

# Connectivity Maintenance: Global and Optimized approach through Control Barrier Functions

Beatrice Capelli, Lorenzo Sabattini

**Abstract**—Connectivity maintenance is an essential aspect to consider while controlling a multi-robot system. In general, a multi-robot system should be connected to obtain a certain common objective. Connectivity must be kept regardless of the control strategy or the objective of the multi-robot system. Two main methods exist for connectivity maintenance: keep the initial connections (local connectivity) or allow modifications to the initial connections, but always keeping the overall system connected (global connectivity). In this paper we present a method that allows, at the same time, to maintain global connectivity and to implement the desired control strategy (e.g., consensus, formation control, coverage), all in an optimized fashion. For this purpose, we defined and implemented a Control Barrier Function that can incorporate constraints and objectives. We provide a mathematical proof of the method, and we demonstrate its versatility with simulations of different applications.

## I. INTRODUCTION

Multi-robot systems are becoming more and more frequent in a wide range of domains, such as industrial [1], agricultural [2], marine [3], [4], and aerial [5]. Communication among the robots is crucial for most cooperative applications, such as search and rescue [6], patrolling [7], [8], or exploration [9], [10]. Hence, the controller for a multi-robot system should take into account communication, which is usually represented by a communication graph, and should be able to maintain connectivity (i.e., possibility for the robots to exchange information among each other).

However, in several applications of interest, the desired input, namely the input defined to achieve the desired objective, can lead to a disconnection of the group. For example, if we want to perform measurements with sensors and maximize the area we analyze, we should control the robots in such a way that they spread as much as possible all over the area. But, as they usually have a limited range of communication, we may cause the disconnection of the group.

In general, to maintain connectivity there are two approaches: local and global. The local approach consists in keeping the connection between two robots if it exists when the task begins, so if the group starts connected it will be connected during the overall task. The main disadvantage of this method is the low flexibility due to the inability to change the communication graph. However, it can be easily implemented in a decentralized manner, thanks to the fact that it only needs local information, and in literature there exist a lot of examples [11]–[13]. On the other side, global

connectivity consists in keeping the group connected allowing a disconnection between the robots, if such disconnection does not lead to a loss of communication in the group. This method allows a higher flexibility, allowing the robots to re-arrange their interconnections to accommodate external needs, such as the presence of obstacles that obstruct their motion (e.g., a narrow passage). In [14]–[16] a decentralized connectivity maintenance is built through a gradient descent approach, while a recent comparison between the two approaches can be found in [17]. In order to increase the flexibility and the efficiency of the multi-robot system, we aim at considering global connectivity preservation together with some desired control input, and combine them in such a way to optimize the task while maintaining communication constraints.

Control Barrier Functions (CBFs) are a suitable method for our problem. In fact, they allow to consider, at the same time, different objectives and constraints. CBFs are commonly used for safety critical applications, such as in the automotive field where they are used for adaptive cruise control [18] and line keeping [19]. Another field of application is human-robot interaction, where CBFs allow to respect some safety constraints and, at the same time, optimize the task [20]. In the multi-robot domain, CBFs are used in [21] to guarantee collision-free movement of a group of mobile robots. In [22] they are exploited to combine different aspects: safety, connectivity, coverage and energy management. Moreover, in [23] CBFs are used to optimize connectivity and coverage. In [22], [23] the approach towards connectivity is local because the CBFs focus on maintaining all the initial links between the robots.

In this paper we propose to integrate global connectivity into a desired control (e.g., coverage, formation control, patrolling), by means of a CBF. This method allows to optimize the desired behavior of the system and, at the same time, to maintain connectivity. The optimization process allows to avoid the tuning procedure, which is usually present when two or more objectives or constraints are considered in the same controller [24]. Like in other global approaches, the initial configuration should generate a connected graph. But, differently from others (e.g., [15]), where the initial value of connectivity should be above a certain desired value, we are able to increase connectivity to the desired value, or above. This property is due to the fact that the chosen CBF is also a Control Lyapunov Function (CLF).

In the paper we give a general introduction of connectivity and of CBFs, we give the demonstration of the implemented method and an extensive evaluation through simulations.

Authors are with the Department of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Italy {beatrice.capelli, lorenzo.sabattini}@unimore.it

## II. NOTATION

The symbol  $\mathcal{L}$  will represent the set of locally Lipschitz functions.  $\mathbb{R}$ ,  $\mathbb{R}_0^+$  are the set of real and real non-negative numbers, respectively. A continuous function  $\Omega(\cdot) : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$  is a class  $\mathcal{K}$  function if it is strictly increasing and  $\Omega(0) = 0$ . It is an extended class  $\mathcal{K}$  if it is a class  $\mathcal{K}$  function, and it is defined on the entire real line, e.g.,  $\Omega(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ .

## III. BACKGROUND

### A. Connectivity

We consider a group of robots, which have a limited communication range  $R$ . The communication network is represented by a communication graph<sup>1</sup>  $\mathcal{G} = \{V, E\}$ , where the set of vertices  $V$  represents the robots and the set of the edges  $E$  consists of the existing communication links. An edge  $e_{i,j}$  exists if and only if the  $i$ -th and the  $j$ -th robot are within the communication range  $R$ . The existence of the edge  $e_{i,j}$  implies also the existence of the edge  $e_{j,i}$ , because we are considering an undirected graph. We define the set of neighbors of the  $i$ -th robot as  $\mathcal{N}_i = \{j \in V | e_{i,j} \in E\}$ , which are all the robots that can communicate with it.

Algebraic connectivity of the system is available directly from the mathematical representation of the graph  $\mathcal{G}$ . Firstly, we introduce the adjacency matrix  $A \in \mathbb{R}^{N \times N}$ :

$$A = \begin{cases} a_{i,j} > 0 & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where  $a_{i,j}$  is the edge weight of  $e_{i,j}$ .

Then we define the degree matrix  $D \in \mathbb{R}^{N \times N}$ , which is a diagonal matrix and each element of the diagonal is equal to  $\psi_{i,i} = \sum_{j=1}^N a_{i,j}$ . Hence,  $D = \text{diag}\{\psi_{i,i}\}$ .

Combining the adjacency and the degree matrix, we obtain the (weighted) Laplacian matrix  $L = D - A$ . The Laplacian matrix has some important properties [25]:

- $L\mathbf{1} = 0$ , which means that the first eigenvalue of the Laplacian is zero and the associated eigenvector is  $\mathbf{1}$ , namely the column vector of all ones;
- Considering the set of eigenvalues of the Laplacian matrix  $(\lambda_1, \dots, \lambda_N)$  we know that:
  - They can be ordered as:  $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$ ;
  - The second eigenvalue  $\lambda_2$  represents the algebraic connectivity of the graph, and we have  $\lambda_2 > 0$  if and only if  $\mathcal{G}$  is connected;

### B. Control Barrier Functions

In this section we introduce the concept of Control Barrier Functions (CBFs), which are an extension of Barrier Functions for systems that do not present a forward invariant set for the desired state of the system. For more details the reader is referred to [26].

We consider an affine control system:

$$\dot{\chi} = f(\chi) + g(\chi)\mu \quad (2)$$

<sup>1</sup>All the formulation is time dependent because the communication network can change over time (e.g.,  $\mathcal{G} = \mathcal{G}(t)$ ). However, when not strictly necessary, we omit dependence on time, for ease of notation.

where  $\chi \in \mathbb{R}^p$  represents the state of the system,  $\mu \in U \subseteq \mathbb{R}^q$  is the control input,  $f(\chi), g(\chi) \in \mathcal{L}$ .

What CBFs aim to do is keeping, or eventually carrying, the system (2) in a certain closed set  $\mathcal{C} \subset \mathbb{R}^p$  defined as:

$$\begin{aligned} \mathcal{C} &= \{\chi \in \mathbb{R}^p | h(\chi) \geq 0\} \\ \partial\mathcal{C} &= \{\chi \in \mathbb{R}^p | h(\chi) = 0\} \\ \text{Int}(\mathcal{C}) &= \{\chi \in \mathbb{R}^p | h(\chi) > 0\} \end{aligned} \quad (3)$$

where  $h(\chi) : \mathbb{R}^p \rightarrow \mathbb{R}$  is a continuously differentiable function. At this point there are two different approaches to define the control strategy: Reciprocal Control Barrier Functions (RCBFs) and Zeroing Control Barrier Functions (ZCBFs). For RCBFs we must define a function  $B(\chi)$  that tends to infinity while the state approaches the boundary of the set  $\mathcal{C}$ , while for ZCBFs the function  $h(\chi)$  must tend to zero on the boundary. We decided to use ZCBFs because they do not present the drawbacks of a function that assumes unbounded values on the boundary and they allow also to consider some perturbations or errors in the system (defining the function  $h(\chi)$  on a set  $\mathcal{D}$  larger than  $\mathcal{C}$ ,  $\mathcal{C} \subseteq \mathcal{D} \subset \mathbb{R}^p$ ). In addition, under certain conditions, they can be straightforwardly transformed in Control Lyapunov Functions, because they are well defined also outside of the set  $\mathcal{C}$ . In the following we will use the term CBF meaning ZCBF.

The CBF should keep the state of the system inside<sup>2</sup> the set  $\mathcal{C}$ , and this is guaranteed choosing the input  $\mu$  in (2) in such a way that:

$$\frac{d}{dt}h(\chi) \geq -\alpha(h(\chi)) \quad (4)$$

where  $\alpha(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^p$  is an extended class  $\mathcal{K}$  function<sup>3</sup> and  $\alpha(\cdot) \in \mathcal{L}$ . Using (2), we can rewrite (4) as an explicit function of  $\mu$ , as follow:

$$\frac{d}{dt}h(\chi) = \frac{\partial h(\chi)}{\partial \chi} \dot{\chi} = L_f h(\chi) + L_g h(\chi)\mu \geq -\alpha(h(\chi))$$

where  $L_f$  and  $L_g$  are the Lie derivatives of  $h(\chi)$ :  $L_f h(\chi) = \frac{\partial h(\chi)}{\partial \chi} f(\chi)$ ,  $L_g h(\chi) = \frac{\partial h(\chi)}{\partial \chi} g(\chi)$ .

Then, the function  $h(\chi)$  is a CBF if the following properties are satisfied [26]:

**Property 1**  $h(\chi)$  is continuously differentiable.

**Property 2**  $h(\chi)$  is of relative degree one (i.e., its first order time derivative depends explicitly on the control input).

**Property 3** It is possible to find an extended class  $\mathcal{K}$  function  $\alpha(h(\chi))$  such that:

$$\sup_{\mu \in U} [L_f h(\chi) + L_g h(\chi)\mu + \alpha(h(\chi))] \geq 0, \quad \forall \chi \in \mathcal{D} \quad (5)$$

In order to make the set  $\mathcal{C}$  forward invariant, we can apply any controller  $\mu(\chi) : \mathcal{D} \rightarrow U$ ,  $\mu(\chi) \in \mathcal{L}$  such that

<sup>2</sup>Formally, to make the set  $\mathcal{C}$  forward invariant (i.e., if the system starts in whatever state  $\chi(0) \in \mathcal{C}$ , then  $\chi(t) \in \mathcal{C}$ ,  $\forall t > 0$ ).

<sup>3</sup>Typically,  $\alpha(h)$  is chosen equal to a linear function  $\alpha(h(\chi)) = kh(\chi)$  or to the cube  $\alpha(h(\chi)) = kh(\chi)^3$ , with  $k > 0$ .

$\mu(\chi) \in K_{cbf}(\chi)$ . The set  $K_{cbf}(\chi)$  is defined as:

$$K_{cbf}(\chi) = \{\mu \in U \mid L_f h(\chi) + L_g h(\chi)\mu + \alpha(h(\chi)) \geq 0\}$$

It is important to notice that the closed loop controller  $\mu(\chi) \in K_{cbf}(\chi)$  does not ensure that the system will converge to the given set  $\mathcal{C}$  if it starts outside of it.

For this purpose, we can introduce a Control Lyapunov Function (CLF), which is a continuously differentiable, positive definite function  $\Phi(\chi) : \mathcal{W} \subset \mathbb{R}^p \rightarrow \mathbb{R}_0^+$  such that:

$$\inf_{\mu \in U} [L_f \Phi(\chi) + L_g \Phi(\chi)\mu] \leq -\zeta(\Phi(\chi)), \quad \forall \chi \in \mathcal{W}$$

where  $\zeta$  is a class  $\mathcal{K}$  function. Hence, it is possible to find an input  $\mu \in U$  that stabilizes a point  $\chi^* \in \mathbb{R}^p$ , or a set. More details can be found in [27].

#### IV. SYSTEM DEFINITION AND PROBLEM STATEMENT

We consider a system composed by  $N$  robots moving in a  $n$ -dimensional space, and we define the position of the  $i$ -th robot as  $x_i \in \mathbb{R}^n$ . Then we can stack all the positions in a vector, which will represent the state of the system:  $x = [x_1^T, \dots, x_N^T]^T \in \mathbb{R}^{nN}$ . We consider single-integrator dynamics:

$$\dot{x} = u \quad (6)$$

where  $u \in U \subseteq \mathbb{R}^{nN}$  is the control input of the system. Hence, (6) can be represented by the general model of an affine control system, reported in (2), with  $f(x) = \mathbb{O} \in \mathbb{R}^{nN \times nN}$ ,  $g(x) = \mathbb{I} \in \mathbb{R}^{nN \times nN}$ , where  $\mathbb{O}$  and  $\mathbb{I}$  are, respectively, the null and the identity matrix.

As in [28], we define the edge weights of the graph  $\mathcal{G}$  as:

$$a_{i,j} = \begin{cases} e^{(R^2 - d_{i,j}^2)^\sigma} - 1 & \text{if } d_{i,j} \leq R \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where  $d_{i,j} = \|x_i - x_j\|$  is the Euclidean distance between the  $i$ -th and the  $j$ -th robot,  $R > 0$  is the communication distance, and  $\sigma > 0$  is a positive constant to set the edge weight  $a_{i,j} \leq 1$ , for normalization purpose. The above definition of the weight respects the following constraints:

- $a_{i,j} \geq 0$  to have similar properties to the unweighted Laplacian [25], and  $a_{i,j} = 0$  if  $d_{i,j} > R$ ;
- continuously differentiable with respect to the distance between the robots;
- decreasing as the distance increments.

We will hereafter assume that the Laplacian matrix has simple eigenvalues, i.e.,  $\lambda_i \neq \lambda_{i+1}, \forall i \in [2, \dots, N-1]$ .

This assumption is almost always verified in the considered scenario, in which edge weights are defined, in (7), as a function of the time-varying inter-robot distance. In fact, as shown in [29], in the pathological case in which the Laplacian has multiple eigenvalues, a small perturbation (introduced, in our case, by the motion of the robots) in the edge weights is sufficient to recover the simple eigenvalues situation.

In this paper, we aim at solving the following problem: a multi-robot system, starting from an initial configuration where the group is connected ( $\lambda_2(x(0)) > 0$ ), remains

connected, while being controlled to achieve some desired objective.

From a mathematical point of view, we want to minimize the difference between the desired control input  $u_{des} \in U \subseteq \mathbb{R}^{nN}$  and the actual input  $u \in U \subseteq \mathbb{R}^{nN}$ , which should ensure the connectivity of the system, encoded by  $h(x)$ . This problem will be solved, exploiting the CBF method, by means of a Quadratic Program (QP) subject to constraints:

$$\begin{aligned} u(x) &= \operatorname{argmin}_{u \in \mathbb{R}^{nN}} \frac{1}{2} \|u - u_{des}(x)\|^2 \\ \text{s.t.} \quad & L_f h(x) + L_g h(x)u \geq -\alpha(h(x)) \\ & u(x) \in \mathcal{L} \end{aligned} \quad (8)$$

#### V. CBF FOR CONNECTIVITY MAINTENANCE

In order to solve the problem stated in Section IV, we propose the following candidate CBF:

$$h(x) = \lambda_2(x) - \epsilon \quad (9)$$

where  $\lambda_2(x)$  is the second eigenvalue of the Laplacian matrix and  $\epsilon > 0$  is introduced in order to keep connectivity above a desired value. In the following, we will omit the dependence of  $\lambda_2$  from the state  $x$ , for ease of notation.

The CBF  $h(x)$  given in (9) defines the set  $\mathcal{C}$ , described as in (3), in which we want to keep our system:

$$\mathcal{C} = \{x \in \mathbb{R}^{nN} \mid \lambda_2 \geq \epsilon\}$$

In addition, the CBF (9) is defined on a set  $\mathcal{D}$ , with  $\mathcal{C} \subset \mathcal{D}$ :

$$\mathcal{D} = \{x \in \mathbb{R}^{nN} \mid \lambda_2 > 0\}$$

This means that, in the following, we assume that the system always starts in a connected configuration.

We will now demonstrate the existence of a solution to the QP problem (8), choosing  $\alpha(h(x)) = \varphi \cdot h(x)$ , with  $\varphi > 0$ . Without loss of generality, in the following we will consider  $\varphi = 1$ , for ease of notation. Then we obtain:

$$\frac{\partial \lambda_2}{\partial x} u + (\lambda_2 - \epsilon) \geq 0 \quad (10)$$

We rename the partial derivative with  $\beta$ :

$$\beta = [\beta_1, \dots, \beta_N] = \frac{\partial \lambda_2}{\partial x} = \left[ \frac{\partial \lambda_2}{\partial x_1}, \dots, \frac{\partial \lambda_2}{\partial x_N} \right] \quad (11)$$

As shown in [14], the  $i$ -th component of  $\beta$  can be written as:

$$\beta_i = \frac{\partial \lambda_2}{\partial x_i} = \sum_{j \in \mathcal{N}_i} \frac{\partial a_{i,j}}{\partial x_i} (v_2^i - v_2^j)^2 \quad (12)$$

where  $v_2^i$  and  $v_2^j$  are the  $i$ -th and  $j$ -th component of the eigenvector associated to  $\lambda_2$ , respectively.

We will now show that at least one component of  $\beta$  is different from zero. This result is instrumental for the main results, that will be subsequently presented. We make the following assumption:

**Assumption 1** *A collision avoidance mechanism is implemented, that prevents robots from colliding among each other.*

**Lemma 1** Consider a multi-robot system with communication topology defined according to the edge weights given in (7), and consider the definition of  $\beta$  given in (11). Let the communication graph be connected. Then, a value  $k \in [1, \dots, N]$  always exists such that  $\beta_k \neq 0$ .

*Proof:* For ease of notation the demonstration is carried on component-wise, considering  $x_i \in \mathbb{R}$ . All the results can be trivially extended to the multi-dimensional case.

To prove the statement we first divide the communication graph  $\mathcal{G}$  in  $z$  subgraphs, where  $z \geq 2$ . The rule to divide the nodes in the subgraphs is their corresponding component inside the second eigenvector of the Laplacian matrix, namely  $v_2$ , which is the eigenvector associated to  $\lambda_2$ . As discussed in Section III-A, the eigenvector associated to the null eigenvalue of the Laplacian matrix is the column vector of all ones, i.e.,  $\mathbf{1}$ . Since the eigenvectors of a matrix are orthogonal, we know in particular that  $v_2$  is orthogonal to  $\mathbf{1}$ , that is:  $v_2 \neq \rho \mathbf{1}$ , where  $\rho \in \mathbb{R} \setminus \{0\}$ . Then, defining  $\gamma_1, \dots, \gamma_N \in \mathbb{R}$  as the components of  $v_2$  (e.g.,  $\gamma_1 = v_2^k$  and  $\gamma_2 = v_2^j$ ), we can always find at least two values such that  $\gamma_1 \neq \gamma_2$ .

We divide the communication graph  $\mathcal{G}$  in  $z$  subgraphs  $\mathcal{G}_i$ , where  $i = 1, \dots, z$ . Let:

$$\begin{aligned} \mathcal{G}_1 &= \{V_1, E_1\} & \dots & \mathcal{G}_z = \{V_z, E_z\} \\ V_1 &= \{\omega \in V | v_2^\omega = \gamma_1\} & \dots & V_z = \{\omega \in V | v_2^\omega = \gamma_z\} \\ E_1 &= \{e_{i,j} \in E | i, j \in V_1\} & \dots & E_z = \{e_{i,j} \in E | i, j \in V_z\} \end{aligned}$$

Hence,  $\mathcal{G} = \bigcup_{i=1}^z \mathcal{G}_i \cup \bar{\mathcal{G}}$ , where  $\bar{\mathcal{G}} = \{\bar{V}, \bar{E}\}$ ,  $V = \bigcup_{i=1}^z V_i$  and  $E = \bigcup_{i=1}^z E_i \cup \bar{E}$ .  $\bar{E}$  represents the set of edges that link the subgraphs, namely the edges whose adjacent vertices belong to different subgraphs (the set of these vertices is  $\bar{V}$ ):

$$\bar{E} = \{e_{i,j} \in E | (i \in V_s \wedge j \in V_t) \vee (i \in V_t \wedge j \in V_s)\}$$

with  $s, t \in Z$ ,  $Z = [1, \dots, z]$ .

We define  $i \in V$  an extreme node if:

$$x_i < x_j \vee x_i > x_j, \forall j \neq i \in V$$

We know that two nodes always exist that verify this definition for Assumption 1.

Consider now the subgraph  $\bar{\mathcal{G}}$  and in particular one extreme node of it, namely the  $k$ -th vertex. For this vertex we demonstrate that  $\beta_k \neq 0$ . The term  $\beta_k$  consists of a summation of terms, defined as the multiplication of two terms:  $(v_2^k - v_2^j)^2$ , and  $\frac{\partial a_{k,j}}{\partial x_k}$ . Considering the first term, namely  $(v_2^k - v_2^j)^2$ , we know that it is different from zero for all  $j \neq k \in \bar{V}$ , for the definition of the subgraphs. The second term  $\frac{\partial a_{k,j}}{\partial x_k}$  can be divided in two additional terms  $\left(\frac{\partial a_{k,j}}{\partial x_k} = \frac{\partial a_{k,j}}{\partial d_{k,j}} \frac{\partial d_{k,j}}{\partial x_k}\right)$ . For  $j \neq k \in \bar{V} \wedge j \in \mathcal{N}_k$ , we can analyze the additional terms as follows, considering the definition of  $a_{k,j}$  in (7):

- the term  $\frac{\partial a_{k,j}}{\partial d_{k,j}}$  is different from zero, since  $j \in \mathcal{N}_k$ ;
- the term  $\frac{\partial d_{k,j}}{\partial x_k}$  is different from zero if  $d_{k,j}$  is different from zero, namely if the positions of robots  $i$  and  $j$  do not coincide. This is ensured according to Assumption 1.

If the  $k$ -th vertex has just one neighbor in  $\bar{\mathcal{G}}$ , then the proof is completed.

If instead it has multiple neighbors in  $\bar{\mathcal{G}}$  we need to consider the summation of multiple terms. In particular, we know that, if the summation is composed by terms of equal sign, then the summation will be different from zero. Considering the elements that build each addend, we know that the sign is defined by the difference between the position of the two vertices. As  $x_k$  is an extreme vertex then it will have all these terms with the same sign, which ends the demonstration.  $\blacksquare$

If we can ensure that the optimization problem (8) admits always a solution, then we can guarantee that the multi-robot system stays connected during the desired task.

**Theorem 1** Given the CBF described in (9), consider the QP problem described in (8). Then, we can always find  $u \in U$  that verifies the constraint (10) of the QP problem.

*Proof:* Substituting  $\beta$ , defined in (11), in the constraint of the QP problem (10), we obtain  $\beta u \geq \epsilon - \lambda_2$ . We note that, if we can find an input  $u$  such that:

$$\beta u \geq \epsilon \quad (13)$$

then this input will also satisfy the main constraint (10) because, from the characteristics of the Laplacian matrix, we know that  $\lambda_2 \geq 0$  (Section III-A).

Inequality (13) can be rewritten in an extended version:  $\beta_1 u_1 + \beta_2 u_2 + \dots + \beta_n u_n \geq \epsilon$ . We focus on the  $k$ -th element:

$$\beta_k u_k \geq \epsilon - \sum_{i=1, i \neq k}^n \beta_i u_i \quad (14)$$

Define  $\sigma_k = \sum_{i=1, i \neq k}^n \beta_i u_i$  and rewrite (14) as:

$$\beta_k u_k \geq \epsilon - \sigma_k \quad (15)$$

If we consider an unbounded input, namely  $U = \mathbb{R}^m$ , we can always find a component  $u_k$  such that:

$$u_k \geq \frac{\epsilon - \sigma}{\beta_k} \text{ if } \beta_k > 0, \quad u_k \leq \frac{\epsilon - \sigma}{\beta_k} \text{ if } \beta_k < 0 \quad (16)$$

The definition in (16) is not well defined for  $\beta_k = 0$ . However, according to Lemma 1, it is always possible to find at least one value  $k$  for which  $\beta_k \neq 0$ .

Concluding, we can always find a value  $k \in [1, \dots, N]$  for which  $u_k$  verifies (15) and hence (10).  $\blacksquare$

We will now show that the candidate CBF given in (9) is a valid CBF.

**Lemma 2** Function  $h(x)$  introduced in (9) is a valid CBF.

*Proof:* In order to prove the statement, we need to show that  $h(x)$  in (9) satisfies Properties 1, 2 and 3 (Section III-B).

Property 1 requires  $h(x)$  to be continuously differentiable. Considering the definition given in (9), we can write

$$\frac{\partial h(x)}{\partial x} = \frac{\partial \lambda_2}{\partial x}$$

As shown in [30], simple eigenvalues of a real symmetric matrix are infinite times differentiable with respect to the variation of the element of the matrix. Considering the Laplacian matrix defined in Section III-A, such elements are the edge weights defined in (7), or their summation. The edge weights are by construction infinite times differentiable with respect to the state  $x$  of the system. Hence, Property 1 is verified.

Property 2 requires  $h(x)$  to be of relative degree one. This can be easily seen computing the first order time derivative of  $h(x)$  in (9), as follows:

$$\frac{d}{dt}h(x) = \frac{\partial \lambda_2}{\partial x} \frac{d}{dt}x = \frac{\partial \lambda_2}{\partial x} u$$

Property 3 can be demonstrated according to Theorem 1. ■

The chosen CBF (9) can also ensure an increase of  $\lambda_2$ , till at least the threshold value  $\epsilon$ , if the system starts connected. This means that, if the system starts inside  $\mathcal{D}$ , then the input  $u \in U$  that results from the QP problem will carry the system inside the set  $\mathcal{C}$ .

**Theorem 2** Let  $\Phi_{\mathcal{C}}(x) : \mathcal{D} \rightarrow \mathbb{R}$  be:

$$\Phi_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{C} \\ -h(x) & \text{if } x \in \mathcal{D} \setminus \mathcal{C} \end{cases} \quad (17)$$

Consider the CBF  $h(x)$  defined in (9). Then,  $\Phi_{\mathcal{C}}(x)$  is a CLF for the system.

*Proof:* The candidate CLF (17) is continuously differentiable and positive definite by construction. We should demonstrate that the orbital derivative  $\dot{\Phi}_{\mathcal{C}}(x)$  is negative definite or semidefinite. We evaluate  $\dot{\Phi}_{\mathcal{C}}(x)$ :

$$\dot{\Phi}_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{C} \\ -\dot{h}(x) & \text{if } x \in \mathcal{D} \setminus \mathcal{C} \end{cases}$$

We focus on the case  $x \in \mathcal{D} \setminus \mathcal{C}$ :

$$\dot{\Phi}_{\mathcal{C}}(x) = -\frac{\partial h(x)}{\partial x} \dot{x} = -\dot{h}(x) \leq \alpha(h(x))$$

where  $\alpha(\cdot)$  is a  $\mathcal{K}$  function, which is guaranteed by (5). Given the chosen CBF (9), we know that:

$$\alpha(h(x)) < 0 \text{ if } x \in \mathcal{D} \setminus \mathcal{C}$$

Combining the set of inequalities above, we get:

$$\dot{\Phi}_{\mathcal{C}}(x) < 0$$

which ensures that the  $\Phi_{\mathcal{C}}(x)$  is a CLF and asymptotically stabilizes the set  $\mathcal{C}$ . ■

## VI. EVALUATION

In this section we evaluate the proposed method considering a few different applications. In all the simulations, performed in Matlab, we imposed the threshold value  $\epsilon = 0.1$ .

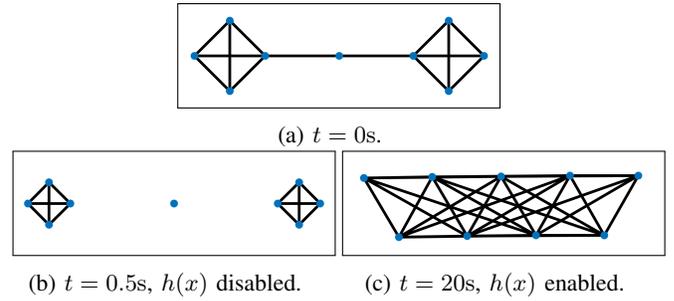
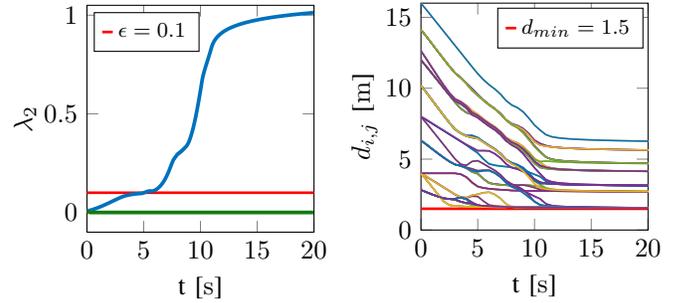


Fig. 1: Some snapshots of the consensus simulation.



(a) Value of  $\lambda_2$  for: ■  $h(x)$  enabled, ■  $h(x)$  disabled. (b) Distance between the neighboring robots.

Fig. 2: Graphs of the consensus simulation.

In order to avoid collisions among the robots, thus verifying Assumption 1, along the lines of [22], we included an additional CBF  $h_{safety}$ :

$$h_{safety}(x_i, x_j) = d_{i,j}^2 - d_{min}^2 \quad (18)$$

where  $d_{min} > 0$  represents the distance between the robots that we consider safe, based on their the moving and sensing capabilities. This additional CBF can be considered together with  $h(x)$ , defined in (9), in a cumulative CBF:

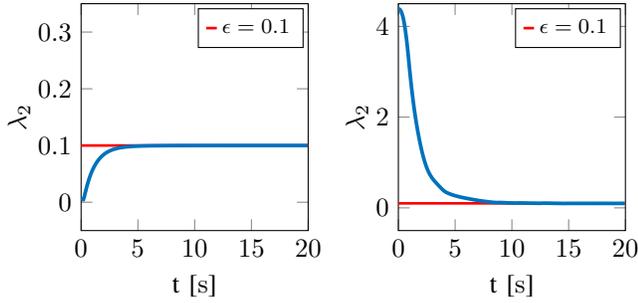
$$h_{tot} = h \wedge h_{safety} = \min\{h, h_{safety}\} \quad (19)$$

The composition of CBF is studied in [22], [31] and it will not be addressed in this paper, due to space limitations.

Extended results are shown in the attached video.

### A. Consensus

The first example is based on consensus, which is a standard control strategy for multi-robot system. It is usually referred to as an aggregative behavior, but in [11] it has been found that, in some configurations (Fig. 1(a)), it can carry to a disconnection of the group. Fig. 2(a) reports the behavior of  $\lambda_2$  with and without the CBF: it is easy to see how, without the constraint on the global connectivity, the consensus in this particular configuration leads to a disconnection of the group (Fig. 1(b)). Fig. 2(b) reports the distance between the robots and it demonstrates that the collision avoidance, together with the connectivity maintenance, can be achieved with the proposed method. The final configuration is reported in Fig. 1(c).



(a)  $N = 4$ ,  $\lambda_2(x(0)) < \epsilon$ . (b)  $N = 40$ ,  $\lambda_2(x(0)) > \epsilon$ .

Fig. 3: Value of  $\lambda_2$  in simulations with disconnecting  $u_{des}$  (20).

### B. Disconnecting control action

In this section we want to demonstrate that the proposed method can maintain connectivity against a desired controller that tends to break down the group. The desired controller is defined for the  $i$ -th robot, where  $i \in [1, \dots, N]$ , as:

$$u_{des}^i = \left[ k \cos\left(\frac{2\pi}{N+1}i\right) \quad k \sin\left(\frac{2\pi}{N+1}i\right) \right]^T \quad (20)$$

where  $k > 0$  is a tuning parameter.

We report two main trials of this series of simulations. The first has 4 robots and the system starts with  $0 < \lambda_2 < \epsilon$ , namely  $x(0) \in \mathcal{D} \setminus \mathcal{C}$ . Fig 3(a) reports the trend of  $\lambda_2$  during the simulation and, as stated in Theorem 2, the CLF defined as (17) makes the set  $\mathcal{C}$  asymptotically stable and the chosen CBF can actually increase the connectivity value up to the desired threshold.

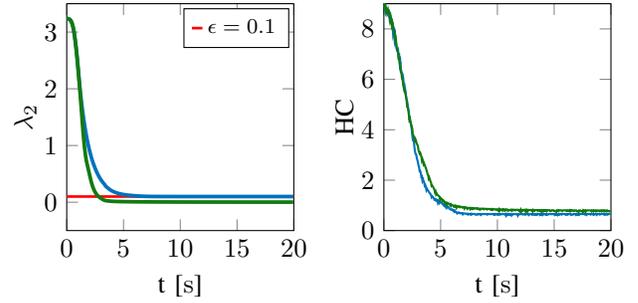
In the second example, the system, composed of 40 robots, starts inside the set  $\mathcal{C}$ . Fig. 3(b) reports the behavior of  $\lambda_2$ .

### C. Coverage

In many typical applications of multi-robot system, the main objective is searching or monitoring a certain quantity or data with ad-hoc sensors. Usually the area  $\Upsilon \in \mathbb{R}^n$  in which the robots are deployed has some points of interest, e.g., a higher probability to find victims in a search and rescue scenario or to measure the quantity that should be monitored. This different importance of the zones is usually described by means of a density function associated to the domain [9], [32], namely  $\phi(\cdot) : \Upsilon \rightarrow \mathbb{R}_0^+$ . These density functions can be defined for robot with different sensors [33], exploiting heterogeneity in multi-robot systems, or time-varying [34], to consider the evolution of the area during the task. However, spreading robots over an area can lead to loss of connectivity.

In this simulation, we solved the QP problem (8) using, as desired input, the one coming from a Voronoi-based coverage algorithm. In particular, we chose to implement the standard Lloyd approach [32], [35], which defines the input for the  $i$ -th robot as  $u_{des}^i = -k(x_i - c_i)$ , where  $k > 0$  is a tuning parameter, and  $c_i$  is the center of mass of the  $i$ -th Voronoi cell  $\mathcal{V}_i$ :

$$\mathcal{V}_i(x) = \{q \in \Upsilon \mid \|q - x_i\| \leq \|q - x_j\|, \forall j \in [1, \dots, N]\}$$



(a) Value of  $\lambda_2$ . (b) Value of  $H(x)$ .

Fig. 4: Coverage simulation with comparison between: ■  $h(x)$  and ■  $h_{local}$ .

In addition, we report a comparison with a local connectivity approach, used in [22], [23]. The approach proposed in these works is based on finding a CBF  $h_{local}$  that defines the desired set as the set that maintains all the initial links:

$$h_{local}(x_i, x_j) = R^2 - d_{i,j}^2 \quad (21)$$

This CBF, as  $h_{safety}$  (18), is defined for each pair of connected robots and not for the whole system as, vice-versa, our  $h(x)$  (9). The comparison is based on the locational cost [32], which is usually used to measure the efficiency of the coverage algorithms (lower value indicates better performance):

$$\mathcal{H}(x) = \sum_{i=1}^N \int_{\mathcal{V}_i(x)} \|q - x_i\|^2 \phi(q) dq \quad (22)$$

We report, in the following, a significant example that highlights the differences between the proposed method and the existing one. The simulation was conducted with 4 robots, starting from a connected configuration above the given threshold ( $x(0) \in \mathcal{C}$ ). Fig. 4(a) reports the behavior of the algebraic connectivity and, as might be expected, the proposed method keeps the value above the threshold, while the trial with  $h_{local}$  keeps the group connected, but with a lower value of  $\lambda_2$ . In addition, the proposed method allows to obtain a lower value of locational cost (22), which is compared to the one obtained with  $h_{local}$  in Fig. 4(b).

## VII. CONCLUSION

In this paper we propose a method to incorporate, in an optimal way, connectivity maintenance and a desired controller. The method is based on the definition of a Control Barrier Function that ensures the maintenance of the algebraic connectivity above a given value. The simulations show the feasibility and the flexibility of the method over different typical tasks. It is important to highlight that the proposed method was developed considering a centralized implementation. About the future work, we are going to implement the proposed method in a decentralized fashion. Starting from the works presented in [21], [36], where a decentralized version of the CBF is introduced, and building upon the results in [15] for decentralized estimation of the algebraic connectivity.

## REFERENCES

- [1] L. Sabattini, M. Aikio, P. Beinschob, M. Boehning, E. Cardarelli, V. Digani, A. Kregel, M. Magnani, S. Mandici, F. Oleari, C. Reinke, D. Ronzoni, C. Stimming, R. Varga, A. Vatavu, S. Castells Lopez, C. Fantuzzi, A. Mäyrä, S. Nedeveschi, C. Secchi, and K. Fuerstenberg, "The pan-robots project: Advanced automated guided vehicle systems for industrial logistics," *IEEE Robotics Automation Magazine*, vol. 25, no. 1, pp. 55–64, March 2018.
- [2] D. Ball, P. Ross, A. English, T. Patten, B. Upcroft, R. Fitch, S. Sukkarieh, G. Wyeth, and P. Corke, "Robotics for sustainable broad-acre agriculture," in *Field and Service Robotics*. Springer, 2015, pp. 439–453.
- [3] H. Hajieghrary, M. A. Hsieh, and I. B. Schwartz, "Multi-agent search for source localization in a turbulent medium," *Physics Letters A*, vol. 380, no. 20, pp. 1698–1705, 2016.
- [4] S. Kemna, H. Heiarsson, and G. S. Sukhatme, "On-board adaptive informative sampling for auvs: a feasibility study," in *Proc. 2018 MTS/IEEE OCEANS*. IEEE, 2018, pp. 1–10.
- [5] M. Saska, T. Baca, J. Thomas, J. Chudoba, L. Preucil, T. Krajník, J. Faigl, G. Loianno, and V. Kumar, "System for deployment of groups of unmanned micro aerial vehicles in gps-denied environments using onboard visual relative localization," *Autonomous Robots*, vol. 41, no. 4, pp. 919–944, 2017.
- [6] S. Hayat, E. Yanmaz, T. X. Brown, and C. Bettstetter, "Multi-objective uav path planning for search and rescue," in *Proc. 2017 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5569–5574.
- [7] D. Portugal and R. Rocha, "A survey on multi-robot patrolling algorithms," in *Doctoral Conf. on Computing, Electrical and Industrial Systems*. Springer, 2011, pp. 139–146.
- [8] F. Pasqualetti, A. Franchi, and F. Bullo, "On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms," *IEEE Trans. on Robotics*, vol. 28, no. 3, pp. 592–606, 2012.
- [9] J. Cortés, "Coverage optimization and spatial load balancing by robotic sensor networks," *IEEE Trans. on Automatic Control*, vol. 55, no. 3, pp. 749–754, 2010.
- [10] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Trans. on Robotics*, vol. 21, no. 3, pp. 376–386, 2005.
- [11] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Trans. on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.
- [12] D. V. Dimarogonas and K. H. Johansson, "Decentralized connectivity maintenance in mobile networks with bounded inputs," in *Proc. 2008 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2008, pp. 1507–1512.
- [13] A. Ajorlou, A. Momeni, and A. G. Aghdam, "A class of bounded distributed control strategies for connectivity preservation in multi-agent systems," *IEEE Trans. on Automatic Control*, vol. 55, no. 12, pp. 2828–2833, 2010.
- [14] 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.
- [15] L. Sabattini, N. Chopra, and C. Secchi, "Decentralized connectivity maintenance for cooperative control of mobile robotic systems," *The Int. J. of Robotics Research*, vol. 32, no. 12, pp. 1411–1423, 2013.
- [16] X. Li, D. Sun, and J. Yang, "A bounded controller for multirobot navigation while maintaining network connectivity in the presence of obstacles," *Automatica*, vol. 49, no. 1, pp. 285–292, 2013.
- [17] 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 Robotics and Automation Letters*, vol. 4, no. 2, pp. 633–640, 2019.
- [18] 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. on Decision and Control*. IEEE, 2014, pp. 6271–6278.
- [19] X. Xu, T. Waters, D. Pickem, P. Glotfelter, M. Egerstedt, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Realizing simultaneous lane keeping and adaptive speed regulation on accessible mobile robot testbeds," in *Proc. 2017 IEEE Conf. on Control Technology and Applications (CCTA)*. IEEE, 2017, pp. 1769–1775.
- [20] C. T. Landi, F. Ferraguti, S. Costi, M. Bonfè, and C. Secchi, "Safety barrier functions for human-robot interaction with industrial manipulators," in *Proc. 18th European Control Conf. (ECC)*. IEEE, 2019, pp. 2565–2570.
- [21] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Trans. on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.
- [22] M. Egerstedt, J. N. Pauli, G. Notomista, and S. Hutchinson, "Robot ecology: Constraint-based control design for long duration autonomy," *Annual Reviews in Control*, 2018.
- [23] W. Luo and K. Sycara, "Voronoi-based coverage control with connectivity maintenance for robotic sensor networks," in *Proc. 2019 Int. Symp. on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2019.
- [24] L. Siligardi, J. Panerati, M. Kaufmann, M. Minelli, C. Ghedini, G. Beltrame, and L. Sabattini, "Robust area coverage with connectivity maintenance," in *Proc. 2019 Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2202–2208.
- [25] B. Mohar, Y. Alavi, G. Chartrand, and O. Oellermann, "The laplacian spectrum of graphs," *Graph theory, combinatorics, and applications*, vol. 2, no. 871–898, p. 12, 1991.
- [26] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.
- [27] H. K. Khalil, "Nonlinear systems," *Upper Saddle River*, 2002.
- [28] A. Gasparri, L. Sabattini, and G. Ulivi, "Bounded control law for global connectivity maintenance in cooperative multi-robot systems," *IEEE Trans. on Robotics*, vol. 33, no. 3, pp. 700–717, June 2017.
- [29] C. Poinard, T. Pereira, and J. P. Pade, "Spectra of laplacian matrices of weighted graphs: structural genericity properties," *SIAM J. on Applied Mathematics*, vol. 78, no. 1, pp. 372–394, 2018.
- [30] J. R. Magnus, "On differentiating eigenvalues and eigenvectors," *Econometric Theory*, vol. 1, no. 2, pp. 179–191, 1985.
- [31] P. Glotfelter, J. Cortés, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE control systems letters*, vol. 1, no. 2, pp. 310–315, 2017.
- [32] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
- [33] M. Santos, Y. Diaz-Mercado, and M. Egerstedt, "Coverage control for multirobot teams with heterogeneous sensing capabilities," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 919–925, 2018.
- [34] S. G. Lee, Y. Diaz-Mercado, and M. Egerstedt, "Multirobot control using time-varying density functions," *IEEE Trans. on Robotics*, vol. 31, no. 2, pp. 489–493, 2015.
- [35] S. Lloyd, "Least squares quantization in pcm," *IEEE Trans. on Inform. Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [36] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, "Control barrier certificates for safe swarm behavior," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015.