

## RESEARCH ARTICLE

# Directed Graph Topology Preservation in Multi-Robot Systems With Limited Field of View Using Control Barrier Functions

FILIPPO BERTONCELLI<sup>1</sup>, (Member, IEEE), VIVEK RADHAKRISHNAN<sup>2,3</sup>,  
MATTIA CAPELLANI<sup>1</sup>, (Graduate Student Member, IEEE), GIUSEPPE LOIANNI<sup>2</sup>, (Member, IEEE),  
AND LORENZO SABATTINI<sup>1</sup>, (Senior Member, IEEE)

<sup>1</sup>Department of Sciences and Methods of Engineering, University of Modena and Reggio Emilia, 42122 Modena, Italy

<sup>2</sup>NYU Tandon School of Engineering, New York University, Brooklyn, NY 11201, USA

<sup>3</sup>Autonomous Robotics Research Center-Technology Innovation Institute, Abu Dhabi, United Arab Emirates

Corresponding author: Filippo Bertongcelli (filippo.bertoncelli@unimore.it)

This work was supported by the Technology Innovation Institute.

**ABSTRACT** This paper addresses the crucial challenge of maintaining the directed graph topology in multi-robot systems, particularly when operating under limited field-of-view constraints and with a lack of communication among robots. Traditional methods for multi-robot coordination rely heavily on inter-robot communication, which may not always be feasible, particularly in constrained or hostile environments. Our work presents a novel distributed control algorithm that leverages Control Barrier Functions (CBFs) to maintain the graph topology of a multi-robot system based solely on local, onboard sensor data. This approach is particularly beneficial in situations where external communication channels are disrupted or unavailable. The key contributions of this research are threefold: First, we design a novel control algorithm that efficiently maintains the graph topology in multi-robot systems using CBFs, which operate on neighbor detection data. Second, we perform an experimental evaluation of the algorithm, demonstrating its efficacy in controlling the flight of a team of drones using only local robot data. Third, we apply our methodology to a distributed coverage control scenario, showing that our approach can effectively manage a multi-robot system using only local information.

**INDEX TERMS** Multi-robot systems, topology, optimal control, distributed robot systems.

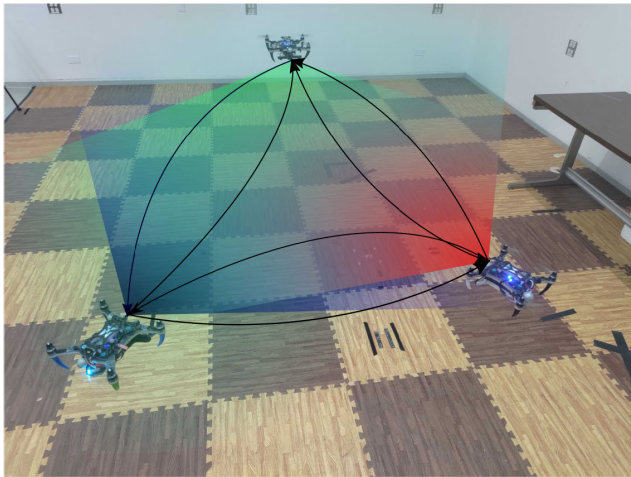
## I. INTRODUCTION

The research interest in multi-robot systems has surged in recent years due to the multiple advantages that teams of robots can provide in several missions. The usage of multiple agents can guarantee resilience to single points of failure and can potentially increase the speed at which tasks are executed compared to single robot equivalents, through cooperation and coordination. During the last two decades there has been significant progress in the study and development of control strategies to enable different means of coordination and cooperation among teams of robots [1]. Examples of tasks where swarms of robots are successfully applied and provide several advantages compared to the single robot

case are: coverage [2], whose aim is to maximize the area surveilled by the system; monitoring and exploration of the environment [3], [4], [5]; target tracking [6], [7] and collaborative transportation [8], [9]. The large majority of the distributed algorithms employed to coordinate swarms of robots rely on the ability of each agent in the swarm to obtain information about its surroundings. The transfer of information through communication channels can be modeled using a graph whose nodes represent the robots and the edges are the direct communication links. The correct performance of many distributed algorithms relies on the properties of the graph describing the communication network. Preservation of topological properties similar to the one depicted in Fig. 1, such as connectivity, has been studied in the literature, typically considering undirected (i.e., bi-directional) communication, and explicit communication

The associate editor coordinating the review of this manuscript and approving it for publication was Jianxiang Xi<sup>1</sup>.

among the robots [10], [11], [12], [13], [14]. However, in a scenario where the robots cannot communicate with each other, the conventional network graph modeling and its properties do not appropriately describe the system, nor provide performance guarantees. Each robot then has to rely on its onboard sensors to decide its future action. However, in scenarios where robots are incapacitated



**Figure 1.** Graph topology preservation according to the depicted arrows during a coverage control problem task with CBF enforcing visual constraint using aerial robots.

from inter-communication, the conventional network graph model becomes inadequate, failing to guarantee system performance. In such contexts, robots must depend solely on onboard sensors for decision-making. The effectiveness of distributed algorithms, even under these constraints, has been demonstrated, particularly when only relative positional information of neighbors is available. For instance, in coverage control, a robot may only need data from its immediate neighbors to compute its target.

### A. RELATED WORKS & CONTRIBUTIONS

Our work addresses the challenge of topology preservation in a fleet of drones governed by a distributed algorithm, specifically in scenarios where communication is suddenly interrupted. The drones must then independently maintain the requisite level of information for proper operation. While a variety of sensors, from IR to lidars and cameras, are available for environment scanning and neighbor detection, these often have anisotropic capabilities. The information level of a single drone can thus be represented by a directed graph. Maintaining this graph's topology (i.e., preserving all outgoing edges from a node) equates to sustaining the information level of each drone. Similar solutions for this problem have been presented in the literature, however, there are some key differences to be considered. In [15], the authors face the problem of maintaining the continuous tracking of a target using a fleet of drones. They propose a multi-objective distributed optimization problem that ensures connectivity maintenance of the fleet, obstacle avoidance,

and occlusion avoidance. To achieve these safety constraints, the authors employ a controller based on Control Barrier Functions (CBFs), that modifies the motion of the drones. However, since in their example drones can communicate and exchange information, the authors did not include the maintenance of the target inside the field of view in the optimization problem, allowing partial loss of visibility. Without the ability to communicate, drones not facing each other would crash into each other since they do not have accurate information on their surroundings. In [16], the authors propose a Barrier Function-based Image Servoing control that maintains detected features in the field of view. The controller enables the system to position itself with respect to a detected object while maintaining said object in the field of view to ensure proper feedback information is provided. This problem is very similar to the problem addressed in this paper but with a substantial difference: the target for the system, by definition, satisfies the constraints. In this paper, we treat the problem of achieving a general task that may have a setpoint that does not satisfy the constraints. The contribution of this paper can then be summarized as follows

- The design of a novel distributed control algorithm that ensures the graph topology maintenance for a multi-robot system, leveraging CBFs on neighbor detection to allow the continuation of the task at hand if possible.
- The experimental evaluation of the algorithm to control the flight of a team of drones, using only local robot data (i.e., no communication).
- The application of the proposed approach to coverage control using only local information.

### II. PRELIMINARIES

We denote by  $\mathbb{N}$ ,  $\mathbb{R}$ ,  $\mathbb{R}_{\geq 0}$ , and  $\mathbb{R}_{> 0}$  the set of natural, real, real non-negative, and real positive numbers, respectively. Given  $\mathbf{x} \in \mathbb{R}^n$ , let  $\|\mathbf{x}\|$  be the Euclidean norm.

Let  $\mathcal{G} = (\mathcal{U}, \mathcal{E})$  be a graph characterized by a set  $\mathcal{U}$  of vertices and a set  $\mathcal{E} \subseteq \mathcal{U} \times \mathcal{U}$  of directed edges. Given an edge  $(i, j) \in \mathcal{E}$ , then the vertex  $j$  is a neighbor of the vertex  $i$ . Let  $\mathcal{N}_i \in \mathbb{N}$  be the set of neighbors of the vertex  $i$  in  $\mathcal{G}$ . A graph  $\mathcal{G}$  is said to be *undirected* if  $(i, j) \in \mathcal{E}$  implies  $(j, i) \in \mathcal{E}$ .

Let  $\mathbb{F}(\mathbb{R}^2)$  be the collection of finite point sets in  $\mathbb{R}^2$ . We denote an element of  $\mathbb{F}(\mathbb{R}^2)$  as  $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_n\} \subset \mathbb{R}^2$ , where  $\{\mathbf{p}_1, \dots, \mathbf{p}_n\}$  are points in  $\mathbb{R}^2$ . Let  $Q \subset \mathbb{R}^2$  be a generic polygon representing the operational space in the environment. An arbitrary point in  $Q$  is denoted by  $\mathbf{q} \in Q$ . Let us define the *proximity graph* as a graph  $\mathcal{G}_P$  in which the edge set  $\mathcal{E}_{\mathcal{G}_P}$  depends on the location of the vertices. In this paper, we consider graphs defined for points  $\{\mathbf{p}_1, \dots, \mathbf{p}_n\}$  in  $\mathbb{R}^2$ . Hence, we can define a proximity graph function  $G : \mathbb{F}(\mathbb{R}^2) \rightarrow (\mathcal{U}, \mathcal{E})$  that associates to  $\mathcal{P} \in \mathbb{F}(\mathbb{R}^2)$  an undirected graph with the set  $\mathcal{P}$  of vertices and the set  $\mathcal{E}_{\mathcal{G}}(\mathcal{P})$  of edges, where  $\mathcal{E}_{\mathcal{G}} : \mathbb{F}(\mathbb{R}^2) \rightarrow \mathbb{F}(\mathbb{R}^2 \times \mathbb{R}^2)$  has the property that  $\mathcal{E}_{\mathcal{G}}(\mathcal{P}) \subseteq \mathcal{P} \times \mathcal{P} \setminus \text{diag}(\mathcal{P} \times \mathcal{P})$ .

We denote, for  $\mathbf{p} \in \mathbb{R}^2$  and  $r \in \mathbb{R}_{> 0}$ , the closed and open ball in  $\mathbb{R}^2$  centered at  $\mathbf{p}$  with radius  $r$  with

$\bar{B}(\mathbf{p}, r) = \{\mathbf{q} \in Q \mid \|\mathbf{q} - \mathbf{p}\| \leq r\}$  and  $B(\mathbf{p}, r) = \{\mathbf{q} \in Q \mid \|\mathbf{q} - \mathbf{p}\| < r\}$ , respectively. The set of edges of the proximity graph is defined as  $\mathcal{E}_{\mathcal{G}_p} = \{(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{P} \times \mathcal{P} \mid \mathbf{p}_j \in \bar{B}(\mathbf{p}_i, r)\}$ . We consider robots moving in a closed environment free of obstacles. The inertial reference frame associated with the environment is denoted with  $\Sigma_w$ . The local reference frame attached to the body of each robot  $i$  in the environment is denoted with  $\Sigma_i$ . Figure 2 provides a visualization of the considered reference frames. The notation  ${}^w\Sigma_i$  denotes the relative transformation from the global reference frame  $\Sigma_w$  to the local frame  $\Sigma_i$  expressed in the global frame. Similarly, the notation  ${}^i\mathbf{x}$  indicates a generic vector  $\mathbf{x}$  expressed relatively to  $\Sigma_i$ . If not specified, a vector is expressed with respect to  $\Sigma_w$ . The notation  ${}^wR_i \in SO2$  is used to denote the rotation matrix from  $\Sigma_w$  to  $\Sigma_i$ , such that  ${}^w\mathbf{x} = {}^wR_i {}^i\mathbf{x}$ .

Following CBF theory [17], let  $\mathcal{C} = \{\mathbf{p} \in \mathbb{R}^m : h(\mathbf{p}) \geq 0\}$  be the set of configurations that satisfy the safety requirements for the system, also known as the safe set  $\mathcal{C}$  which is defined as the super level set of a smooth function  $h : \mathbb{R}^m \rightarrow \mathbb{R}$  with  $\frac{\partial h}{\partial \mathbf{p}}(\mathbf{p}) \neq 0 \forall \mathbf{p} \in \partial \mathcal{C}$ . According to Nagumo's theorem [18], with the extension provided in [19] and [20], CBFs are defined by the following relation:

$$\exists \mathbf{u} \text{ s.t. } \dot{h}(\mathbf{p}, \mathbf{u}) \geq -\alpha(h(\mathbf{p})) \iff \mathcal{C} \text{ is invariant,}$$

where  $\alpha$  is an extended class  $\mathcal{K}$  function. With this condition it is possible to synthesize optimization-based controllers that modify the desired control input  $\mathbf{u}^*$  in a minimally invasive fashion, solving the following optimization problem:

$$\begin{aligned} \mathbf{u}(\mathbf{p}) &= \arg \min_{\mathbf{u} \in \mathbb{R}^m} \frac{1}{2} \|\mathbf{u} - \mathbf{u}^*\|^2 \\ \text{s.t. } \dot{h}(\mathbf{p}, \mathbf{u}) &\geq -\alpha(h(\mathbf{p})). \end{aligned} \quad (1)$$

This controller aims therefore at providing the control input  $\mathbf{u}$  closest to the desired input  $\mathbf{u}^*$  that satisfies the constraints.

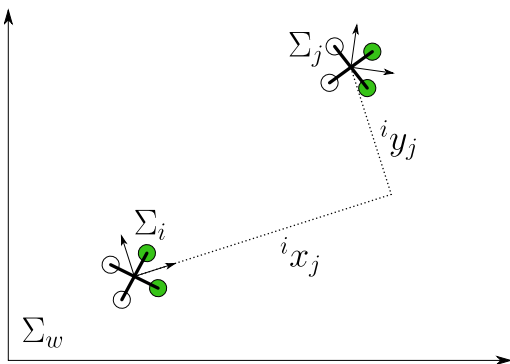


Figure 2. Visualization of the system and relative coordinates.

### III. PROBLEM DESCRIPTION

Consider a team of  $n$  aerial robots flying in an obstacle-free bounded environment. The aerial robots fly using only information from onboard sensors. Each drone knows its

global position at takeoff and estimates the pose of its local reference frame  ${}^w\Sigma_i$  through odometry information. In the following, we assume that all the drones fly at the same altitude and move with a bounded planar velocity. Furthermore, each drone can detect the neighbors that are in the field of view of its camera. The set of detected neighbors for each robot  $i$  is denoted as  $\mathcal{N}_i \in \mathbb{N}$ . The relative 2D position of the neighbour  $j$  with respect to  $\Sigma_i$  is expressed as  ${}^i\mathbf{p}_j = [{}^i x_j, {}^i y_j]^T$ . Let  $\mathcal{G}$  be the directed graph describing the connection between the aerial robots. Specifically, each aerial robot is represented in  $\mathcal{G}$  as a vertex, while an edge  $(i, j)$  is in the graph if and only if  $j$  is detected by  $i$ , formally expressed as  $(i, j) \in \mathcal{E}_{\mathcal{G}} \iff j \in \mathcal{N}_i$ . A visualized example of the graph is provided in Fig. 4. The state of each aerial robot  $i$  is denoted with  ${}^w\chi_i = [{}^w x_i \ {}^w y_i \ {}^w \theta_i]^T$ , where  ${}^w x_i, {}^w y_i \in \mathbb{R}$  represent the coordinates of the 2D position of  $\Sigma_i$  in  $\Sigma_w$ , while  ${}^w \theta_i$  is the yaw angle. The altitude is excluded since we assume every robot flies at the same constant altitude. The aerial robot is controlled to obtain the following kinematic model

$${}^w\dot{\chi}_i = {}^wR_i {}^i\mathbf{u}_i = {}^wR_i \begin{bmatrix} {}^i v_x \\ {}^i v_y \\ {}^i \omega \end{bmatrix}, \quad (2)$$

where  ${}^i v_x, {}^i v_y, \omega \in \mathbb{R}$  represent the linear and angular velocities of the drone in  $\Sigma_i$ . Furthermore, each aerial robot has a limited angular sector in which it can detect neighboring drones. This angular sector is centered at the local X axis with amplitude  $\beta$ , as visualized in Fig. 3.

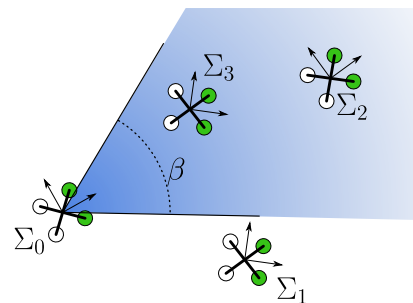
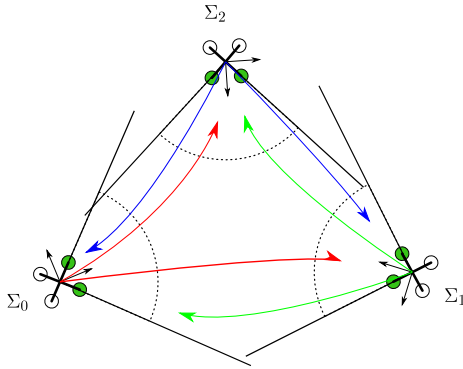


Figure 3. Visualization of the field of view of drone 0. In particular, only drone 3 and 2 are detected since 1 is on the right of the visible space.

The drones in this configuration are instructed to perform a decentralized task. Assume, without loss of generality, that the task requires each drone to know the relative position of its neighbors as well as its own position in the environment. In ideal conditions, this information is usually obtained through wireless communication channels, onboard sensors, or a combination of both. However, in the analyzed scenario, the communications are externally interrupted, therefore the drones are required to operate only with data from onboard sensors. As a conservative measure in this situation, the robots should behave to maintain the current level of operation, therefore we propose to solve the following problem:

*Problem 1:* Maintain the topology of the directed graph  $\mathcal{G}$  describing the system, to provide each robotic agent with



**Figure 4.** Visualization of the graph describing the system. Each drone has all the others inside the field of view, therefore each drone has information on the whole system.

the necessary information to allow the safe execution of the global task assigned to the team, despite the absence of communication between the agents. While avoiding collisions with other agents.

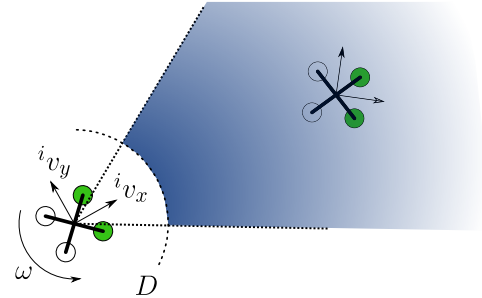
The problem presented can be divided into three core aspects:

- Maintaining the topology of the directed graph  $\mathcal{G}$ : each drone should have the necessary information to execute the assigned (external) task.
- Enforce safe execution: each drone should avoid crashing with other drones.
- Execute the global task: the controller should minimally modify the behavior of the drones such that the constraints are enforced while the objective of the task is being pursued.

### A. PROPOSED SOLUTION

Given the considered limitations on the communication between the agents, the problem of maintaining the topology is equivalent to maintaining the current connections between the robots. A connection between the robots depends only on the relative pose of the robots with respect to each other, therefore to maintain a connection each robot should move in order to maintain the currently detected neighbors inside its field of view. We, therefore, propose a CBF-based controller that minimally modifies a desired input velocity to maintain the currently detected robots in the field of view. The CBF is defined starting from the relative position of the detected robots with two separated components, one from the left side of the field of view and one from the right side. Furthermore, to increase the safety of the motion, a third component is added to guarantee the maintenance of a safe distance from the neighbors. The three components are treated as different constraints inserted into an optimization solver. Figure 5 displays the area where a robot should keep its neighbors in order to avoid losing the connection and avoid crashing. From the premises, the problem of maintaining the topology of the graph is distributed since each drone is responsible for maintaining its outward edges. If a single drone does not adhere to the behavior, even by simply rotating in place  $180^\circ$ , it can change the topology of the graph removing

some edges. Therefore each drone is equally responsible for maintaining the topology of the system, implying that the proposed solution solves Problem 1 in a distributed fashion, moreover the CBF-constrained controller is formulated as a Quadratic Programming problem with linear constraints, whose solution can be obtained in milliseconds.



**Figure 5.** Depiction of the safe space where robot  $i$  aims to keep the detected drones.

### IV. FIELD OF VIEW MAINTENANCE

In this section, we detail the formulation of the motion constraint for each drone, in order to maintain its detected neighbors in the camera field of view. In order to maintain the drone  $j$  in the field of view of drone  $i$ , we propose the addition of a CBF to optimize the velocity input to ensure the feasibility of the constraint. For the problem at hand, we propose the use of a CBF to guarantee the constraint is always satisfied during the motion of the drones. For this purpose, we introduce, for each drone  $i$ , the following  $h(\cdot)$  function, defined for each neighbour  $j \in \mathcal{N}_i$

$$h(\mathbf{p}_j) = \begin{bmatrix} h_1(\mathbf{p}_j) \\ h_2(\mathbf{p}_j) \\ h_3(\mathbf{p}_j) \end{bmatrix} = \begin{bmatrix} \tan(\beta/2)^i x_j + y_j \\ \tan(\beta/2)^i x_j - y_j \\ \|\mathbf{p}_j\|^2 - D^2 \end{bmatrix}. \quad (3)$$

The first two components  $h_1(\cdot)$  and  $h_2(\cdot)$  express the left and right border of the field of view as a linear function. The third component expresses the difference between the distance of drone  $j$  from drone  $i$  with the safe distance  $D \in \mathbb{R}_{\geq 0}$ . Imposing the condition  $h(\cdot) \geq \mathbf{0}$  is equivalent to constraining drone  $j$  to be: 1) on the left of the right border, 2) on the right of the left border, 3) at a safe distance from  $i$ . This component is equivalent to the CBF proposed in [21], we refer interested readers to the original paper for further details. The associated optimization problem is formulated as follows. Let  $\mathbf{u}_i = [v_x, v_y, \omega]^T \in \mathbb{R}^3$  be the vector of input velocities for the  $i$ -th drone expressed in its body frame. Let  $\mathbf{u}_i^*$  be the desired velocity for the drone obtained from a top-level distributed algorithm, i.e. the task controller. The following optimization-based controller is employed to ensure the safety condition while modifying the desired velocity in a minimally invasive fashion as

$$\begin{aligned} & \arg \min_{\mathbf{u}_i \in \mathbb{R}^m} (\mathbf{u}_i - \mathbf{u}_i^*)^T H (\mathbf{u}_i - \mathbf{u}_i^*) \\ & \text{s.t. } \dot{h}(\mathbf{p}_j, \mathbf{u}_i) \geq -\alpha(h(\mathbf{p}_j)), \quad \forall j \in \mathcal{N}_i. \end{aligned} \quad (4)$$

The cost function of the optimization is expressed as a quadratic function of the error between the input  ${}^i\mathbf{u}_i$  and the desired control action  ${}^i\mathbf{u}_i^*$ . The positive definite matrix  $H \in \mathbb{R}^{3 \times 3}$  is inserted in order to assign different weights to specific components of the velocity error. We define the class  $\mathcal{K}$  function  $\alpha(\cdot)$  as the function

$$\alpha(h({}^i\mathbf{p}_j)) = \begin{bmatrix} \gamma_F h_1({}^i\mathbf{p}_j)^3 \\ \gamma_F h_2({}^i\mathbf{p}_j)^3 \\ \gamma_S h_3({}^i\mathbf{p}_j) \end{bmatrix}, \quad (5)$$

where  $\gamma_F, \gamma_S \in \mathbb{R}_{\geq 0}$  are positive constants. The use of such  $\alpha(\cdot)$  function is motivated by the necessity of slowing the relative motion when the robot approaches the border of the field of view from the inside. The cubic function possesses the same requirements as the linear function but its values are smaller when  $h(\cdot)$  is close to zero, with respect to the linear function used for example in  $h_3$ . This difference in shape imposes that  $\dot{h}$  has to be “less decreasing” when approaching the border with respect to the linear counterpart. This effect dampens the behavior of  $h$  allowing for smoother behavior.

The time derivative  $\dot{h}(\cdot)$  is obtained as follows. The function  $h({}^i\mathbf{p}_j)$  introduced in eq. (3) can be rewritten as

$$h({}^i\mathbf{p}_j) = \begin{bmatrix} \tan(\beta/2) & 1 \\ \tan(\beta/2) & -1 \\ i x_j & i y_j \end{bmatrix} {}^i\mathbf{p}_j - \begin{bmatrix} 0 \\ 0 \\ D^2 \end{bmatrix}. \quad (6)$$

The time derivative of this formulation is then expressed as

$$\dot{h}({}^i\mathbf{p}_j, {}^i\mathbf{u}_i) = \frac{\partial h({}^i\mathbf{p}_j)}{\partial {}^i\mathbf{p}_j} {}^i\dot{\mathbf{p}}_j = \begin{bmatrix} \tan(\beta/2) & 1 \\ \tan(\beta/2) & -1 \\ -2^i x_j & -2^i y_j \end{bmatrix} {}^i\dot{\mathbf{p}}_j. \quad (7)$$

The velocity of  $j$  with respect to  $i$  is then expressed through kinematic computations

$${}^i\dot{\mathbf{p}}_j = {}^iR_w {}^w\dot{\mathbf{p}}_j - \begin{bmatrix} i v_x \\ i v_y \end{bmatrix} + \omega \begin{bmatrix} i y_j \\ -i x_j \end{bmatrix}. \quad (8)$$

Since we assume not to have information on the 2D velocity of the neighbor drones, as a conservative approach we assume each neighbor is moving towards the closest border at the maximum allowed velocity. It is then possible to write (7) as

$$\dot{h}({}^i\mathbf{p}_j, {}^i\mathbf{u}_i) = \frac{\partial h({}^i\mathbf{p}_j)}{\partial {}^i\mathbf{p}_j} {}^iR_w {}^w\dot{\mathbf{p}}_j + A_c {}^i\mathbf{u}_i \quad (9)$$

where

$$A_c = \begin{bmatrix} -\tan(\beta/2) & -1 & \tan(\beta/2)^i y_j - i x_j \\ -\tan(\beta/2) & 1 & \tan(\beta/2)^i y_j + i x_j \\ -2^i x_j & -2^i y_j & 0 \end{bmatrix}. \quad (10)$$

The optimization problem in (4) therefore becomes

$$\begin{aligned} & \arg \min_{{}^i\mathbf{u}_i \in \mathbb{R}^m} ({}^i\mathbf{u}_i - {}^i\mathbf{u}_i^*)^T H ({}^i\mathbf{u}_i - {}^i\mathbf{u}_i^*) \\ & \text{s.t. } A_c {}^i\mathbf{u}_i \geq -\alpha(h({}^i\mathbf{p}_j)) - \frac{\partial h({}^i\mathbf{p}_j)}{\partial {}^i\mathbf{p}_j} {}^iR_w {}^w\dot{\mathbf{p}}_j, \quad \forall j \in \mathcal{N}_i \end{aligned} \quad (11)$$

In this formulation of the optimization problem, the constraint is separated into terms that depend on the decision variable  ${}^i\mathbf{u}_i$

and terms that depend only on the current configuration. Such formulation facilitates the implementation of the Quadratic Programming problem into a solver to easily and quickly obtain the optimal solution.

Enforcing a neighbor drone  $j$  to remain in the field of view of  $i$  ensures that the existing outgoing edges of the associated graph  $\mathcal{G}$  are preserved.

## V. CASE STUDY: COVERAGE CONTROL

The topology maintenance approach proposed in this paper can be employed along many distributed multi-robot control algorithms. Among many possible options, such as [22], [23], [24], and [25], as a case study, we explore the effect of anisotropic neighbor detection on coverage control and how the proposed approach can guarantee the safe execution of the coverage task.

Briefly, coverage control [2] exploits Lloyd’s algorithm to reach the configuration of the system which maximizes the information that the swarm can obtain, indicated by a probability density function  $\phi(\cdot)$ . The environment is divided into cells following the Voronoi partitioning algorithm. Every cell  $V_i$  is calculated from every robot’s position  $\mathbf{p}_i \in \mathcal{P}$  and contains areas of the environment closer to the  $i$ -th robot than the others *and* are inside the sensing radius of the  $i$ -th robot. Each robot has a limited sensing range, therefore the Voronoi cells can be expressed according to the definition in [26]

$$V_i(\mathcal{P}) = \{\mathbf{q} \in \bar{B}(\mathbf{p}_i, r_{sens}) \mid \|\mathbf{q} - \mathbf{p}_i\| \leq \|\mathbf{q} - \mathbf{p}_j\|, \forall \mathbf{p}_j \in \mathcal{P}\}. \quad (12)$$

Subsequently, the centroid of each Voronoi cell  $C_{V_i}$  is calculated taking into account the probability density distribution  $\phi(\mathbf{q})$

$$C_{V_i} = \frac{\int_{V_i} \mathbf{q} \phi(\mathbf{q}) d\mathbf{q}}{\int_{V_i} \phi(\mathbf{q}) d\mathbf{q}}. \quad (13)$$

The probability function  $\phi(\mathbf{q})$  can be defined to highlight the higher importance of specific areas of the environment with respect to others. Finally, the desired input for the  $i$ -th robot is calculated to move toward the centroid of its cell according to the law

$$\mathbf{u}_i^* = -k_{prop}(\mathbf{p}_i - C_{V_i}) \quad (14)$$

where  $k_{prop} \in \mathbb{R}_{\geq 0}$  is a proportional gain. Such proportional gain dictates how fast the system converges to the configuration that optimizes the displacement of the drones in the space. Therefore it should be chosen considering both performances and dynamic limitations of the drones. It is worth noting that defining a non-uniform probability density  $\phi(\cdot)$  leads the agents to a desired area of the environment. It is also important to highlight that, according to (12) and (13), both the limited Voronoi partitioning and the centroid can be computed in a distributed manner, therefore no global knowledge is needed to calculate the control action. In fact, each robot can compute its own control input simply from information about its own location, the neighbors’

location, and the probability density of the information in the environment, as was extensively tested in [27]. The integration of the coverage action into the optimization problem is performed using the velocity vector  $\mathbf{u}_i^*$  calculated in (14) as the desired control input for the cost function in (4).

Coverage control is proven asymptotically stable for multi-robot systems where the graph is connected. A graph disconnecting introduces instability in the information flow and the algorithm is not guaranteed to be stable.

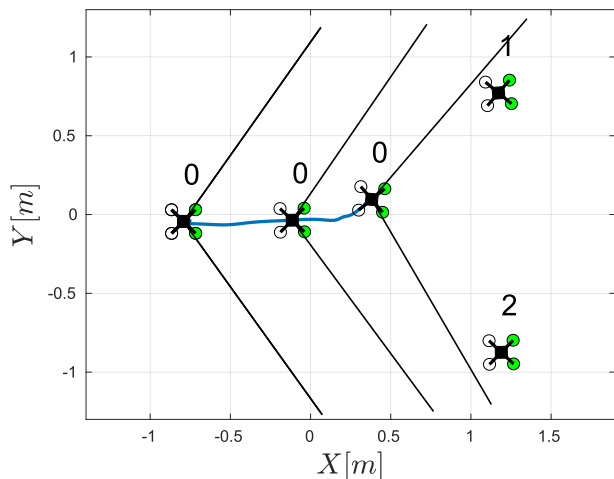


Figure 6. 2D plot of the position of the aerial robots while one is tasked to move in between the other two.

## VI. EXPERIMENTAL EVALUATION

### A. EXPERIMENTAL SETUP

We performed experiments with 3 quadrotors deployed in an indoor flying space of  $10 \times 6 \times 4 \text{ m}^3$  at the Agile Robotics and Perception Lab (ARPL) lab at New York University. The used aerial platforms, depicted in Fig. 1 are custom small-scale aerial robots equipped with a Qualcomm<sup>®</sup> Snapdragon<sup>™</sup> Flight<sup>™</sup> Pro board and on-board VIO, planning, and control based on [28]. The framework has been developed in ROS. Communication among drones is implemented using a synchronized multi-master network module [29]. The optimization problem realizing the CBF is implemented using OSQP [30] and given as input to the low-level control on each robot. The value of the field of view  $\beta$  is set to  $100^\circ$ . The minimum distance  $D$  among drones is set to  $0.5\text{m}$ . The maximum velocity magnitude used to compute  $\dot{\mathbf{p}}_j$  is set to  $0.1\text{m/s}$ .

The cost matrix  $H$  is a diagonal matrix set to  $H = \text{diag}(1.0, 1.0, 0.05)$  in order to inform the optimization solver that the components  $i_{v_x}, i_{v_y}$  of  $\mathbf{u}_i$  are more important than  $\omega$ . The solution to the CBF optimization problem considers only a maximum of 5 neighbor drones. Since the constraint is active only if a neighbor drone reaches the edge of the field of view, potentially occluding other drones, this limitation of the number of considered drones on the 5 with the lowest  $h(\cdot)$  allows the solution on board of the problem with a computation time of 0.4 ms on average. In all the

following tests, the parameters  $\gamma_F$  and  $\gamma_S$  are set to 1.0 and 2.0, respectively. The value for  $\gamma_F$  has been empirically tuned to allow the usage of most of the available field of view. The value for  $\gamma_S$  has been selected according to what is presented in [21] to maintain the drones distant enough to not generate any downdraft. The safe distance  $D$  is set to  $0.5\text{m}$ . The detection of the neighbors for each agent is computed off-board using measurements from a motion-capture system that are filtered given the orientation of the considered drone.

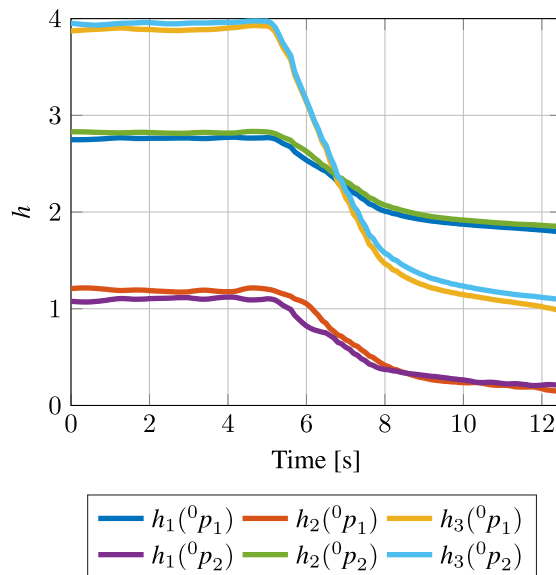


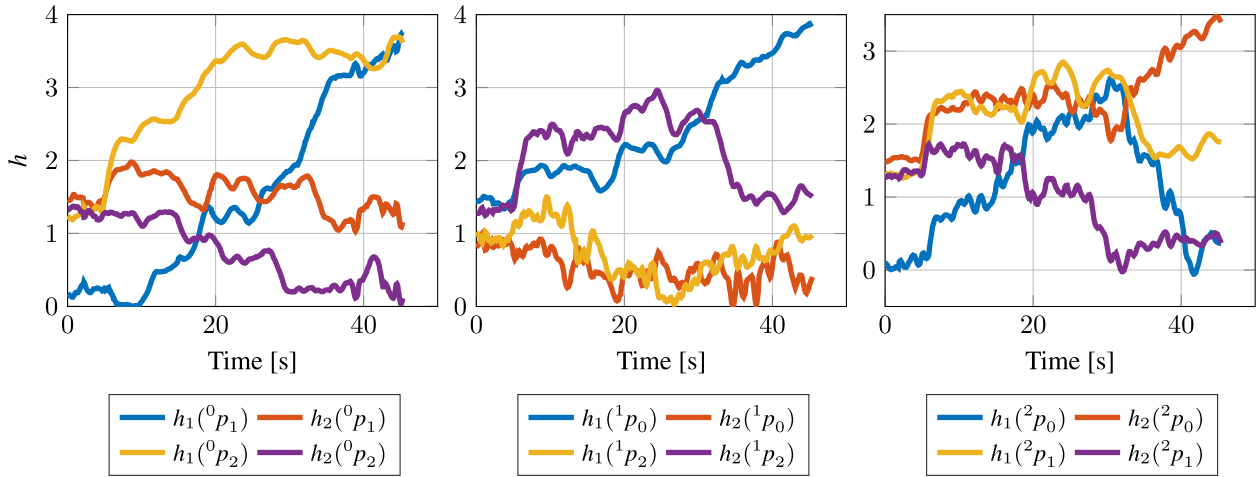
Figure 7. Visualization of the values of  $h$  during the test. The value is generally maintained greater than 0.

### B. CBF VALIDATION

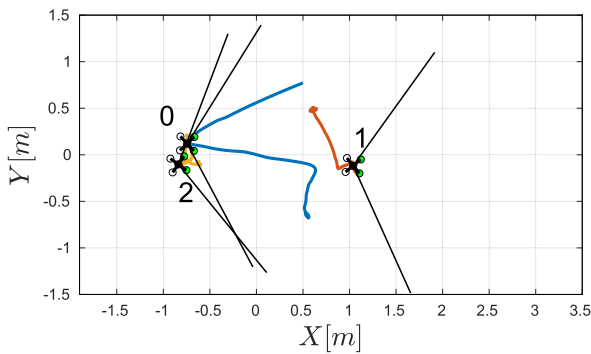
To validate the proposed formulation we propose the following test. A single aerial platform is tasked to move in between two other drones. Without any restrictions, the drone would simply pass through, ignoring the positions of the other two drones. The introduction of the proposed CBF stops the drone motion maintaining the other two drones in the field of view. Figure 6 displays the 2D poses of the 3 drones involved, the drone on the left is tasked to pass in the middle of the drones on the right. The motion is stopped before the neighbors exit the field of view. Figure 7 displays the behavior of the components of  $h$  of drone 0 with respect to both neighbor drones. It is clearly shown that the neighbors are kept in a safe set and any violation is negligible. This confirms the effect of the CBF on maintaining the current connection between the drones.

### C. CASE STUDY: COVERAGE CONTROL

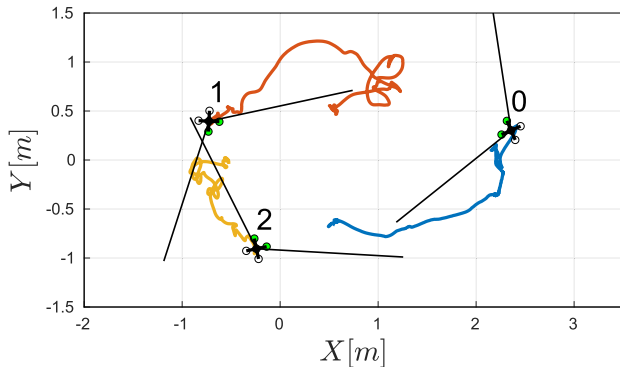
The following test performs coverage control in a closed environment using only limited information coming from the field of view of the robot. The algorithm starts from a configuration where each robot can see the other two, generating a complete graph connecting the robots. Figure 9



**Figure 8.** Visualization of the values of  $h$  for the three robots (drones 0, 1, and 2 are displayed from left to right). Each robot detects two other drones. The left component  $h_1(\cdot)$  is visualized in blue and yellow, while the right component  $h_2(\cdot)$  is visualized in red and purple.



**Figure 9.** Coverage control with a limited field of view. The aerial platforms end up crashing with each other because of the lack of information.



**Figure 10.** Coverage control. The drones start by facing each other and move in the environment according to the motion generated by the control barrier function. The trajectory of drones 0, 1, and 2 are depicted in blue, red, and yellow respectively.

shows the behavior of the drones without any constraint. It is shown that two aerial robots clash with each other since one robot has no information on the presence of the other, therefore the coverage algorithm instructs the robot to move towards what it considers “free space”, causing the accident. Figure 10 displays the behavior of the controlled system with the effect of the proposed CBF.

The robots initially spread over the environment until the constraint becomes active. The modified control action generates a motion where the drones start circling in the counter-clockwise direction till they are set in the final displayed position, where every drone has the other two in the field of view. Figure 8 displays the behavior of  $h$  for the robots throughout the test. The data shows again that the field of view boundaries are enforced by constraining the requested motion. By maintaining the initial information flows on the robot’s controller, the safety of the algorithm is improved since the robot knows all the necessary information to avoid crashing with the other robots. It can be noted that the cumulative application of the CBF to the motion of all three aerial robots generates significant oscillations in the values of  $h$  when the robot is inside the safe set. These oscillations are greatly reduced when the detected robot approaches a border of the field of view, due to the effect of the CBF. To further visualize the presented results we refer readers to the accompanying video displaying the experiments and the associated position plots.

## VII. CONCLUSION

In this paper, we proposed a novel control formulation to maintain the topology of the directed graph describing the connection between agents in a multi-robot system. Specifically, we explore the case where the agents have a limited field of view due to local sensing and propose a CBF-based optimization controller that modifies a generic desired input in a minimally invasive fashion. The CBF is formulated in order to enforce the maintenance of the current neighbors inside the field of view. The validity of the CBF is tested using aerial platforms by performing a scenario where the desired input purposely directs platforms toward an undesirable configuration. The data shows how the control is capable of stopping the undesired motion. The control layer is then inserted in the coverage control scenario where the robots monitor an area. The provided data shows the

effectiveness of the proposed solution on the maintenance of the current connections and therefore improving the safe and proper execution of the coverage task.

In this work, we assume to not have information on the neighbors' velocity therefore we considered the robots moving toward the closest border as part of the CBF formulation. This assumption, alongside the assumption on the bounded speed and planar motion of the aerial platform, limits the performance at which the team of robots can accomplish the required task. Future work will focus on relaxing these assumptions by incorporating the whole system dynamics in order to increase the range of applications of the proposed controller. Furthermore, since the current validation is based on data from a motion capture system, we will focus on extending the proposed solution using only onboard robot sensor data.

## ACKNOWLEDGMENT

Giuseppe Loianno serves as consultant for the Technology Innovation Institute. This arrangement has been reviewed and approved by the New York University in accordance with its policy on objectivity in research.

## REFERENCES

- [1] M. Dorigo, G. Theraulaz, and V. Trianni, "Reflections on the future of swarm robotics," *Sci. Robot.*, vol. 5, no. 49, Dec. 2020, Art. no. eabe4385.
- [2] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.
- [3] M. Kegeleirs, D. Garzón Ramos, and M. Birattari, "Random walk exploration for swarm mapping," in *Proc. Annu. Conf. Towards Auto. Robot. Syst.* Cham, Switzerland: Springer, 2019, pp. 211–222.
- [4] J. Kim, "Cooperative exploration and networking while preserving collision avoidance," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4038–4048, Dec. 2017.
- [5] M. Catellani, F. Pratisoli, F. Bertonecchi, and L. Sabattini, "Coverage control for exploration of unknown non-convex environments with limited range multi-robot systems," in *Proc. 16th Int. Symp. Distrib. Auto. Robot. Syst.*, 2022.
- [6] M. Senanayake, I. Senthoooran, J. C. Barca, H. Chung, J. Kamruzzaman, and M. Murshed, "Search and tracking algorithms for swarms of robots: A survey," *Robot. Auto. Syst.*, vol. 75, pp. 422–434, Jan. 2016.
- [7] R. Ge, M. Lee, V. Radhakrishnan, Y. Zhou, G. Li, and G. Loianno, "Vision-based relative detection and tracking for teams of micro aerial vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 380–387.
- [8] C. Cheng, X.-Y. Yu, L.-L. Ou, and Y.-K. Guo, "Research on multi-robot collaborative transportation control system," in *Proc. Chin. Control Decis. Conf. (CCDC)*, May 2016, pp. 4886–4891.
- [9] G. Li, R. Ge, and G. Loianno, "Cooperative transportation of cable suspended payloads with MAVs using monocular vision and inertial sensing," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5316–5323, Jul. 2021.
- [10] L. Sabattini, C. Secchi, N. Chopra, and A. Gasparri, "Distributed control of multirobot systems with global connectivity maintenance," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1326–1332, Oct. 2013.
- [11] M. Fiacchini and I.-C. Morarescu, "Convex conditions on decentralized control for graph topology preservation," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1640–1645, Jun. 2014.
- [12] K. Khateri, M. Pourgholi, M. Montazeri, and L. Sabattini, "A comparison between decentralized local and global methods for connectivity maintenance of multi-robot networks," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 633–640, Apr. 2019.
- [13] S. Yi, W. Luo, and K. Sycara, "Distributed topology correction for flexible connectivity maintenance in multi-robot systems," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 8874–8880.
- [14] P. Ong, B. Capelli, L. Sabattini, and J. Cortés, "Network connectivity maintenance via nonsmooth control barrier functions," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, Dec. 2021, pp. 4786–4791.
- [15] H.-A. Hung, H.-H. Hsu, and T.-H. Cheng, "Image-based multi-UAV tracking system in a cluttered environment," *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 4, pp. 1863–1874, Dec. 2022.
- [16] I. Salehi, G. Rotithor, R. Saltus, and A. P. Dani, "Constrained image-based visual servoing using barrier functions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 14254–14260.
- [17] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 3420–3431.
- [18] M. Nagumo, "Über die Lage der integralkurven gewöhnlicher differentialgleichungen," in *Proc. Physico-Math. Soc. Jpn.*, vol. 24, 1942, pp. 551–559.
- [19] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 6271–6278.
- [20] 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.
- [21] F. Ferraguti, C. T. Landi, A. Singletary, H.-C. Lin, A. Ames, C. Secchi, and M. Bonfè, "Safety and efficiency in robotics: The control barrier functions approach," *IEEE Robot. Autom. Mag.*, vol. 29, no. 3, pp. 139–151, Sep. 2022.
- [22] C. Bhowmick, L. Behera, A. Shukla, and H. Karki, "Flocking control of multi-agent system with leader–follower architecture using consensus based estimated flocking center," in *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2016, pp. 166–171.
- [23] J. Chen, T. Li, Y. Zhang, T. You, Y. Lu, P. Tiwari, and N. Kumar, "Global-and-local attention-based reinforcement learning for cooperative behaviour control of multiple UAVs," *IEEE Trans. Veh. Technol.*, early access, 2024, doi: 10.1109/TVT.2023.3327571.
- [24] J. Chen, P. Han, Y. Zhang, T. You, and P. Zheng, "Scheduling energy consumption-constrained workflows in heterogeneous multi-processor embedded systems," *J. Syst. Archit.*, vol. 142, Sep. 2023, Art. no. 102938. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1383762123001170>
- [25] Z. Wang, Y. Hirata, and K. Kosuge, "Control a rigid caging formation for cooperative object transportation by multiple mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Mar. 2004, pp. 1580–1585.
- [26] F. Pratisoli, B. Capelli, and L. Sabattini, "On coverage control for limited range multi-robot systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 9957–9963.
- [27] F. Bertonecchi, M. Belal, D. Albani, F. Pratisoli, and L. Sabattini, "On limited-range coverage control for large-scale teams of aerial drones: Deployment and study," in *Proc. 16th Int. Symp. Distrib. Auto. Robot. Syst.*, 2022.
- [28] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, "Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and IMU," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 404–411, Apr. 2017.
- [29] J. S. Hernandez and C. F. Herrero, "Multi-master ROS systems technical report," Institut de Robotica i Inf. Ind., Barcelona, Spain, Tech. Rep. IRI-TR-15-0, 2015. [Online]. Available: <https://www.iri.upc.edu/publications/show/1607>
- [30] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Math. Program. Comput.*, vol. 12, no. 4, pp. 637–672, Dec. 2020.



**FILIPPO BERTONCELLI** (Member, IEEE) received the B.Sc. and M.Sc. degrees in mechatronics engineering and the Ph.D. degree in industrial innovation engineering from the University of Modena and Reggio Emilia, Italy, in 2015, 2018, and 2022, respectively. He has been a Postdoctoral Researcher with the Department of Science and Methods for Engineering, since 2022. His research interests include multirobot systems, decentralized control, mobile robotics, multirobot cooperative manipulation, and non-prehensile manipulation.





**VIVEK RADHAKRISHNAN** received the Bachelor of Engineering degree (Hons.) in electrical and electronics engineering from BITS Pilani, Dubai Campus, in 2014, and the Master of Science degree in mechatronics and robotics with the NYU Tandon School of Engineering, New York, in 2023.

He has over nine years of experience in the robotics industry. He was a Robotics Engineer with Algorithma and B.U.T, Abu Dhabi, and as a Senior Researcher with the Technology Innovation Institute, Abu Dhabi. In 2023, he co-founded a startup that is developing innovative robotics solutions.



**MATTIA CATELLANI** (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in mechatronics engineering from the University of Modena and Reggio Emilia, Italy, in 2019 and 2022, respectively, where he is currently pursuing the Ph.D. degree in industrial innovation engineering. His research interests include multi-robot systems and distributed control.



**GIUSEPPE LOIANNO** (Member, IEEE) received the Ph.D. degree in robotics from the University of Naples Federico II, Naples, Italy, in 2014. He is currently an Assistant Professor with New York University, New York, NY, USA, and the Director of the Agile Robotics and Perception Laboratory, involved on autonomous robots. Prior to joining NYU, he was a Postdoctoral Researcher, a Research Scientist, and a Team Leader with the GRASP Laboratory, University of Pennsylvania,

Philadelphia, PA, USA. He has authored or coauthored more than 70 conference papers, journal articles, and book chapters. His research interests include perception, learning, and control for autonomous robots. He was a recipient of the NSF CAREER Award in 2022, the DARPA Young Faculty Award in 2022, the IROS Toshio Fukuda Young Professional Award in 2022, the Conference Editorial Board Best Associate Editor Award at ICRA 2022, and the Best Reviewer Award at ICRA 2016. He was selected as the Rising Star in AI from KAUST in 2023. He is the Co-Chair of the IEEE RAS Technical Committee on Aerial Robotics and Unmanned Aerial Vehicles. He was the General Chair of the IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), in 2021, and the Program Chair, in 2019, 2020, and 2022. His work has been featured in a large number of renowned international news and magazines.



**LORENZO SABATTINI** (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in mechatronic engineering from the University of Modena and Reggio Emilia, Italy, in 2005 and 2007, respectively, and the Ph.D. degree in control systems and operational research from the University of Bologna, Italy, in 2012. In 2010, he was a Visiting Researcher with the University of Maryland, College Park, MD, USA. He has been an Associate Professor with the Department of Sciences and

Methods for Engineering, University of Modena and Reggio Emilia, since 2018. His research interests include multirobot systems, decentralized estimation and control, and mobile robotics. He was the Founding Co-Chair of the IEEE RAS Technical Committee on Multi-Robot Systems and served as the corresponding co-chair from 2014 to 2021. He served as an Associate Editor for IEEE ROBOTICS AND AUTOMATION LETTERS from 2015 to 2018 and *IEEE Robotics and Automation Magazine* from 2017 to 2019. He is also serving as an Editor for IEEE ICRA and IEEE/RSJ IROS conferences and an Associate Editor for *The International Journal of Robotics Research* (IJRR).

...

Open Access funding provided by 'Università degli Studi di Modena e Reggio Emilia' within the CRUI CARE Agreement