Guaranteed Obstacle Avoidance for Multi-Robot Operations With Limited Actuation: A Control Barrier Function Approach

Yuxiao Chen, Andrew Singletary, and and Aaron D. Ames, *Senior Member, IEEE*

***Abstract*—This letter considers the problem of obstacle avoidance for multiple robotic agents moving in an environment with obstacles. A decentralized supervisory controller is synthesized based on control barrier functions (CBF) that guarantees obstacle avoidance with limited actuation capability. The proposed method is applicable to general nonlinear robot dynamics and is scalable to an arbitrary number of agents. Agent-to-agent communication is not required, yet a simple broadcasting scheme improves the performance of the algorithm. The key idea is based on a control barrier function constructed with a backup controller, and we show that by assuming other agents respecting the same CBF condition, the supervisory control algorithm can be implemented decentrally and guarantees obstacle avoidance for all agents.**

***Index Terms*—Robotics, decentralized control, autonomous vehicles.**

I. Introduction

OBSTACLE avoidance is one of the core requirements for robotic applications. There exist a plethora of methods in the literature, such as the artificial potential field [1] and cell decomposition method [2]. When the robot is subject to both kinetic and dynamic constraint such as nonholonomic dynamic constraint and limited actuation, the problem is referred to as kinodynamic motion planning. In this case, roadmap methods such as probabilistic roadmaps (PRM) [3], [4] and rapidly-exploring random tree (RRT) [5], [6], and optimization based method such as spline optimization [7], [8] and model predictive control [9], [10] are among the popular methods. In addition, approaches based on reachability analysis and set invariance such as the Hamilton Jacobi Isaac PDE [11] and control barrier functions [12], [13] were proposed which are capable of keeping the state within the safe set for all time, thus guaranteeing obstacle avoidance. However, these methods usually suffer from poor scalability and cannot be applied to high dimensional systems. For decentralized control structure, guaranteeing safety is particularly hard due to the lack of information [14].

Based on [15], we propose a CBF approach for multi-agent obstacle avoidance that

- 1. can guarantee 100% obstacle avoidance under limited actuation for popular robotic systems.
- 2. is computed with simple forward simulation and thus is applicable to general nonlinear dynamics.
- 3. can be implemented completely decentrally for multiple agents and scales to an arbitrary number of agents.

The core idea is to equip each agent with one or multiple backup strategies that bring the agent to an equilibrium point and check whether the corresponding backup trajectory satisfies the safety constraint. In fact, we will show later in this letter that all initial conditions whose corresponding backup trajectories satisfy the safety constraint constitute a control invariant set. Then by enforcing the CBF supervisory controller, if the backup trajectory associated with the initial condition satisfies the safety constraint, the state can be kept within the safe set indefinitely. Furthermore, we show that the CBF condition can be implemented decentrally with no communication between agents and can scale to an arbitrary number of agents. Nonetheless, communication between agents is helpful, especially in the case with multiple backup strategies, and we propose a simple broadcast scheme that to guarantee compatibility between agents.

The proposed approach bears some similarity to motion primitives [16] as the backup trajectory can be viewed as a simple motion primitive. However, the key difference is that the agent almost never execute the backup strategy. Instead, the backup strategy is used as a feasibility check to make sure that an equilibrium point can be always be reached safely.

For the remainder of this letter, Section II reviews the basics of control barrier functions and the backup strategy approach for generating control barrier functions. Sections III and IV present the main result, using control barrier functions with a single or multiple backup strategies to achieve obstacle avoidance for multi-robot systems. The simulation and experimental results are presented in Section V and finally we draw conclusions in Section VI.

Nomenclature: For the remainder of this letter, given a control strategy π, $f_\pi \doteq f(x,\pi(x))$ denotes the closed loop dynamics under π. $\Phi_{f_\pi} : \mathbb{R}^n \times [0,\infty)$ denotes the flow map

Manuscript received March 16, 2020; revised May 13, 2020; accepted June 1, 2020. Date of publication June 9, 2020; date of current version June 22, 2020. The work of Yuxiao Chen was supported by SRI/AFOSR under Award FA8750-19-C-0089. The work of Andrew Singletary was supported by DARPA under Award NNN12AA01C. The work of Aaron D. Ames was supported by NSF CPS under Award 1932091. Recommended by Senior Editor F. Dabbene. *(Corresponding author: Yuxiao Chen.)*

The authors are with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA 91106 USA (e-mail: chenyx@caltech.edu).

Digital Object Identifier 10.1109/LCSYS.2020.3000748

of f_π, that is, $\Phi_{f_\pi}(x_0, t)$ is the state at time t given the initial state $x(0) = x_0$ and state evolution following $\dot{x} = f_\pi(x)$. For a fixed t, to simplify the notation, $\Phi^t_{f_\pi} \doteq \Phi_{f_\pi}(\cdot, t)$.

II. Preliminaries

A. Control Barrier Functions

Control barrier functions [17], [18] uses a supervisory controller that keeps the system safe with minimum intervention. Specifically, consider the dynamic system described as

$$\dot{x} = f(x, u), \quad x \in \mathbb{R}^n, u \in \mathcal{U} \subseteq \mathbb{R}^m, \tag{1}$$

where f is assumed to be affine in u. Suppose there exists a function $h : \mathbb{R}^n \to \mathbb{R}$ that satisfies

$$\begin{aligned} &\forall\ x \in \mathcal{X}_0, && h(x) \geq 0 \\ &\forall\ x \in \mathcal{X}_d, && h(x) < 0 \\ &\forall\ x \in \{x|h(x) \geq 0\}, \quad \exists\ u \in \mathcal{U} \text{ s.t. } && \dot{h} + \alpha(h) \geq 0, \end{aligned} \tag{2}$$

where $\mathcal{X}_0$ is the set of initial states and $\mathcal{X}_d$ is the danger set that we want to keep the state away from. $\alpha(\cdot)$ is a class-$\mathcal{K}$ function, i.e., $\alpha(\cdot)$ is strictly increasing and satisfies $\alpha(0) = 0$. Then h is called a control barrier function, and for any legacy controller, the CBF controller is a supervisory controller that enforces the state to stay inside $\{x|h(x) \geq 0\}$ with the following quadratic programming:

$$\begin{aligned} u^\star = \arg\min_{u\in\mathcal{U}} \ & \left\| u - u^0 \right\|^2 \\ \text{s.t.} \ & \nabla h \cdot f(x, u) + \alpha(h) \geq 0, \end{aligned} \tag{3}$$

where u^0 is the input of the legacy controller. To ensure that (3) is always feasible when $h(x) \geq 0$, $\{x|h(x) \geq 0\}$ need to be a control invariant set, which is defined as follows.

Definition 1: A set $\mathcal{S}$ is a control invariant set if there exists a control law $\pi : \mathbb{R}^n \to \mathcal{U}$ such that for all initial condition $x(0) \in \mathcal{S}$, $\forall t \geq 0$, $\Phi^t_{f_\pi}(x(0)) \in \mathcal{S}$.

Depending on the dynamics, there exists numerous methods to construct control invariant sets, such as polytopic operations [19], [20], linear programming [21], [22], and sum of squares programming [23], [24]. In Section II-B we shall generate a control invariant set from a backup strategy.

B. Control Barrier Function With Backup Controller

In this section, we review the backup strategy approach for control invariant set generation [15]. The following problem is considered. Given a dynamic system described by (1), the following constraint is required to hold for all $t \geq 0$:

$$x(t) \in \mathcal{C} \doteq \{x|h^{\mathcal{C}}(x) \geq 0\},$$

where $h^{\mathcal{C}} : \mathbb{R}^n \to \mathbb{R}$ is the function that defines the safe set $\mathcal{C}$, assumed to be smooth. Ideally, if a control invariant set $\mathcal{S} \subseteq \mathcal{C}$ is known, a control barrier function can be constructed and the supervisory controller (3) guarantees that for any initial condition $x(0) \in \mathcal{S}$, the state will be kept within $\mathcal{S}$, and thus within $\mathcal{C}$. However, such an $\mathcal{S}$ might be difficult to compute, especially when the state dimension is high and $\mathcal{S}$ is required to be large. To resolve this problem, the backup strategy approach was first discussed in [15] and [25]. Suppose we obtain a small initial control invariant set $\mathcal{S}_0 \doteq \{x|h^{\mathcal{S}}(x) \geq 0\} \subseteq \mathcal{C}$, where $h^{\mathcal{S}}$ is assumed to be smooth. This is a reasonable assumption since it is often easier to obtain a small control invariant set, e.g., by linearizing the dynamics. One extreme case is that any equilibrium point of (1) inside $\mathcal{C}$ satisfies the requirement. Then, for a given backup strategy $\pi:\mathbb{R}^n \to \mathcal{U}$ and a fixed horizon T, define

$$\mathcal{S} \doteq \{x|\Phi^T_{f_\pi}(x) \in \mathcal{S}_0 \wedge \forall t \in [0, T], \Phi^t_{f_\pi}(x) \in \mathcal{C}\}, \tag{4}$$

which is the set of all initial conditions from which the backup strategy would bring the state to $\mathcal{S}_0$ at $t = T$ while satisfying the constraint $\forall t \in [0, T], x(t) \in \mathcal{C}$. For the remainder of this letter, $t \to \Phi^t_{f_\pi}(x)$ is denoted as the backup trajectory.

Since $\mathcal{S}_0$ is a control invariant set, by Definition 1, there exists a control law π_0 that keeps any state starting inside $\mathcal{S}_0$ within $\mathcal{S}_0$. Therefore, we fix $\pi|_{\mathcal{S}_0} = \pi_0|_{\mathcal{S}_0}$, i.e., any state reaching $\mathcal{S}_0$ will be kept within $\mathcal{S}_0$ under π. Then, $\mathcal{S}$ denotes the initial condition that can reach $\mathcal{S}_0$ within $[0, T]$ (instead of exactly at T) while satisfying the state constraint.

Lemma 1: $\mathcal{S}$ is a control invariant set and $\mathcal{S}_0 \subseteq \mathcal{S} \subseteq \mathcal{C}$.

Proof: Simply take the backup strategy π as the control law, any state $x(0) \in \mathcal{S}$ is kept within $\mathcal{S}$. Then by Definition 1, $\mathcal{S}$ is a control invariant set. $\mathcal{S}_0 \subseteq \mathcal{S}$ follows from the fact that $\mathcal{S}_0$ is a control invariant set, therefore $x \in \mathcal{S}_0 \to x \in \mathcal{S}$. $\mathcal{S} \subseteq \mathcal{C}$ follows from the fact that $x \notin \mathcal{C} \to x \notin \mathcal{S}$. ∎

Next, we show that a control barrier function can be constructed from the control invariant set $\mathcal{S}$.

Lemma 2: $\mathcal{S}$ is the 0-level set of the following function

$$h(x) = \min\{\min_{t\in[0,T]} h^{\mathcal{C}}(\Phi^t_{f_\pi}(x)), h^{\mathcal{S}}(\Phi^T_{f_\pi}(x))\}. \tag{5}$$

Proof: First notice that by the continuity of the flow function Φ_{f_π} and the min function, h is continuous. For all $x \in \mathcal{S}$, by definition, under the backup strategy π, the state evolution $\Phi^t_{f_\pi}(x)$ would satisfy the constraint and reach $\mathcal{S}_0$ at time T, therefore $h(x) \geq 0$. On the other hand, for all $x \notin \mathcal{S}$, under the backup strategy π, the state evolution either violates the state constraint at some t, i.e., $\exists t \in [0, T], h^{\mathcal{C}}(\Phi^t_{f_\pi}(x)) < 0$, or does not reach $\mathcal{S}_0$ within the horizon T, i.e., $h^{\mathcal{S}}(\Phi^T_{f_\pi}(x)) < 0$, indicating that $h(x) < 0$. Therefore, $\mathcal{S} = \{x|h(x) \geq 0\}$. ∎

In order to use h as a control barrier function, the derivative (or subderivative) of h is needed. For any $t \in [0, T]$, the total derivative of $h^{\mathcal{C}}(\Phi^t_{f_\pi}(x))$ is computed as

$$\frac{dh^{\mathcal{C}}(\Phi^t_{f_\pi}(x))}{dt} = \nabla h^{\mathcal{C}} \frac{d\Phi^t_{f_\pi}}{dt} = \nabla h^{\mathcal{C}}(\nabla\Phi^t_{f_\pi}(x)f(x, u) - \frac{\partial \Phi^t_{f_\pi}}{\partial t}),$$

where $\nabla\Phi^t_{f_\pi}$ is the sensitivity matrix, i.e., the Jacobian of the future state $x(t)$ following the closed loop dynamics f_π with respect to the current state. For a fixed dynamics f_π and initial condition x, denote the sensitivity matrix at time t as $Q_{x,f_\pi}(t) \doteq \nabla\Phi^t_{f_\pi}$. Since $\Phi^t_{f_\pi}(x) = \int_0^t f_\pi(x(\tau))d\tau$, by the chain rule we have

$$Q_{x,f_\pi}(0) = I, \dot{Q}_{x,f_\pi}(t) = \nabla f_\pi \cdot Q_{x,f_\pi}(t), \tag{6}$$

which is an ordinary differential equation and can be solved efficiently. For more details, see [26]. Similar procedure is used to compute $\nabla(h^{\mathcal{S}} \circ \Phi^T_{f_\pi})$. With the gradient computed, to avoid the nondifferentiability of the min function, the following supervisory controller is used, which enforces a sufficient condition of the CBF condition:

$$u^\star = \arg\min_{u\in\mathcal{U}} \left\| u - u^0 \right\|^2$$

$$\text{s.t.}\quad \forall t \in [0,T],\ \frac{dh^{\mathcal{C}}(\Phi^t_{f_\pi}(x))}{dt}(x,u) + \alpha(h^{\mathcal{C}}(\Phi^t_{f_\pi}(x))) \geq 0,$$
$$\frac{dh^{\mathcal{S}}(\Phi^T_{f_\pi}(x))}{dt}(x,u) + \alpha(h^{\mathcal{S}}(\Phi^T_{f_\pi}(x))) \geq 0. \tag{7}$$

Proposition 1: For all $x \in \{x|h(x) \geq 0\}$, (7) is always feasible and (7) implies $\dot{h}(x,u) + \alpha(h(x)) \geq 0$. Moreover, h is a control barrier function that satisfies

- $\forall x \notin \mathcal{C}, h(x) < 0$,
- $\forall\ x \in \{x|h(x) \geq 0\}, \exists\ u \in \mathcal{U}$, s.t. $\dot{h}(x,u) + \alpha(h(x)) \geq 0$.

Proof: Note that $\frac{d\Phi^t_{f_\pi}}{dt}$ describes how the state flow deviates from the backup trajectory given the change of current state, thus by definition, it vanishes when taking $u = \pi(x)$. Therefore, when $h \geq 0$, $h^{\mathcal{C}}(\Phi^t_{f_\pi}(x)) \geq 0$, $h^{\mathcal{S}}(\Phi^T_{f_\pi}(x)) \geq 0$, which implies that $\alpha(h^{\mathcal{C}}(\Phi^t_{f_\pi}(x))) \geq 0$ and $\alpha(h^{\mathcal{S}}(\Phi^T_{f_\pi}(x))) \geq 0$. Therefore, $u = \pi(x)$ is a feasible solution to (7). Furthermore, by the definition of h and the properties of the min function, (7) implies $\dot{h} + \alpha(h) \geq 0$. For the second part of the proposition, the first property comes from the fact that $\forall x \notin \mathcal{C}, h(x) \leq h^{\mathcal{C}}(\Phi^0_{f_\pi}(x)) < 0$. The second property follows from the feasibility of (7). ■

III. Multiagent Obstacle Avoidance with CBFs

In this section, we show how the control barrier function based on backup strategies can be applied to multi-agent obstacle avoidance. We consider a multi-agent system consisting of N agents with state $x_1, \ldots, x_N$, respectively. The N states evolve with potentially heterogeneous dynamics:

$$\dot{x}_i = f_i(x_i, u_i), x_i \in \mathbb{R}^{n_i}, u_i \in \mathcal{U}_i. \tag{8}$$

For the whole system, let $x = [x_1^\intercal, x_2^\intercal, \ldots, x_N^\intercal]^\intercal$ denote the aggregated state and the dynamics for x is the following

$$\dot{x} = f(x,u) = \begin{bmatrix} f_1(x_1,u_1)^\intercal & \ldots & f_N(x_N,u_N)^\intercal \end{bmatrix}^\intercal.$$

For agent i, let $\pi_i : \mathbb{R}^n \to \mathcal{U}_i$ be its backup strategy and let $f_{\pi_i}(x) \doteq f_i(x_i, \pi_i(x))$ be the closed loop dynamics under π_i. The overall backup strategy given $\pi_1, \pi_2, \ldots, \pi_N$ is then denoted as π, where $u = \pi(x) = [\pi_1(x)^\intercal, \ldots, \pi_N(x)^\intercal]^\intercal$.

As introduced in Section II-B, the control barrier function constructed from a backup strategy requires an invariant set $\mathcal{S}_0$ to begin with. Although control invariant set can be difficult to compute, one control invariant set is almost free to obtain for most of the commonly seen robotic systems.

Definition 2: A point $x^e_\pi \in \mathbb{R}^n$ is a stable equilibrium point for a given backup strategy π if $f(x^e_\pi, \pi(x^e_\pi)) = 0$ and x^e_π is stable in the sense of Lyapunov under f_π.

We let $\mathcal{X}^e_\pi$ denote the set of all equilibrium points under π. For example, if the backup strategy π stabilizes the steady hovering maneuver of a drone, then any steady hovering state is an equilibrium point under π. Obviously, any subset of $\mathcal{X}^e_\pi$ is a control invariant set, and $\mathcal{S}_0$ is taken as $\mathcal{X}^e_\pi \cap \mathcal{C}$.

Given a multi-agent system as described previously, if all agents are controlled by a centralized controller, the CBF scheme should work with any backup strategy π that result in a nonempty $\mathcal{S}_0$. However, as mentioned previously, centralized control is usually not realizable due to the communication and scalability limitations. Therefore, the focus of this letter is on a decentralized implementation of a CBF-based supervisory controller. We assume that each agent can measure the states of other agents but independently select the control input. For the proposed decentralized scheme to work, the following assumption is needed.

Assumption 1: The state constraint $\mathcal{C}$ is pairwise decomposable, i.e.,

$$\mathcal{C} = \{h^{\mathcal{C}}(x) \geq 0\} = \{(\bigwedge_i h^{\mathcal{C}}_i(x_i) \geq 0) \wedge (\bigwedge_{i \neq j} h^{\mathcal{C}}_{ij}(x_i, x_j) \geq 0)\},$$

where $h^{\mathcal{C}}_i : \mathbb{R}^{n_i} \to \mathbb{R}$ is a constraint of only x_i, $h^{\mathcal{C}}_{ij} : \mathbb{R}^{n_i} \times \mathbb{R}^{n_j} \to \mathbb{R}$ is a constraint of x_i and x_j.

This is a reasonable assumption since for static obstacles, the obstacle avoidance constraint is on each agent; for agent-to-agent collision avoidance, the constraint is defined pairwise. Under Assumption 1, since $\mathcal{S}_0$ is taken as $\mathcal{X}^e_\pi \cap \mathcal{C}$, and the equilibrium point is defined on the state of each agent, $\mathcal{S}_0$ is also pairwise decomposable:

$$\mathcal{S}_0 = \{h^{\mathcal{S}}(x) \geq 0\} = \{(\bigwedge_i h^{\mathcal{S}}_i(x_i) \geq 0) \wedge (\bigwedge_{i \neq j} h^{\mathcal{S}}_{ij}(x_i, x_j) \geq 0)\}.$$

Next, we show how the CBF QP can be implemented in a decentralized structure. In a decentralized setting, the backup strategy for each agent is restricted to be a function that only depends on the agent itself, i.e., π_i only depends on x_i. Under Assumption 1, the original CBF QP in (7) becomes

$$\begin{aligned}
u^\star = \arg\min_{u \in \mathcal{U}}\ & \|u - u^0\|^2 \\
\text{s.t.}\ & \forall t \in [0,T],\ \forall i \neq j \in \{1,2,\ldots,N\}, \\
& \nabla h^{\mathcal{C}}_i(\nabla_{x_i}\Phi^t_{f_{\pi_i}} f_i(x_i,u_i) - \partial \Phi^t_{f_{\pi_i}}/\partial t) \\
& + \alpha(h^{\mathcal{C}}_i \Phi^t_{f_{\pi_i}}(x_i))) \geq 0, \\
& \nabla_{x_i} h^{\mathcal{C}}_{ij}(\nabla_{x_i}\Phi^t_{f_{\pi_i}} f_i(x_i,u_i) - \partial \Phi^t_{f_{\pi_i}}/\partial t) \\
& + \nabla_{x_j} h^{\mathcal{C}}_{ij}(\nabla_{x_j}\Phi^t_{f_{\pi_j}} f_i(x_j,u_j) - \partial \Phi^t_{f_{\pi_j}}/\partial t) \\
& + \alpha(h^{\mathcal{C}}_{ij}(\Phi^t_{f_{\pi_i}}(x_i), \Phi^t_{f_{\pi_j}}(x_j)) \geq 0, \\
& \nabla h^{\mathcal{S}}_i(\nabla_{x_i}\Phi^T_{f_{\pi_i}} f_i(x_i,u_i) - (\partial \Phi^t_{f_{\pi_i}}/\partial t)|_{t=T}) \\
& + \alpha(h^{\mathcal{S}}_i \Phi^T_{f_{\pi_i}}(x_i))) \geq 0, \\
& \nabla_{x_i} h^{\mathcal{S}}_{ij}(\nabla_{x_i}\Phi^T_{f_{\pi_i}} f_i(x_i,u_i) - (\partial \Phi^T_{f_{\pi_i}}/\partial t)|_{t=T}) \\
& + \nabla_{x_j} h^{\mathcal{S}}_{ij}(\nabla_{x_j}\Phi^T_{f_{\pi_j}} f_i(x_j,u_j) - (\partial \Phi^t_{f_{\pi_j}}/\partial t)|_{t=T}) \\
& + \alpha(h^{\mathcal{S}}_{ij}(\Phi^T_{f_{\pi_i}}(x_i), \Phi^T_{f_{\pi_j}}(x_j))) \geq 0.
\end{aligned} \tag{9}$$

By an argument similar to the proof of Proposition 1, if all agents follow their backup strategies, (9) is feasible whenever $h(x) \geq 0$. However, due to the coupling constraints between agents, this CBF QP cannot be solved decentrally. In particular, for each $\dot{h}^{\mathcal{C}}_{ij}$ and $\dot{h}^{\mathcal{S}}_{ij}$, the derivatives contain two parts, one determined by $\dot{x}_i$ and one by $\dot{x}_j$. To resolve this problem, notice that the terms containing u_i and u_j are summed together. Therefore, with a decomposition of the CBF derivative, the CBF QP with a sufficient condition of (9) can be solved decentrally. For the agent i, the following CBF QP is solved:

$$\begin{aligned}
u^\star = \arg\min_{u_i \in \mathcal{U}_i}\ & \|u_i - u^0_i\|^2 \\
\text{s.t.}\ & \forall t \in [0,T],\ \nabla h^{\mathcal{C}}_i(\nabla_{x_i}\Phi^t_{f_{\pi_i}} f_i(x_i,u_i) - \partial \Phi^t_{f_{\pi_i}}/\partial t) \\
& + \alpha(h^{\mathcal{C}}_i \Phi^t_{f_{\pi_i}}(x_i))) \geq 0, \\
& \forall j \neq i, \nabla_{x_i} h^{\mathcal{C}}_{ij}(\nabla_{x_i}\Phi^t_{f_{\pi_i}} f_i(x_i,u_i) - \partial \Phi^t_{f_{\pi_i}}/\partial t) \\
& + 0.5\alpha(h^{\mathcal{C}}_{ij}(\Phi^t_{f_{\pi_i}}(x_i), \Phi^t_{f_{\pi_j}}(x_j)) \geq 0, \\
& \nabla h^{\mathcal{S}}_i(\nabla_{x_i}\Phi^T_{f_{\pi_i}} f_i(x_i,u_i) - (\partial \Phi^t_{f_{\pi_i}}/\partial t)|_{t=T})
\end{aligned}$$

$$+ \alpha(h_i^{\mathcal{S}}\Phi_{f_{\pi_i}}^T(x_i))) \geq 0,$$
$$\nabla_{x_i} h_{ij}^{\mathcal{S}}(\nabla_{x_i}\Phi_{f_{\pi_i}}^T f_i(x_i, u_i) - (\partial\Phi_{f_{\pi_i}}^T/\partial t)|_{t=T})$$
$$+ 0.5\alpha(h_{ij}^{\mathcal{S}}(\Phi_{f_{\pi_i}}^T(x_i), \Phi_{f_{\pi_j}}^T(x_j))) \geq 0. \quad (10)$$

where u_i^0 is the desired input for agent i from the legacy controller. Note that this optimization only depends on information of agent i and $\Phi_{\pi_j}^t(x_j)$, i.e., the backup trajectories of other agents. Since we assume that each agent can measure the state of other agents, if π_j is known a priori, $\Phi_{\pi_j}^t(x_j)$ can be solved by agent i by a simple integration scheme.

Theorem 1: For all $x \in \{x|h(x) \geq 0\}$, (10) is always feasible for every agent; and when each agent implement the supervisory controller in (10), the solution $[u_1^\star; \cdots u_N^\star]$ is a feasible solution to (9) (not necessarily the optimal solution).

Proof: The feasibility comes from the fact that $u_i = \pi_i(x_i)$ is a feasible solution. Given $[u_1^\star; \cdots u_N^\star]$ as the solutions to (10) for each agent, for each $i \neq j$, $\forall t \in [0, T]$, we have

$$\begin{aligned}\frac{dh_{ij}^{\mathcal{C}}(\Phi_{f_{\pi_i}}^t(x_i), \Phi_{f_{\pi_j}}^t(x_j))}{dt} &= \nabla_{x_i} h_{ij}^{\mathcal{C}}(\nabla_{x_i}\Phi_{f_{\pi_i}}^t f_i(x_i, u_i) - \partial\Phi_{f_{\pi_i}}^t/\partial t) \\ &\quad + \nabla_{x_j} h_{ij}^{\mathcal{C}}(\nabla_{x_j}\Phi_{f_{\pi_j}}^t f_i(x_j, u_j) - \partial\Phi_{f_{\pi_j}}^t/\partial t) \\ &\geq -0.5\alpha(h_{ij}^{\mathcal{C}}(\Phi_{f_{\pi_i}}^t(x_i), \Phi_{f_{\pi_j}}^t(x_j)) \times 2 \\ &= -\alpha(h_{ij}^{\mathcal{C}}(\Phi_{f_{\pi_i}}^t(x_i), \Phi_{f_{\pi_j}}^t(x_j)) \geq 0.\end{aligned}$$

The same is true for the constraints on $h_{ij}^{\mathcal{S}}$, therefore $[u_1^\star; \cdots u_N^\star]$ is a feasible solution to (9). ■

To conclude, the proposed decentralized CBF supervisory controller begins by selecting a backup strategy for each agent in the system that brings the agent to a stable equilibrium point under the backup strategy. The backup strategies for all agents are known a priori to every agent as part of the centralized design. Then each agent measures the state of the adjacent agents (agents that are far away do not pose any danger of collision) and makes sure that if other agents execute the backup strategy, its own backup strategy would avoid collision with both the static and other agents. This is achieved by every agent solving (10) decentrally. We show that the decentralized CBF QP is always feasible when the CBF $h(x) \geq 0$. Furthermore, since the computation only depends on the states of adjacent agents, whose number is bounded (due to the clearance requirement), the algorithm can scale to an arbitrary number of agents.

IV. CBF With Multiple Backup Strategies

The strategy in Section III guarantees obstacle avoidance for a multi-agent system, but the mobility of the system may be compromised for safety. Since the CBF intervention is based on the backup strategy, one natural way to increase mobility is to equip the agents with multiple backup strategies and the CBF condition only need to hold for one of the backup strategies. However, we show that this is not always implementable, especially in the cases without communication. We present a simple broadcast scheme that enables the implementation of CBF controllers with multiple backup strategies for each agent.

Let m_i denote the number of backup strategies for agent i and let π_i^k denote the k-th backup strategy for agent i. Given x_i and x_j, we say that π_i^k and π_j^l are two compatible backup strategies for agent i and j if $\forall t \in [0, T]$,

$$h_i^{\mathcal{C}}(\Phi_{\pi_i^k}^t(x_i)) \geq 0,\ h_j^{\mathcal{C}}(\Phi_{\pi_j^l}^t(x_j)) \geq 0,\ h_{ij}^{\mathcal{C}}(\Phi_{\pi_i^k}^t(x_i), \Phi_{\pi_j^l}^t(x_j)) \geq 0,$$
$$h_i^{\mathcal{S}}(\Phi_{\pi_i^k}^T(x_i)) \geq 0,\ h_j^{\mathcal{S}}(\Phi_{\pi_j^l}^T(x_j)) \geq 0,\ h_{ij}^{\mathcal{S}}(\Phi_{\pi_i^k}^T(x_i), \Phi_{\pi_j^l}^T(x_j)) \geq 0,$$

that is, if the backup trajectories of agent i and j under π_i^k and π_j^l satisfy the state constraint and terminal constraint.

With multiple backup strategies for each agent, a choice of backup strategies for the whole multi-agent system $\{\pi_i^{k_i}\}_{i=1}^N$ is a feasible backup strategy if for each agent pair i and j, $\pi_i^{k_i}$ and $\pi_j^{k_j}$ are compatible.

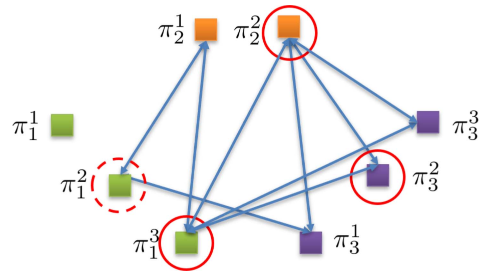

Fig. 1. Compatibility of multiple backup strategies.

Unfortunately, decentralized CBF with multiple backup strategies without communication between agents is in general not implementable. Consider the situation depicted in Fig. 1 consisting of 3 agents with 3, 2, and 3 backup strategies. Each line indicates that the two backup strategies it links are compatible. For the whole system, $(\pi_1^3, \pi_2^2, \pi_3^2)$ and $(\pi_1^3, \pi_2^2, \pi_3^3)$ are the two feasible backup strategies. However, agent 1 would not be able to tell that π_1^2 is not a valid choice without the information about the compatibility between the backup strategies of agent 2 and 3.

To resolve this problem, we propose a broadcast scheme in which each agent broadcast its currently selected backup strategy and use the information about other agents' selected backup strategies to determine whether it can change its current backup strategy. For example, in the situation depicted in Fig. 1, if the circled backup strategies are selected and broadcasted by the agents, agent 3 can switch from π_3^2 to π_3^3 since it is able to determine that π_3^3 is also compatible with the backup strategies selected by other agents.

Let π_i^s denote the selected backup strategy for agent i, and based on the selected backup strategies of other agents, agent i is able to determine the set of all backup strategies that is compatible with other agents in the system, we denote this set as Π_i^a, the active backup strategy set for agent i.

The CBF QP is solved for every backup strategy in Π_i^a and the one with the smallest intervention is selected and broadcasted to other agents in the system. The input corresponding to the selected backup strategy is then taken as the input u_i of agent i, shown in Algorithm 1.

V. Simulation and Experimental Result

In this section, we present the application of the proposed algorithm on a Dubin's car example and a quadrotor example.

A. Dubin's car

We consider a simple Dubin's car example with the following dynamics:

$$\begin{bmatrix}\dot{X} & \dot{Y} & \dot{v} & \dot{\theta}\end{bmatrix}^\intercal = \begin{bmatrix}v\cos(\theta) & v\sin(\theta) & a & r\end{bmatrix}^\intercal, \quad (11)$$

Algorithm 1 CBF QP With Multiple Backup Strategies

1: **procedure** CBF-QP($\pi^s_{1:N}$, u^0_i, $\mathcal{U}_i$)
2: Compute Π^a_i, set of all compatible backup strategies with $\pi^s_{1:N}$
3: **for** π^k_i in Π^a_i **do**
4: Solve (10) with π^k_i and u^0_i, and obtain $u^\star_{i,\pi^k_i}$
5: **end for**
6: $u_i = \min_{\pi^k_i \in \Pi^a_i} u^\star_{i,\pi^k_i}$
7: $\pi^s_i = \arg\min_{\pi^k_i \in \Pi^a_i} u^\star_{i,\pi^k_i}$
8: **end procedure**

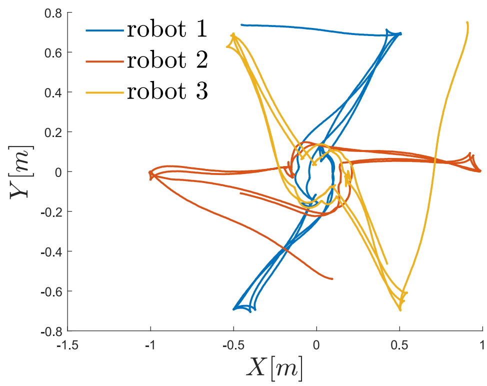

Fig. 2. Robot traces of the Robotarium experiment with 3 robots. Each robot is asked to patrol between 2 positions. The CBF controller guarantees zero collision.

where X, Y, v, θ are the longitudinal and lateral coordinates, the velocity, and the heading angle. The inputs are acceleration a and yaw rate r, and are bounded by $a^{\max}$ and $r^{\max}$. This is a valid model for differentially driven ground robots such as the ones in the Robotarium of Georgia Institute of Technology [27]. We first consider the simple case with only one backup strategy for each robot. In this case, the backup strategy is simply to brake until full stop. In practice, the expression of the CBF in (10) is not implementable since there are uncountably many t in $[0, T]$. We replace the continuous spectrum $[0, T]$ with a finite time sequence $0 = t_0 < t_1 <, \ldots, < t_M = T$ and enforce the CBF condition on these time instances instead. This finite sampling and the finite update rate of the CBF controller call for additional robustness of the control strategy. We proved robust safety under time discretization and the finite sampling of the backup trajectory in [28], check the result therein for detail. The backup trajectory and the sensitivity matrices are computed by solving the corresponding ODEs.

We conduct experiment in the Robotarium environment with 3 and 6 robots and the goal for each robot is to patrol between two way points. The legacy controller u^0 is a simple greedy linear controller that tries to bring the robot to the destination without any knowledge of other robots. The CBF keeps the robot within the boundary (static state constraints) and avoids any collision with other robots in the system.

Fig. 2 shows the traces of the 3 robots where they meet at the center of the state space and rotate to make ways for each other until they can move towards their destinations. Note that the seemingly coordinated behavior is actually the result of a decentralized control structure. With the same setup, we conducted experiment with 6 robots in the Robotarium, and the result shows that the CBF is able to guarantee no collision between the robots. The video of the experiments and the simulations can be found at https://youtu.be/RqsCvHBjf88. The differences between this experiment and the control barrier function approach in [29] are (1) the CBF proposed in this letter uses acceleration rather than speed as control input and guarantees feasibility of the CBF QP under torque limit (2) the proposed CBF approach is implemented decentrally where each robot solves for the safe input without communication with other robots.

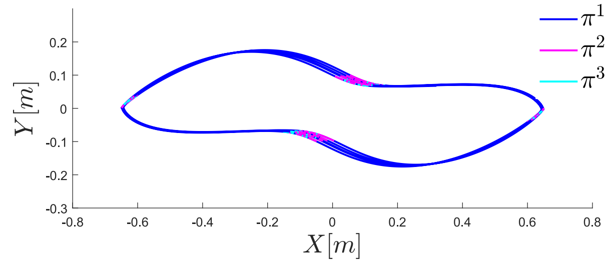

Fig. 3. Traces of 2 robots equipped with 3 backup strategies and broadcasting their current backup strategy. Colors shows the selected backup strategy in the CBF QP.

We also tested the case in which each robot is equipped with 3 backup strategies, where π^1 is to simply break, π^2 is to break and turn right, and π^3 is to break and turn left. Fig. 3 shows the state trajectories of 2 robots performing a similar surveillance task controlled under the CBF with 3 backup strategies and the broadcasting scheme as discussed in Section IV. Different colors were used to mark the segments of the trajectory during which different backup strategies were chosen. When the two robots swerve and avoid each other, π^2 and π^3 are selected so that the intervention needed is minimized.

B. Quadrotor

To showcase the scalability of this method, we now consider a 17-dimensional quadrotor model. The state vector $x = [\mathbf{r}, \mathbf{v}, \mathbf{q}, \mathbf{w}, \Omega]^\intercal$ where $\mathbf{r}$ is the position $[x, y, z]^\intercal$ in $\mathbb{R}^3$, $\mathbf{v}$ is the velocity $[v_x, v_y, v_z]^\intercal$ in the world frame, $\mathbf{q}$ is the quaternion $[q_w, q_x, q_y, q_z]^\intercal$, $\mathbf{w}$ is the angular velocity vector $[w_x, w_y, w_z]^\intercal$ in the body frame, and Ω is the vector of angular velocities of the propellers, $[\Omega_1, \Omega_2, \Omega_3, \Omega_4]^\intercal$. The control input is the voltages applied at the motors $u = [V_1, V_2, V_3, V_4]^\intercal$.

The dynamics are derived from force-balance equations in a rotating frame, as well as a first order motor model. The gradients of the dynamics w.r.t. the state are computed symbolically. The resulting symbolic expressions for $f(x)$ and $\frac{\partial f(x,u)}{\partial x}$, and $\frac{\partial f(x,u)}{\partial u}$ are then exported to C++ using code generation from MATLAB for the simulation environment.

The backup policy aims to simply stop the quadrotor and set the pitch and roll angles to zero, which is achieved through simple PD controllers around the linear velocities, angular rates, and angles. The gradients of these dynamics with respect to the state are also computed symbolically and exported to C++. The corresponding backup set is a small ball around linear velocity, pitch, and roll angles equal to 0.

For each quadrotor, the barrier function $h(x)$ seeks to avoid a ball of radius r around the closest point on the other

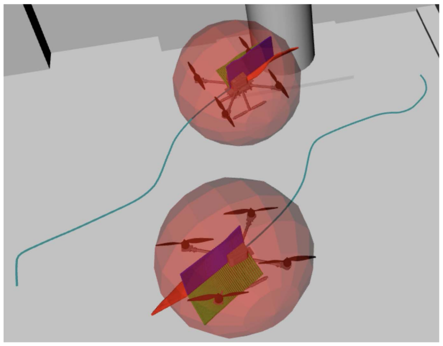

Fig. 4. Swerving maneuvers of the drones under the CBF controller when commanded to fly at each other.

quadrotor's backup trajectory ($[x_c, y_c, z_c]^\intercal$), giving

$$h(x) = (x - x_c)^2 + (x - x_c)^2 + (x - x_c)^2 - r^2$$

The final expression needed is the gradient of this function ∇h, which can be derived trivially.

The simulation environment is based on ROS and is written in C++. The code can be found at https://github.com/DrewSingletary/uav_sim_ros, including the MATLAB dynamics and C++ code generation. The solver used was the OSQP solver [30], and the nominal controller tracked linear velocity and yaw rate commands using the same PD control strategy as the backup controller. Fig. 4 shows a snapshot of the simulation, when the two drones are sent directly at each other.

VI. Conclusion

We present a backup strategy-based control barrier function approach that is able to guarantee obstacle avoidance for multiple agents with limited actuation capacity. The idea is to let every agent share a centrally designed contract that can be implemented decentrally and guarantees obstacle avoidance with CBF QP. Furthermore, the single backup strategy case can be easily extended to multiple backup strategy case to improve mobility with a broadcasting scheme. We prove collision avoidance with guarantee, which is validated with experiments with wheeled robots and simulation with drones.

References

[1] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous Robot Vehicles*. New York, NY, USA: Springer, 1986, pp. 396–404.

[2] J. Borenstein and Y. Koren, "Histogramic in-motion mapping for mobile robot obstacle avoidance," *IEEE Trans. Robot. Autom.*, vol. 7, no. 4, pp. 535–539, Aug. 1991.

[3] D. Hsu, L. E. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin, "On finding narrow passages with probabilistic roadmap planners," in *Proc. 3rd Workshop Algorithmic Found. Robot. Robot. Algorithmic Perspect.*, 1998, pp. 141–154.

[4] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *Int. J. Robot. Res.*, vol. 21, no. 3, pp. 233–255, 2002.

[5] S. M. LaValle, J. J. Kuffner, B. Donald, "Rapidly-exploring random trees: Progress and prospects," in *Proc. Workshop Algorithmic Found. Robot. Algorithmic Comput. Robot. New Directions*, 2001, pp. 293–308.

[6] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *Proc. 49th IEEE Conf. Decis. Control (CDC)*, Atlanta, GA, USA, 2010, pp. 7681–7687.

[7] B. Lau, C. Sprunk, and W. Burgard, "Kinodynamic motion planning for mobile robots using splines," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, 2009, pp. 2427–2433.

[8] K. Sreenath, T. Lee, and V. Kumar, "Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load," in *Proc. 52nd IEEE Conf. Decis. Control*, Florence, Italy, 2013, pp. 2269–2274.

[9] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads," in *Proc. ASME Dyn. Syst. Control Conf.*, 2010, pp. 265–272.

[10] U. Rosolia, S. De Bruyne, and A. G. Alleyne, "Autonomous vehicle control: A nonconvex approach for obstacle avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 2, pp. 469–484, Mar. 2017.

[11] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton–Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Autom. control*, vol. 50, no. 7, pp. 947–957, Jul. 2005.

[12] Y. Chen, H. Peng, and J. Grizzle, "Obstacle avoidance for low-speed autonomous vehicles with barrier function," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 1, pp. 194–206, Jan. 2018.

[13] U. Borrmann, 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.

[14] Y. Chen, H. Peng, and J. W. Grizzle, "Validating noncooperative control designs through a Lyapunov approach," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 2, pp. 527–539, Mar. 2019.

[15] T. Gurriet, M. Mote, A. D. Ames, and E. Feron, "An online approach to active set invariance," in *Proc. IEEE Conf. Decis. Control (CDC)*, Miami Beach, FL, USA, 2018, pp. 3592–3599.

[16] E. Frazzoli, M. A. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1077–1091, Dec. 2005.

[17] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Proc. IEEE 53rd Annu. Conf. Decis. Control (CDC)*, Las Angeles, CA, USA, 2014, pp. 6271–6278.

[18] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

[19] S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne, "Invariant approximations of the minimal robust positively invariant set," *IEEE Trans. Autom. Control*, vol. 50, no. 3, pp. 406–410, Mar. 2005.

[20] I. Kolmanovsky and E. G. Gilbert, "Theory and computation of disturbance invariant sets for discrete-time linear systems," *Math. Problems Eng.*, vol. 4, no. 4, pp. 317–367, 1998.

[21] P. Trodden, "A one-step approach to computing a polytopic robust positively invariant set," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 4100–4105, Dec. 2016.

[22] Y. Chen, H. Peng, J. Grizzle, and N. Ozay, "Data-driven computation of minimal robust control invariant set," in *Proc. IEEE Conf. Decis. Control (CDC)*, Miami Beach, FL, USA, 2018, pp. 4052–4058.

[23] A. Papachristodoulou and S. Prajna, "A tutorial on sum of squares techniques for systems analysis," in *Proc. IEEE Amer. Control Conf.*, Portland, OR, USA, 2005, pp. 2686–2700.

[24] X. Xu, J. W. Grizzle, P. Tabuada, and A. D. Ames, "Correctness guarantees for the composition of lane keeping and adaptive cruise control," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 3, pp. 1216–1229, Jul. 2018.

[25] A. Singletary, P. Nilsson, T. Gurriet, and A. D. Ames, "Online active safety for robotic manipulators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Macau, China, 2019, pp. 173–178.

[26] H. Seywald and R. Kumar, "Desensitized optimal trajectories," in *Analytical Mechanics Associates Rept*. New York, NY, USA: Distrib. Plenum Press, 2003, pp. 03–16.

[27] D. Pickem *et al.*, "The robotarium: A remotely accessible swarm robotics research testbed," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Singapore, 2017, pp. 1699–1706.

[28] A. Singletary, Y. Chen, and A. D. Ames, "Control barrier functions for sampled-data systems with input delays," May 2020. [Online]. Available: https://arxiv.org/abs/2005.06418.

[29] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Trans. Robot.*, vol. 33, no. 3, pp. 661–674, Jun. 2017.

[30] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," Nov. 2017. [Online]. Available: https://arxiv.org/abs/1711.08013.