Distributed Control Barrier Functions for Global Connectivity Maintenance

Nicola De Carli, Paolo Salaris, Paolo Robuffo Giordano

***Abstract*— In this work, we propose a framework for the distributed implementation of Quadratic Programs-based controllers, building upon and rectifying a significant limitation in a previously presented approach. The proposed framework is primarily motivated by the distributed implementation of Control Barrier Functions (CBFs), whose primary objective is to make minimal adjustments to a nominal controller while ensuring constraint satisfaction. By improving over some limitations in the current state-of-the-art, we are able to apply distributed CBFs to the problem of global connectivity maintenance in presence of communication and sensing constraints. Specifically, we consider the problem of preserving connectivity for a group of quadrotors with onboard sensors under distance and field of view constraints. Leveraging distributed control barrier functions, our approach maintains global graph connectivity while optimizing the performance of the desired task. Numerical simulations validate its effectiveness.**

I. Introduction

Coordinated teams of robots are finding more and more applications in critical domains, including tasks such as mapping, surveillance, disaster response, exploration, border security, and patrol missions [1]–[3]. Effective autonomous coordination in multi-robot systems relies on the possibility to exchange information across the network and on the availability of accurate relative localization in a shared coordinate frame. However, centralized infrastructures like the Global Positioning System (GPS) may not always be available, motivating research on the topic of sensor-based cooperative localization in a common frame using, e.g., onboard cameras or Ultra Wideband (UWB) sensors [4]–[6].

Communication and sensing interactions in robot teams are often modeled by graph theory, and connectivity of the interaction graph is crucial for information flow and, thus, for convergence of cooperative algorithms such as for localization. In real-world scenarios, however, communication and relative sensing may be hindered by limitations such as maximum range, limited Field of View (FoV), and so on. Therefore, any motion strategy unaware of these limitations can result in graph disconnections and the consequent mission failure. This has indeed motivated extensive research on the topic of connectivity maintenance for ensuring the fulfilment of missions also in presence of sensing/communication constraints [7]–[12]. Connectivity maintenance strategies can be categorized as *local* [13]–[15] and *global* [8]–[12]. The local approaches aim at maintaining all local connections, but they can severely restrict robot mobility. In contrast, the global approaches employ algebraic graph theory concepts, such as the *connectivity* (or *Fiedler*) *eigenvalue*, enabling robots to create and lose edges at runtime while preserving global graph connectivity, thus granting greater mobility at the cost of an increased complexity [16].

Preserving connectivity of the sensing graph can be even more challenging than for the communication graph because of the typical stricter sensing constraints (which, sometimes, can also result in a *directed* sensing graph). As a result, there exists a need for connectivity maintenance algorithms that minimally impact the execution of any higher-level mission. While most of the works on connectivity maintenance are based on gradient descent of suitable potential functions [8]–[10], [12], recent works [11], [17] have leveraged (centralized) Control Barrier Functions (CBFs) to achieve minimal modifications of any nominal controller, providing both constraint satisfaction and performance optimization [18]. Consequently, when compared to methods based on potential functions, CBF-based methods offer greater freedom of movement while respecting the constraints, which is a critical advantage in multi-robot coordination.

CBFs have become a widespread tool for ensuring constraint satisfaction while optimizing performance in nonlinear control problems [18]. They have found applications in various multi-agent scenarios, including collision avoidance [19], connectivity maintenance [11], [17], target tracking for observability maintenance [20], and temporal logic tasks [21]. These works either solved the CBF-induced Quadratic Program (QP) in a centralized way [11], [17], or employed pre-allocation schemes, ensuring distributed constraint satisfaction but at the cost of optimality [19]–[21]. Some other methods [22] could achieve centralized optimality but without ensuring constraint satisfaction at all times. In [23], a novel solution was introduced that converges to the optimal centralized solution in finite time while guaranteeing constraint satisfaction at each instant. This approach was extended in [24] to accommodate multiple CBFs constraints. However, this approach suffers from restrictive assumptions regarding the local Lie derivative of the CBF. Specifically, [23] requires that the local Lie derivative norm should never approach zero, and even small values can easily lead to numerical problems. The assumption of a non-vanishing local Lie derivative proves excessively restrictive and, as it will be shown in the following, it is often not met in various applications of interest including connectivity maintenance.

Our work builds upon [23] but it removes the assumption

N. De Carli and P. Robuffo Giordano are with CNRS, Univ Rennes, Inria, IRISA, Rennes, France. {nicola.de-carli,prg}@irisa.fr

P. Salaris is with the Research Center "E. Piaggio" and the Dept. of Information Engineering, University of Pisa, Pisa, Italy. paolo.salaris@unipi.it

This work was supported by the ANR-20-CHIA-0017 project "MULTI-SHARED"

of a non-vanishing local Lie derivative of the CBF. This improvement enables us to design a CBF-based approach for maintaining global connectivity in a formation of quadrotors with range and FoV constraints and, crucially, without the need to comply with the (restrictive) assumptions of [23]. Furthermore, the CBF control design allows for a better performance w.r.t. the more classical approaches based on potentials (e.g., [9]), since the performance of the main task can be explicitly optimized while always ensuring constraint satisfaction. Furthermore, with respect to [11], [17], the proposed approach is fully distributed and it includes FoV limitations.

The paper is structured as follows: Section II introduces modeling assumptions, Section III presents the distributed CBFs algorithm, Section IV applies the algorithm to connectivity maintenance, Section V showcases simulation results, and finally, Sect. VI provides concluding remarks.

II. Preliminaries

A. Interaction Model

We consider a group of N robots whose sensing interactions are modeled by a time-varying directed graph. The existence of a directed edge $e_{dk}(t) := (i, j)$ is associated to a state dependent weight $\bar{a}_{ij}(t) \geq 0$, such that $\bar{a}_{ij}(t) > 0$ if robot i can sense robot j and $\bar{a}_{ij}(t) = 0$ otherwise. Furthermore, we denote the undirected counterpart of the sensing graph with $\mathcal{G}_u := (\mathcal{V}, \mathcal{E}_u(t))$, which is used to model the communication interactions (assumed bidirectional). Therefore, an edge $e_{uk}(t) := (i, j) \in \mathcal{E}_u(t)$ exists if either robot i can sense robot j or vice versa. The weight associated to the edge is given by $a_{ij}(t) := \bar{a}_{ij}(t) + \bar{a}_{ji}(t) \geq 0$, such that $a_{ij}(t) > 0$ if either robot i can measure robot j or vice versa, and $a_{ij}(t) = 0$ otherwise. Two robots i and j are defined as neighbors at time t if $(i, j) \in \mathcal{E}_u(t)$ and the set of neighbors of robot i is denoted by $\mathcal{N}_i(t) := \{j \in \mathcal{V} | (i, j) \in \mathcal{E}_u(t)\}$. Given the weighted *adjacency matrix* $\boldsymbol{A}_w(t) := [a_{ij}(t)] \in \mathbb{R}^{N \times N}$ associated to the undirected graph, the Laplacian matrix associated to the graph $\mathcal{G}_u$ is defined as $\boldsymbol{L}_w(t) = \mathrm{diag}(\boldsymbol{A}_w(t)\mathbf{1}_N) - \boldsymbol{A}_w(t)$, where $\mathbf{1}_N \in \mathbb{R}^N$ is a vector of all-ones. The Laplacian matrix is positive semidefinite with $\boldsymbol{L}_w \mathbf{1}_N = \mathbf{0}$, i.e. it has zero row-sum. The Laplacian of a graph has eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_N$ with eigenvectors $\{\mathbf{1}_N/\sqrt{N}, \boldsymbol{v}_2, ..., \boldsymbol{v}_N\}$ and the second (a.k.a. Fiedler) eigenvalue $\lambda_2 > 0$ if and only if the graph is connected [16], [25].

Considering this particular setup with *directed* sensing graph and *undirected* communication graph is relevant whenever the robots need to sense each other and the onboard sensors are not omnidirectional, as it is, e.g., the case of onboard cameras with limited fov.

B. Formation Model

We consider a group of N quadrotor UAVs equipped with onboard Inertial Measurement Units (IMUs), calibrated cameras and relative distance sensors, which are able to exchange data over a radio communication channel. Similarly to previous works [4], [26], we consider a simplified kinematic model in $\mathbb{R}^3 \times \mathcal{S}^1$ for the i-th quadrotor with body-frame velocities and yaw rate commands $\boldsymbol{u}_i := \begin{bmatrix} \boldsymbol{v}_i^T & \omega_i \end{bmatrix}^T$:

$$\begin{bmatrix} \dot{\boldsymbol{p}}_i \\ \dot{\psi}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_i & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_i \\ \omega_i \end{bmatrix} = \boldsymbol{T}(\psi_i)\boldsymbol{u}_i \tag{1}$$

where $\boldsymbol{p}_i \in \mathbb{R}^3$ is the robot position, $\psi_i \in \mathcal{S}^1$ is the yaw and $\boldsymbol{R}_i := \boldsymbol{R}_z(\psi_i) \in SO(3)$ is the associated rotation matrix. We denote the state of the i-th robot as $\boldsymbol{q}_i = \begin{bmatrix} \boldsymbol{p}_i^T & \psi_i \end{bmatrix}^T$. For convenience, we also denote the stack of the state of each robot as $\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_1^T & \dots & \boldsymbol{q}_N^T \end{bmatrix}^T \in (\mathbb{R}^3 \times \mathcal{S}^1)^N$ and analogously for the input $\boldsymbol{u}$. Then, the full dynamics can be written as

$$\dot{\boldsymbol{q}} = \bar{\boldsymbol{T}}(\boldsymbol{\psi})\boldsymbol{u}, \tag{2}$$

with $\bar{\boldsymbol{T}}(\boldsymbol{\psi}) := \mathrm{diag}(\boldsymbol{T}(\psi_i))$ being a block-diagonal matrix.

The robots are assumed to be equipped with a camera-like sensor with limited range and FoV and a sensor providing relative distance measurements. In particular, when robot j is visible by robot i, we assume that robot i can obtain a measurement of the relative position of robot j in its own body-frame, i.e.:

$${}^i\boldsymbol{p}_{ij} = \boldsymbol{R}_i^T(\boldsymbol{p}_j - \boldsymbol{p}_i) \tag{3}$$

From such measurements, the robots are able to estimate their relative orientation, see e.g. [10] assuming that the sensing graph remains weakly connected, hence, motivating the necessity of connectivity maintenance keeping into account sensing constraints.

III. Distributed Control Barrier Functions

In this section, we describe our contribution concerning distributed CBFs, which build upon the approach presented in [23]. In particular, as explained in the Introduction, we are able to lift the restrictive assumption concerning the local Lie derivative of the control barrier function not vanishing [23]. We use the global connectivity maintenance task to show the importance of lifting such assumption.

A. Introduction

Consider a system in control-affine form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{\xi} \tag{4}$$

In this equation, $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1^T & \dots & \boldsymbol{x}_N^T \end{bmatrix}^T \in \mathbb{R}^n$ denotes the state, while $\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\xi}_1^T & \dots & \boldsymbol{\xi}_N^T \end{bmatrix}^T \in \mathbb{R}^m$ represents the input. The functions $\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{f}_1(\boldsymbol{x}_1)^T & \dots & \boldsymbol{f}_N(\boldsymbol{x}_N)^T \end{bmatrix}^T$ and the block-diagonal $\boldsymbol{g}(\boldsymbol{x}) = \mathrm{diag}(\boldsymbol{g}_i(\boldsymbol{x}_i))$ represent the system dynamics. It is important to note that (2) belongs to this class of systems.

In many applications, the evolution of the state of a dynamical system needs to be restricted to a set $\mathcal{C}$, referred to as *safe set*, which can be described by the superlevel set of a differentiable function $h : \mathbb{R}^n \to \mathbb{R}$:

$$\mathcal{C} := \{\boldsymbol{x} \in \mathbb{R}^n : h(\boldsymbol{x}) \geq 0\}. \tag{5}$$

For example, in the connectivity maintenance case one wants to maintain $\lambda_2 \geq \epsilon$ with $\epsilon > 0$, hence, one can define $h(\boldsymbol{x}) := \lambda_2 - \epsilon$. Let us consider the following two definitions instrumental for the following.

Definition III.1. A continuous function $\alpha : (-b, a) \to (-\infty, \infty)$ is an *extended class* $\mathcal{K}$ function if it is strictly increasing and $\alpha(0) = 0$.

Definition III.2 ([18])**.** Let $\mathcal{C}$ be defined by (5). $h(\boldsymbol{x})$ is a *control barrier function* (CBF) for the system (4) if there exists a locally Lipschitz extended class $\mathcal{K}$ function α such that:

$$\sup_{\boldsymbol{\xi}\in\mathbb{R}^m} [L_f h(\boldsymbol{x}) + L_g h(\boldsymbol{x})\boldsymbol{\xi} + \alpha(h(\boldsymbol{x}))] \geq 0 \quad \forall \boldsymbol{x} \in \mathbb{R}^n \quad (6)$$

where $L_f h$ and $L_g h$ are the Lie derivatives along the flows of $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$ respectively. Any locally Lipschitz controller satisfying (6) renders the safe set $\mathcal{C}$ forward invariant and, if $\mathcal{C}$ is compact, asymptotically stable [27]. The class of constraints that we consider is the same as those considered in [23], that is, functions $h(\boldsymbol{x})$ such that the CBF parameters of (6) are locally obtainable, meaning that (6) can be expressed as:

$$\sum_{i=1}^{N} \boldsymbol{a}_i^T(\boldsymbol{x}_i^{\mathcal{N}_i})\boldsymbol{\xi}_i + \sum_{i=1}^{N} b_i(\boldsymbol{x}_i^{\mathcal{N}_i}) \leq 0 \quad (7)$$

where $\boldsymbol{x}_i^{\mathcal{N}_i} := \begin{bmatrix}\boldsymbol{x}_i^T & \{\boldsymbol{x}_j^T\}_{j\in\mathcal{N}_i}\end{bmatrix}^T$ is the stack of the state of robot i itself and the one of its neighbors. Notice that, $\boldsymbol{a}_i^T = -L_{gi}h$ while, b_i can vary based on the type of constraint considered and design choice. For connectivity maintenance, one could consider $b_i = -\kappa_i(\boldsymbol{x})(\alpha(h(\boldsymbol{x})) + L_{f_i}h(\boldsymbol{x}))$, with $\kappa_i(\boldsymbol{x})$ being any partition of the unity, i.e. $\sum_{i=1}^{N}\kappa_i(\boldsymbol{x}) = 1$, for example $\kappa_i = \frac{1}{N}$. Note that, as shown in, e.g., [7], by estimating in a distributed way the Fiedler eigenpair, the gradients for global connectivity maintenance can be computed locally, as required by (7).

B. QP formulation

We assume that a nominal controller provides a Lipschitz continuous desired input $\boldsymbol{\xi}^d$ which does not need to satisfy the condition in (6). Then, the following Quadratic Program (QP) is formulated in order to modify in a minimally invasive way the nominal controller, so that condition (6) is always satisfied

$$\begin{aligned} \min_{\boldsymbol{\xi}\in\mathbb{R}^m} \cdot \quad & \frac{1}{2}\sum_{i=1}^{N}\left\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_i^d\right\|_2^2 \\ \text{s.t.} \quad & \sum_{i=1}^{N}\boldsymbol{a}_i^T\boldsymbol{\xi}_i + \sum_{i=1}^{N} b_i \leq 0 \end{aligned} \quad (8)$$

where we omitted the state dependency of $\boldsymbol{a}_i$ and b_i for the sake of readability. However, we remind that these terms change over time as the state evolves along the system trajectories. Defining $\bar{\boldsymbol{a}} := \begin{bmatrix}\boldsymbol{a}_1^T & \dots & \boldsymbol{a}_N^T\end{bmatrix}^T$ and $\bar{b} := \sum_{i=1}^{N} b_i$, we make the following assumption.

Assumption 1. *The QP* (8) *is feasible, i.e.* $\bar{b} \leq 0$ *whenever* $\bar{\boldsymbol{a}} = 0$.

The QP problem in (8) is centralized. The objective is to develop a distributed algorithm that achieves asymptotic convergence to the time-varying centralized optimal solution of the QP while always enforcing the safety constraint. To achieve this objective, we introduce the following equivalent QP

$$\begin{aligned} \min_{(\boldsymbol{\xi},\boldsymbol{y})\in\mathbb{R}^{m+N}} \cdot \quad & \frac{1}{2}\sum_{i=1}^{N}\left\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_i^d\right\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{a}_i^T\boldsymbol{\xi}_i + \sum_{j\in\mathcal{N}_i}(y_i - y_j) + b_i \leq 0, \quad \forall i \in \mathcal{V} \end{aligned} \quad (9)$$

where $\boldsymbol{y} = \begin{bmatrix}y_1, \dots y_N\end{bmatrix}^T \in \mathbb{R}^N$ is an auxiliary variable, with the element y_i associated to robot i. The equivalence among (9) and (8) was shown in [23]. In particular, this implies that each solution $(\boldsymbol{\xi}', \boldsymbol{y}')$ satisfying the constraint in (9), also satisfies the constraint in the original QP (8) and vice versa. The constraints in the previous QP can be equivalently written in matrix form as:

$$\bar{\boldsymbol{A}}\boldsymbol{\xi} + \boldsymbol{L}\boldsymbol{y} + \boldsymbol{b} \leq 0 \quad (10)$$

with $\bar{\boldsymbol{A}} = \mathrm{diag}(\boldsymbol{a}_i^T)$, $\boldsymbol{L}$ is the unweighted Laplacian matrix of the time-varying undirected graph and $\boldsymbol{b} = \begin{bmatrix}b_1 & \dots & b_N\end{bmatrix}^T$. The equivalence between (8) and (9) is based on the fact that $\boldsymbol{1}^T\boldsymbol{L} = \boldsymbol{0}$. One may, then, formulate the following QP

$$\begin{aligned} \min_{(\boldsymbol{\xi}_i,\boldsymbol{y})\in\mathbb{R}^{m_i+N}} \cdot \quad & \frac{1}{2}\left\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_i^d\right\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{a}_i^T\boldsymbol{\xi}_i + \sum_{j\in\mathcal{N}_i}(y_i - y_j) + b_i \leq 0 \end{aligned}. \quad (11)$$

The challenge here is that consistency of $\boldsymbol{y}$ needs to be preserved across the agents, i.e. the local copy y_j of agent i needs to be equal to the local copy y_j of agent j. To solve this issue, in [23] the Karush-Kuhn-Tucker (KKT) optimality conditions of the problem (9) were studied and a distributed adaptive law for updating $\boldsymbol{y}$ was proposed so that $\boldsymbol{y}$ converges to the optimal $\boldsymbol{y}^*$. Hence, each robot solves (11) but with only $\boldsymbol{\xi}_i$ as decision variables, while $\boldsymbol{y}$ is fixed in the QP. The optimal solution $\boldsymbol{y}^*$ from the KKT conditions needs to satisfy the following condition $\forall i \in \mathcal{V}$:

$$\begin{cases} \boldsymbol{a}_i^T\boldsymbol{\xi}_i^d + \boldsymbol{l}_i^T\boldsymbol{y}^* + b_i \leq 0, & \text{if} \quad \bar{\boldsymbol{a}}^T\boldsymbol{\xi}^d + \bar{b} \leq 0 \\ \boldsymbol{a}_i^T\boldsymbol{\xi}_i^d + \boldsymbol{l}_i^T\boldsymbol{y}^* + b_i = k\boldsymbol{a}_i^T\boldsymbol{a}_i, & \text{if} \quad \bar{\boldsymbol{a}}^T\boldsymbol{\xi}^d + \bar{b} > 0 \end{cases} \quad (12)$$

where $\boldsymbol{l}_i$ is the i-th row of $\boldsymbol{L}$ and

$$k = (\bar{\boldsymbol{a}}^T\boldsymbol{\xi}^d + \bar{b})/\left\|\bar{\boldsymbol{a}}\right\|_2^2 \quad (13)$$

Since $k \leq 0$ when $\bar{\boldsymbol{a}}^T\boldsymbol{\xi}^d + \bar{b} \leq 0$, a sufficient condition for $\boldsymbol{y}^*$ is simply to satisfy (12) as

$$\boldsymbol{a}_i^T\boldsymbol{\xi}_i^d + \boldsymbol{l}_i\boldsymbol{y}^* + b_i = k\boldsymbol{a}_i^T\boldsymbol{a}_i \quad \forall i \in \mathcal{V} \quad (14)$$

which can be rewritten in matrix form as

$$\boldsymbol{L}\boldsymbol{y}^* = k\boldsymbol{a}_a - \bar{\boldsymbol{A}}\boldsymbol{\xi}^d - \boldsymbol{b}. \quad (15)$$

with

$$\boldsymbol{a}_a = \begin{bmatrix}\boldsymbol{a}_1^T\boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_N^T\boldsymbol{a}_N\end{bmatrix}. \quad (16)$$

At this stage we can highlight the difference between what has been proposed in [23] and our contribution. In [23], local variables are defined as

$$k_i = \frac{1}{\boldsymbol{a}_i^T \boldsymbol{a}_i}(\boldsymbol{a}_i^T \boldsymbol{\xi}_i^d + \boldsymbol{l}_i \boldsymbol{y}^* + b_i) \tag{17}$$

and it was shown that, if $k_i = k_j \forall (i,j) \in \mathcal{E}$, then condition (12) is satisfied. Hence it was proposed to update the variables $\boldsymbol{y}$ according to the following finite time modified consensus:

$$\dot{\boldsymbol{y}} = -k_0 \text{sign}(\boldsymbol{L}\boldsymbol{k}) \tag{18}$$

with $\boldsymbol{k} = \begin{bmatrix} k_1 & \dots & k_N \end{bmatrix}^T$. A crucial point is that, due to (17), this approach requires $\|\boldsymbol{a}_i(t)\| > 0 \; \forall t, \forall i$ which, as also acknowledged in [23], can be quite restrictive. Indeed, as it will be discussed in the next section, this assumption is often not verified also for the standard problem of connectivity maintenance.

To cope with this issue, we propose the following solution. Each robot can estimate k in (13) with two dynamic average consensus, one for computing the numerator average $n_{\text{avg}} := 1/N \sum_{i=1}^N (\boldsymbol{a}_i{}^T \boldsymbol{\xi}_i^d + b_i)$ and one for computing the denominator average $d_{\text{avg}} := 1/N \sum_{i=1}^N \|\boldsymbol{a}_i\|^2$, from which each robot can obtain its estimate of k as $k_i = n_{\text{avg}}/d_{\text{avg}}$. Then, the only unknown left in (15) is $\boldsymbol{y}^*$. While solving a linear system in a distributed manner is generally challenging, leveraging the sparsity of the Laplacian matrix and its positive semidefinite nature enables a distributed solution for finding $\boldsymbol{y}^*$ in (15) through the simulation of the following dynamical system:

$$\dot{\boldsymbol{y}}(t) = -k_y \left(\boldsymbol{L}\boldsymbol{y}(t) - \boldsymbol{r}(t)\right) - k_{\text{avg}} \bar{y}(t) \boldsymbol{1}_N \tag{19}$$

with

$$\boldsymbol{r} := \text{diag}(k_i)\boldsymbol{a}_a - \bar{\boldsymbol{A}}\boldsymbol{\xi}^d - \boldsymbol{b} \tag{20}$$

and $\bar{y} = \frac{1}{N}\boldsymbol{1}_N^T \boldsymbol{y}$ is the mean of $\boldsymbol{y}$, which can be estimated with a dynamic average consensus. This system corresponds to the continuous-time version of the Richardson iteration [28], which is used to solve linear systems in parallel computing schemes where the matrix to be inverted is positive definite (in this case $\boldsymbol{L} + \boldsymbol{1}_N \boldsymbol{1}_N^T$). An alternative, which potentially may lead to faster convergence, is represented by the Jacobi over-relaxation method [28]. We point out that (19) is fully distributed, in fact each robot implements:

$$\dot{y}_i = -k_y \left(\sum_{j \in \mathcal{N}_i(t)} (y_i - y_j) - k_i \boldsymbol{a}_i^T \boldsymbol{a}_i + \boldsymbol{a}_i^T \boldsymbol{\xi}_i^d + b_i \right) - k_{\text{avg}} \bar{y}. \tag{21}$$

Notice that the only equilibrium of this system is given by the minimum-norm solution to (15). We also point out that, the average of $\boldsymbol{y}$ is not relevant as it has no effect on the QP constraint since $\boldsymbol{L}\boldsymbol{1}_N = \boldsymbol{0}$. We add the second term in (19) with the purpose of guaranteeing boundedness of $\boldsymbol{y}$. We also point out that $\boldsymbol{L}\boldsymbol{1}_N \subseteq \boldsymbol{1}_N$ for all t, meaning that the consensus subspace is infinitesimally $\boldsymbol{L}$-invariant [29].

Let $\bar{r}$ denote the mean of $\boldsymbol{r}$, and define $\boldsymbol{y}_\perp := \boldsymbol{y} - \bar{y}\boldsymbol{1}_N$, and similarly $\boldsymbol{r}_\perp$. Consider the error $\boldsymbol{e}_\perp := \boldsymbol{L}\boldsymbol{y}_\perp - \boldsymbol{r}_\perp$. Then:

$$\begin{aligned} \dot{\boldsymbol{e}}_\perp &= \boldsymbol{L}\left(-k_y(\boldsymbol{L}\boldsymbol{y}_\perp - \boldsymbol{r}_\perp)\right) - \dot{\boldsymbol{r}}_\perp \\ &= -k_y \boldsymbol{L}\boldsymbol{e}_\perp - \dot{\boldsymbol{r}}_\perp \\ \dot{y}_{\text{avg}} &= -k_{\text{avg}}\bar{y} + k_y \bar{r} \end{aligned} . \tag{22}$$

where in the first equation the average dynamics of $\dot{\boldsymbol{y}}_\perp$ do not appear as they are in the null space of $\boldsymbol{L}$. We point out that, if the graph is time-varying, then $\boldsymbol{L}$ is switching in time. Suppose that the graph switches in time, in such a way that the union of the graphs is connected, then stability of (19) can be shown [16, Sect.4.2]. Also, it is worth noting that we did not account for the dynamics of the dynamic average consensus utilized to estimate $\bar{y}(t)$ in the preceding analysis. In practice, given that $\boldsymbol{y}$ is not required to be zero-mean but merely bounded, selecting k_{avg} considerably smaller than k_y ensures that the impact of the disagreement in computing $\bar{y}$ remains negligible in (19). Then, the error $\boldsymbol{e}_\perp$ is practically exponentially stable with rate $c := k_y \min_{\tau \in [t_0,t]} \lambda_2(t)$ and $\boldsymbol{y}_\perp$ converges to a ball of size $\frac{1}{c}\sup_{\tau \in [t_0,t]} ||\dot{\boldsymbol{r}}_\perp||$ centered in $\boldsymbol{y}_\perp^* = \boldsymbol{L}^\dagger \boldsymbol{r}$, with $\dagger$ indicating the Moore-Penrose pseudoinverse, which gives the minimum-norm solution to equation (15). The origin of the average $\bar{y}$ is practically exponentially stable as well and, in particular, what is of interest in this case is the fact that it remains bounded. Also, notice that, with the proposed solution, it is *not required* to have $\|\boldsymbol{a}_i\| > 0 \; \forall i$.

Remark 1. *This approach can be extended to the case of multiple CBF constraints using soft minimum functions as shown in [24].*

Interestingly, solving the QP (11) without auxiliary variables is equivalent to pre-allocation schemes such as in [19]–[21], which maintains safety but leads to suboptimal solutions. The role of the auxiliary variables is, in fact, to optimally allocate the constraint among the robots.

We also point out that, for simplicity of exposition, we considered a cost function of the type $\sum_{1=1}^N \left\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_i^d\right\|_2^2$. The extension to a more generic quadratic cost function $\sum_{i=1}^N \frac{1}{2}\boldsymbol{\xi}_i^T \boldsymbol{H}_i \boldsymbol{\xi} + \boldsymbol{w}_i^T \boldsymbol{\xi}_i$ with $\boldsymbol{H}_i \succ 0$ is straightforward, see [30] for the expression of k in this case.

IV. Connectivity maintenance

As we stated in Sect. II-A, a very classic result in graph theory is that an undirected graph is connected if and only if $\lambda_2 > 0$. When the connectivity of the graph depends on sensing and/or communication constraints, e.g. limited range and field of view, a classical approach to ensure connectivity is summarized below by considering several previos works on this subject [8]–[12], [17]:

- design a weighted adjacency matrix $\boldsymbol{A}_w = [a_{ij}]$, such that the weight a_{ij} smoothly goes to zero when the edge between robot i and robot j approaches disconnection (because of the sensing/communication constraints);
- estimate in a distributed way the Fiedler eigenpair, i.e. $(\lambda_2, \mathbf{v}_2)$ of the corresponding weighted Laplacian matrix $\boldsymbol{L}_w$ [7], [31];
- design a control strategy, e.g. based on potential functions [8]–[10], [12] or based on CBFs [11], [17], which ensures that $\lambda_2(t) \geq \epsilon$ with $\epsilon > 0 \; \forall t$.

A fundamental assumption is that, at the initial time t_0, the undirected graph $\mathcal{G}_u$ is connected, i.e., $\lambda_2(t_0) > 0$. We point out that an approach based on CBFs, as in the present work,

does not require $\lambda_2(t_0) \geq \epsilon$, in fact, the safe set can be shown to be asymptotically stable under the effect of the CBF [11], [18]. In this work, we consider distance constraints, expressed as $d_{\min} \leq d_{ij} \leq d_{\max}$, where $d_{ij} = \|\boldsymbol{p}_j - \boldsymbol{p}_i\|$ and field of view constraints $c_{ij} = \boldsymbol{\beta}_{ij}^T \boldsymbol{e}_1 \geq c_{\min}$, where $\boldsymbol{\beta}_{ij} = {}^i\boldsymbol{p}_{ij}/d_{ij}$ is the relative body-frame bearing, $\boldsymbol{e}_1 = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^T$ and $c_{\min}$ is the cosine of the maximum FoV angle $\alpha_{\max}$ (see Fig. 1). Notice that c_{ij} is the cosine of the angle among

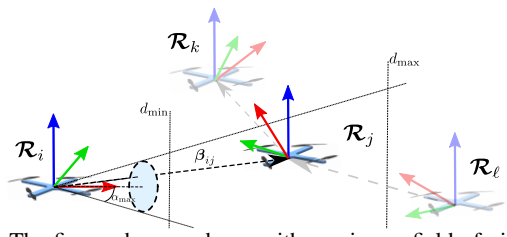

Fig. 1: The figure shows a drone with maximum field of view angle $\alpha_{\max}$ and range limits $d_{\min}$ and $d_{\max}$

the bearing and the robot x-axis to which the camera is assumed to be aligned with. As mentioned in Sect. II-A, we adopt symmetric weights, i.e. $a_{ij} = a_{ji}$, given by $a_{ij} = \bar{a}_{ij} + \bar{a}_{ji}$, with $\bar{a}_{ij} = w_{d_{ij}} w_{b_{ij}}$, where $w_{d_{ij}}$ (resp. $w_{b_{ij}}$) is a continuously differentiable weight which smoothly varies from 1 to 0 as the distance (resp. FoV) limit of the sensor is approached. Considering symmetric weights allows to perform connectivity maintenance on the undirected graph $\mathcal{G}_u$, ensuring weak connectivity of the directed sensing graph. For $w_{d_{ij}}$ and $w_{b_{ij}}$, we employ the following smooth step functions. Consider the following two functions $\zeta_l(x, x_{\min}, x_{\min}^{th}) = (x - x_{\min})/(x_{\min}^{th} - x_{\min})$ and $\zeta_u(x, x_{\max}, x_{\max}^{th}) = (x - x_{\max}^{th})/(x_{\max} - x_{\max}^{th})$, the weights are expressed as (see Fig. 2):

$$w(\zeta_l, \zeta_u) = \begin{cases} w = 0 & \text{if} \quad \zeta_l < 0 \\ w = 6\zeta_l^5 - 15\zeta_l^4 + 10\zeta_l^3 & \text{if} \quad 0 < \zeta_l < 1 \\ w = 1 & \text{if} \quad \zeta_l > 1 \quad \text{and} \quad \zeta_u < 0 \\ w = 1 - 6\zeta_u^5 + 15\zeta_u^4 - 10\zeta_u^3 & \text{if} \quad 0 < \zeta_u < 1 \\ w = 0 & \text{if} \quad \zeta_u > 1 \end{cases} \tag{23}$$

Then, the gradient of the Fiedler eigenvalue with respect to

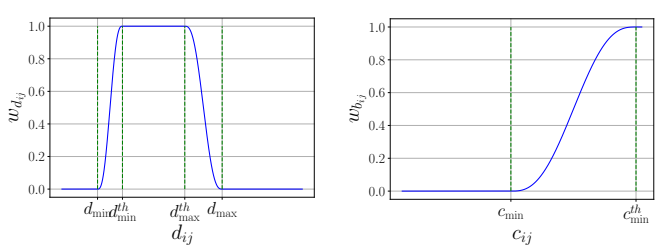

Fig. 2: Weight functions for sensing constraints. On the left distance weight and on the right FoV weight (notice that $c_{ij} \leq 1$).

the robots position can be expressed as it follows [7]:

$$\frac{\partial \lambda_2}{\partial \boldsymbol{q}_i} = \sum_{j \in \mathcal{N}_i} (v_{2j} - v_{2i})^2 \frac{\partial a_{ij}}{\partial \boldsymbol{q}_i} \tag{24}$$

where v_{2i} is the i-th component of the Fiedler eigenvector $\mathbf{v}_2$ and v_{2i} as well as λ_2 are computed in a distributed way using the algorithm presented in [31] and a discrete-time PI consensus [32]. Then in order to maintain $\lambda_2 \geq \epsilon$, each robot solves the following QP:

$$\begin{aligned} \min_{\boldsymbol{u}_i \in \mathbb{R}^4} \cdot \quad & \frac{1}{2} \left\| \boldsymbol{u}_i - \boldsymbol{u}_i^d \right\|_2^2 \\ \text{s.t.} \quad & -\sum_{j \in \mathcal{N}_i} \left((v_{2j} - v_{2i})^2 \frac{\partial a_{ij}}{\partial \boldsymbol{q}_i} \right) \boldsymbol{T}(\psi_i) \boldsymbol{u}_i - \frac{1}{N} \alpha_0 (\lambda_2 - \epsilon)^3 \\ & + \sum_{j \in \mathcal{N}_i} (y_i - y_j) \leq 0, \end{aligned} \tag{25}$$

where we chose to use the extended class $\mathcal{K}$ function $\alpha(x) = \alpha_0 x^3$ with $\alpha_0 > 0$ and $\boldsymbol{y}$ is computed according to (19). Feasibility of the centralized QP for connectivity maintenance is proven in [11] and the same proof is applicable here as the solution of the union of the local QPs is the same as the centralized one. Notice that, as mentioned in Sec. III, it can happen that $\frac{\partial \lambda_2}{\partial \boldsymbol{q}_i} = \mathbf{0}$ but $\frac{\partial \lambda_2}{\partial \boldsymbol{q}} \neq \mathbf{0}$, for example when all the weights of the i-th robot are already at their maximum or when $v_{2i} = v_{2j}$ $\forall j \in \mathcal{N}_i$. In this case, the global QP is still feasible and the i-th robot can simply set $\boldsymbol{u}_i = \boldsymbol{u}_i^d$. Here, the assumption $\boldsymbol{a}_i \neq 0$, made in [23], as in many other applications, is not verified, hence, motivating the alternative approach that we propose.

The algorithm requires the following quantities to be communicated:

- 7 scalar variables for the Power Iteration method [31] and related consensus
- 4 scalar variables to estimate k_i using two PI consensus
- 1 scalar variables to estimate $\boldsymbol{y}$ in (19) and its average

Also, either N (which is used in (25)) is fixed and known by all robots, or it may be estimated using consensus. As previously mentioned, other possibility can be used for $\kappa_i(x)$, other than $\kappa_i = 1/N$, which may be computed using a consensus, e.g. see [21].

V. Simulation Results

This section presents the results of a simulation of $T = 80s$ with $N = 6$ quadrotors, starting from a connected random configuration. The nominal input to the quadrotors is provided by the active sensing controller proposed in [30], which generates trajectories that are exciting for the robot localization but unaware of the sensing constraints. We point out that these trajectories are particularly challenging in our context because the active sensing [30] forces the relative positions among the robots to change very rapidly and, as a consequence, the Laplacian entries and eigenvalues, making it harder to estimate the Fiedler eigenpair and track the optimal solution. The resulting input of each agent is filtered by the connectivity maintenance CBF by solving (25) and updating y_i according to (21). A lower-level safety layer implements collision avoidance using another CBF. The connectivity maintenance runs @25Hz, with the corresponding consensus running @150Hz, while the estimation of the Fiedler eigenpair runs @250Hz and the corresponding consensus run @750Hz. A quadrotor can sense its neighbors if they lie within the distance range $[1, 4.5]m$, with a maximum FoV angle set at 60 degrees. The minimum threshold for the Fiedler eigenvalue λ_2

is set to $\epsilon = 0.2$ and $\alpha_0 = 100$. The norm of the desired input velocity is saturated at $1m/s$ and the yaw-rate at $1rad/s$. The parameters for the power iteration proposed in [31] are set to $\beta = 100$ and $\gamma = 40$, while the consensus used is a discrete-time PI from (31) in [32] with gains $k_p = 0.05$, $k_I = 0.008$ and $\rho = 0.9$. In (21), we use $k_y = 50$ and it runs @250Hz. The robots start to move after $1s$ for allowing the power iteration to reach a satisfactory convergence before starting to move.

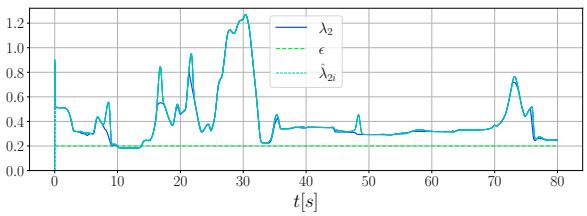

Fig. 3: The plot represents λ_2, the lower bound ϵ and the estimates of the Fiedler eigenvalue $\hat{\lambda}_{2i}$ for each robot (in dashed lines). Notice that for most of the time the estimates are overlapping the real λ_2.

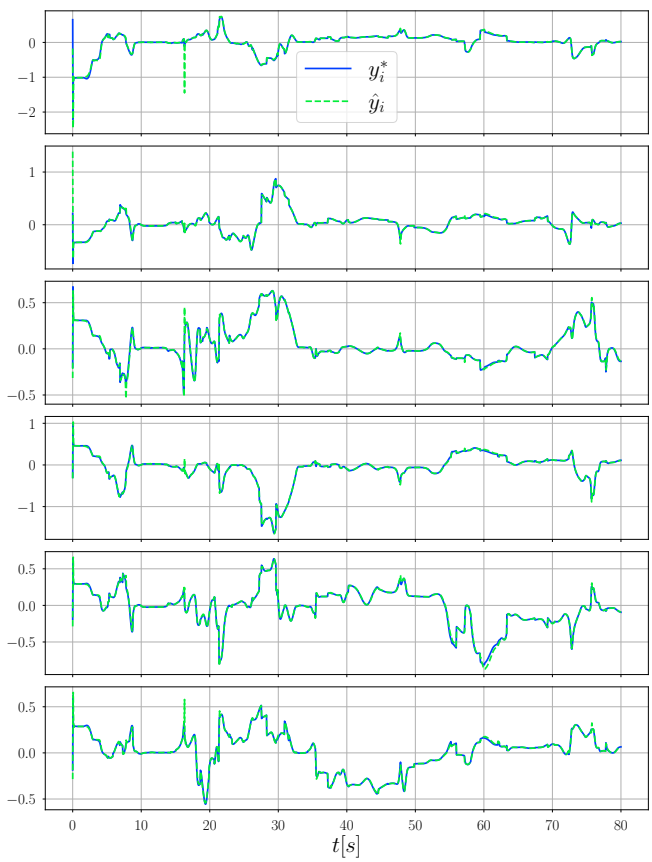

Fig. 4: The plot shows, for each robot, the estimate $\hat{y}_i$ given by (21) (in dashed green) tracking the optimal y_i^* (in solid blue).

In Fig. 3, we report the Fiedler eigenvalue λ_2 along with, the lower bound ϵ and the local estimations for each robot (in dashed lines). We point out that, when λ_3 (and possibly λ_4) get close to λ_2, the estimation becomes much less reliable (as well-known) with the estimate $\hat{\mathbf{v}}_2$ approaching the hyperplane spanned by $\mathbf{v}_2$-$\mathbf{v}_3$ (and possibly $\mathbf{v}_4$).

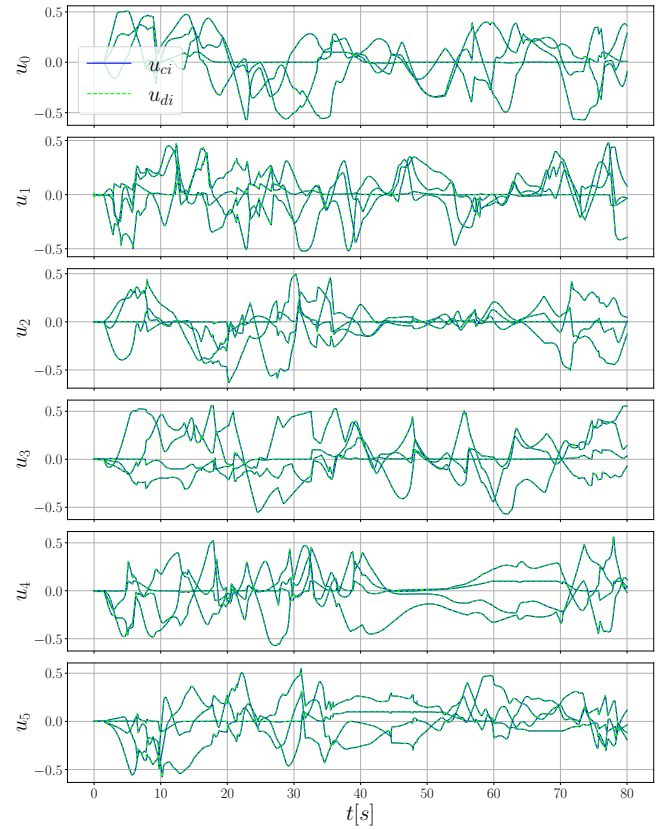

Fig. 5: Comparison among the inputs obtained solving the QP in a distributed way (25) (in dashed green) and in a centralized way (8) (in solid blue). Each plot contains each input components for a certain drone. One can see that after a short initial transient, the trajectories are almost superimposed for most of the time.

We verify the tracking of the optimal $\boldsymbol{y}^*$ in Fig. 4 by plotting $\boldsymbol{y}_\perp$ and the solution $\boldsymbol{L}^\dagger \boldsymbol{r}$ to $\boldsymbol{L}\boldsymbol{y} = \boldsymbol{r}$ as ground truth.

In order to verify the convergence of the distributed solution to the centralized one, we run in parallel the distributed CBFs (25) and the corresponding centralized one (8), whose inputs are not applied and only used as a comparison. The comparative analysis of the resulting inputs is presented in Fig. 5, where it can be noticed the optimal solution is tracked very closely for most of the time.

VI. Conclusions

We have presented a framework for a class of distributed QP-based controllers that typically arise when employing Control Barrier Functions, and we have addressed a significant limitation of the recent literature in this field. Each agent solves a local QP and locally adapts an auxiliary variable. The solutions of the local QPs asymptotically converge to a neighborhood of the centralized optimal solution. This improvement allows us to apply distributed CBFs to the problem of global connectivity maintenance for quadrotor formations, accounting for distance and field of view constraints. The effectiveness of the approach is shown by comparing the obtained inputs with the ones provided by the corresponding centralized CBF.

REFERENCES

[1] R. R. Murphy, K. L. Dreger, S. Newsome, J. Rodocker, E. Steimle, T. Kimura, K. Makabe, F. Matsuno, S. Tadokoro, and K. Kon, "Use of remotely operated marine vehicles at minamisanriku and rikuzentakata japan for disaster recovery," in *2011 IEEE International symposium on safety, security, and rescue robotics*. IEEE, 2011, pp. 19–25.

[2] M. Aggravi, G. Sirignano, P. R. Giordano, and C. Pacchierotti, "Decentralized control of a heterogeneous human–robot team for exploration and patrolling," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3109–3125, 2021.

[3] M. Schranz, M. Umlauft, M. Sende, and W. Elmenreich, "Swarm robotic behaviors and current applications," *Frontiers in Robotics and AI*, p. 36, 2020.

[4] F. Schiano, A. Franchi, D. Zelazo, and P. R. Giordano, "A rigidity-based decentralized bearing formation controller for groups of quadrotor uavs," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 5099–5106.

[5] N. De Carli, P. Salaris, and P. R. Giordano, "Online decentralized perception-aware path planning for multi-robot systems," in *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2021, pp. 128–136.

[6] J. Cano and J. Le Ny, "Ranging-based localizability optimization for mobile robotic networks," *IEEE Transactions on Robotics*, 2023.

[7] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks," *Automatica*, vol. 46, no. 2, pp. 390–396, 2010.

[8] L. Sabattini, C. Secchi, N. Chopra, and A. Gasparri, "Distributed control of multirobot systems with global connectivity maintenance," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1326–1332, 2013.

[9] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "A passivity-based decentralized strategy for generalized connectivity maintenance," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.

[10] M. Bernard, C. Pacchierotti, and P. Robuffo Giordano, "Decentralized Connectivity Maintenance for Quadrotor UAVs with Field of View Constraints," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'23*. Detroit, United States: IEEE, Oct. 2023. [Online]. Available: https://hal.science/hal-04197077

[11] B. Capelli and L. Sabattini, "Connectivity maintenance: Global and optimized approach through control barrier functions," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5590–5596.

[12] P. S. Trakas, C. P. Bechlioulis, and G. A. Rovithakis, "Decentralized global connectivity maintenance for multi-agent systems using prescribed performance average consensus protocols," in *2022 European Control Conference (ECC)*. IEEE, 2022, pp. 524–529.

[13] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.

[14] D. V. Dimarogonas and K. H. Johansson, "Decentralized connectivity maintenance in mobile networks with bounded inputs," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1507–1512.

[15] E. Restrepo, A. Loría, I. Sarras, and J. Marzat, "Edge-based strict lyapunov functions for consensus with connectivity preservation over directed graphs," *Automatica*, vol. 132, p. 109812, 2021.

[16] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[17] P. Ong, B. Capelli, L. Sabattini, and J. Cortés, "Nonsmooth control barrier function design of continuous constraints for network connectivity maintenance," *Automatica*, vol. 156, p. 111209, 2023.

[18] 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)*. IEEE, 2019, pp. 3420–3431.

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

[20] L. Balandi, N. De Carli, and P. R. Giordano, "Persistent monitoring of multiple moving targets using high order control barrier functions," *IEEE Robotics and Automation Letters*, 2023.

[21] L. Lindemann and D. V. Dimarogonas, "Barrier function based collaborative control of multiple robots under signal temporal logic tasks," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1916–1928, 2020.

[22] Y. Chen, M. Santillo, M. Jankovic, and A. D. Ames, "Online decentralized decision making with inequality constraints: an admm approach," *IEEE Control Systems Letters*, vol. 5, no. 6, pp. 2156–2161, 2020.

[23] X. Tan and D. V. Dimarogonas, "Distributed implementation of control barrier functions for multi-agent systems," *IEEE Control Systems Letters*, vol. 6, pp. 1879–1884, 2021.

[24] V. N. Fernandez-Ayala, X. Tan, and D. V. Dimarogonas, "Distributed barrier function-enabled human-in-the-loop control for multi-robot systems," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7706–7712.

[25] F. Bullo, *Lectures on network systems*. Kindle Direct Publishing Seattle, DC, USA, 2020, vol. 1, no. 3.

[26] E. Montijano, D. Zhou, M. Schwager, and C. Sagues, "Distributed formation control without a global reference frame," in *2014 American Control Conference*. IEEE, 2014, pp. 3862–3867.

[27] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[28] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.

[29] F. Bullo, *Contraction Theory for Dynamical Systems*, 1.1 ed. Kindle Direct Publishing, 2023. [Online]. Available: https://fbullo.github.io/ctds

[30] N. De Carli, P. Salaris, and P. R. Giordano, "Multi-robot active sensing for bearing formations," in *2023 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2023, pp. 184–190.

[31] I. Malli, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Robust distributed estimation of the algebraic connectivity for networked multi-robot systems," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9155–9160.

[32] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez, "Tutorial on dynamic average consensus: The problem, its applications, and the algorithms," *IEEE Control Systems Magazine*, vol. 39, no. 3, pp. 40–72, 2019.