M. Egerstedt, "Permissive barrier certificates for safe stabilization using sum-of-squares," in *2018 Annual American Control Conference (ACC)*, pp. 585–590, IEEE, 2018.

[14] T. Gurriet, M. Mote, A. D. Ames, and E. Feron, "An online approach to active set invariance," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 3592–3599, IEEE, 2018.