

This is the peer reviewed version of the following article:

Decentralized Connectivity Maintenance with Time Delays using Control Barrier Functions / Capelli, Beatrice; Fouad, Hassan; Beltrame, Giovanni; Sabattini, Lorenzo. - 2021-May:(2021), pp. 1586-1592. ( 2021 IEEE International Conference on Robotics and Automation, ICRA 2021 Xi'an, China 30 May - 5 June 2021) [10.1109/ICRA48506.2021.9561066].

Institute of Electrical and Electronics Engineers Inc.

*Terms of use:*

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

14/05/2026 05:31

(Article begins on next page)

# Decentralized Connectivity Maintenance with Time Delays using Control Barrier Functions

Beatrice Capelli<sup>1</sup>, Hassan Fouad<sup>2</sup>, Giovanni Beltrame<sup>2</sup>, Lorenzo Sabattini<sup>1</sup>

**Abstract**—Connectivity maintenance is crucial for the real world deployment of multi-robot systems, as it ultimately allows the robots to communicate, coordinate and perform tasks in a collaborative way. A connectivity maintenance controller must keep the multi-robot system connected independently from the system’s mission and in the presence of undesired real world effects such as communication delays, model errors, and computational time delays, among others. In this paper we present the implementation, on a real robotic setup, of a connectivity maintenance control strategy based on Control Barrier Functions. During experimentation, we found that the presence of communication delays has a significant impact on the performance of the controlled system, with respect to the ideal case. We propose a heuristic to counteract the effects of communication delays, and we verify its efficacy both in simulation and with physical robot experiments.

## I. INTRODUCTION

The interest for multi-robot systems is constantly increasing, in a wide range of fields, from industrial [1], to agricultural [2], marine [3], and aerial [4] applications. To be able to collaborate, robots must be able to communicate, and a controller that maintains connectivity is greatly beneficial to the implementation of multi-robot applications.

In literature, the connectivity maintenance problem is usually addressed from two different points of view: local and global. The local approach [5]–[7] preserves the local connections among the robots, resulting in a connected system, in a sort of bottom-up approach. Conversely, global connectivity [8]–[10] considers the overall robot network in a top-down approach. A detailed comparison between the two approaches can be found in [11].

A multi-robot system must also be able to perform a given task in an efficient way, on top of keeping connectivity. For this purpose, [12] presents a Control Barrier Function that meets both requirements. Control Barrier Functions (CBFs) [13] are a control technique that allows a system to simultaneously achieve its objectives while maintaining some constraints. The core of the approach is that of a minimally intrusive controller with respect to the desired one. The desired control can be generated according to the robots’ mission, e.g., coverage [14], formation control [15], flocking [16] or patrolling [17]. For multi-robot systems,

examples of constraints include energy persistence [18], collision avoidance [19], local [20] and global connectivity [12] among others.

In [19], [21] a decentralized approach was introduced, but the implementation was carried out in a centralized way, namely all the calculations were done on a central unit.

When addressing a real deployment, one must take into account time delays. It is well known that delays usually lead to instability in controlled systems, if they are not properly considered [22]. Some recent works [23], [24] addressed the delay problem in the CBF approach. In particular, in [23] time delays were approximated and the proposed solution consisted in using the predicted value of the state for the CBF, instead of the current one. In [24] the time delays were not approximated and the existence of safety functionals was investigated, but the CBF was not explicitly derived due to the complexity of the problem.

The contribution of this paper is the definition of a heuristic method to implement a CBF-based control strategy on a decentralized multi-robot setup, affected by the presence of communication time delays. Building upon the results in [12], we define a control strategy that guarantees connectivity with good performance even in the presence of communication delays.

The paper is organized as follows. Preliminary notions related to graph theory and CBFs are provided in Section II. Section III reports the system definition and the problem statement. In Section IV we introduce the proposed heuristic for the implementation of CBFs in systems that suffer from delays. The results of the experiments and the conclusions are reported in Section V and Section VI, respectively.

## II. BACKGROUND

In this section we introduce the two main theoretical instruments that we use in this work: connectivity, derived from graph theory, and CBFs.

### A. Notation

$\mathbb{R}$ ,  $\mathbb{R}_0^+$ ,  $\mathbb{R}^+$  are the set of real, real non-negative, and real positive numbers, respectively. The set of locally Lipschitz functions is  $\mathcal{L}$ . 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  $\mathcal{K}$  function if it is a class  $\mathcal{K}$  function, and it is defined on the entire real line, e.g.,  $\Omega(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$  [25].

### B. Connectivity

Graph theory is usually used to represent the communication topology of a multi-robot system. In particular, the set

<sup>1</sup> Beatrice Capelli and Lorenzo Sabattini are with the Department of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Italy {beatrice.capelli, lorenzo.sabattini}@unimore.it

<sup>2</sup> Hassan Fouad and Giovanni Beltrame are with the Department of Computer and Software Engineering, École Polytechnique de Montréal, Québec, Canada {hassan.fouad, giovanni.beltrame}@polymtl.ca

of robots are represented as a set of vertices  $V$  and a set of edges  $E$  based on the type of communication model (e.g., R-disk, line of sight, etc.). The set  $E$  consists of the edges  $e_{i,j}$  between two robots  $i$  and  $j$  that are able to communicate. In this paper we consider an R-disk communication model, and hence two robots can communicate if and only if they are within the communication distance  $R$ . In addition, we consider an undirected graph, so if the edge  $e_{i,j}$  exists, then also the edge  $e_{j,i}$  exists. The overall communication topology is represented by the communication graph  $\mathcal{G} = \{V, E\}$ .

For control purposes, it is important to quantify the connectivity status of the system. The most common measure [8] used for this purpose is the algebraic connectivity or Fiedler value [26],  $\lambda_2$ . It represents the sparsity of the graph, and if  $\lambda_2 > 0$ , then the graph is connected.

Algebraic connectivity is the second smallest eigenvalue of the Laplacian matrix [27]. The Laplacian matrix is defined as  $L = D - A$ , where  $D$  is the degree matrix and  $A$  is the adjacency matrix. If we consider a group of  $N$  robots, and defining the neighbors of the  $i$ -th robot  $\mathcal{N}_i = \{j \in V | e_{i,j} \in E\}$ , we can define the adjacency matrix  $A \in \mathbb{R}^{N \times N}$  as:

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

where  $a_{i,j}$  is the edge weight of  $e_{i,j}$ . The degree matrix  $D \in \mathbb{R}^{N \times N}$  is a diagonal matrix  $D = \text{diag}\{\psi_{i,i}\}$ , where  $\psi_{i,i} = \sum_{j=1}^N a_{i,j}$ .

### C. Control Barrier Functions

We want to consider connectivity maintenance as a constraint over the general goal of a multi-robot system. CBFs are a suitable tool for this kind of problem, generating a control input that is minimally intrusive with respect to the ideal one. Consider the affine control system:

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

where  $\chi \in \mathbb{R}^p$  represents the state of the system,  $\mu \in U \subseteq \mathbb{R}^q$  is the control input, with  $U$  defined as the set of admissible inputs for the system. Moreover, we assume  $f(\chi), g(\chi) \in \mathcal{L}$ .

Now consider a desired constraint that can be expressed as a superlevel set of a continuously differentiable function  $h(\chi) : \mathbb{R}^p \rightarrow \mathbb{R}$ . The superlevel set  $\mathcal{C} \subset \mathbb{R}^p$  is called the safety set for the system and it is defined as:  $\mathcal{C} = \{\chi \in \mathbb{R}^p | h(\chi) \geq 0\}$ . The objective of the CBF is to keep the system inside the safety set, i.e., to render the set  $\mathcal{C}$  forward invariant. A set  $\mathcal{C}$  is forward invariant if, for every  $\chi_0 \in \mathcal{C}$ , then  $\chi(t) \in \mathcal{C}$  for  $\chi(0) = \chi_0$  and  $\forall t > 0$ . The system (2) is safe with respect to the set  $\mathcal{C}$  if the set  $\mathcal{C}$  is forward invariant.

The function  $h(\chi) : \mathcal{D} \subset \mathbb{R}^p \rightarrow \mathbb{R}$  is a CBF if there exists an extended class  $\mathcal{K}$  function  $\alpha(h(\chi))$  such that [13]:

$$\sup_{\mu \in U} [L_f h(\chi) + L_g h(\chi)\mu + \alpha(h(\chi))] \geq 0, \quad \forall \chi \in \mathcal{D} \quad (3)$$

where  $L_f$  and  $L_g$  represent the Lie derivatives of  $h(x)$ :  $L_f h(x) = \frac{\partial h(x)}{\partial x} f(x)$ ,  $L_g h(x) = \frac{\partial h(x)}{\partial x} g(x)$ . In order to render the system (2) safe with respect to the desired set  $\mathcal{C}$ , the control input  $\mu(\chi) : \mathcal{D} \rightarrow U$ ,  $\mu(\chi) \in \mathcal{L}$  must belong to the set  $K_{cbf}(\chi)$ , which is defined as:  $K_{cbf}(\chi) = \{\mu \in U | L_f h(\chi) + L_g h(\chi)\mu + \alpha(h(\chi)) \geq 0\}$ .

## III. SYSTEM DEFINITION AND PROBLEM STATEMENT

### A. System dynamics and control strategy

Consider a system of  $N$  robots that are able to move in a  $n$ -dimensional space. In the case of ground robots  $n = 2$  (as in the experiments reported hereafter), while in the case of aerial robots  $n = 3$ . We define the state of the system  $x = [x_1^T, \dots, x_N^T]^T \in \mathbb{R}^{nN}$ , where  $x_i \in \mathbb{R}^n$  represents the position of the  $i$ -th robot.

In addition, we assume a single integrator dynamics:

$$\dot{x} = u \quad (4)$$

where  $u \in U \subseteq \mathbb{R}^{nN}$  represents the control input of the system. It is worth remarking that, by using a sufficiently good Cartesian trajectory tracking controller, it is possible to represent the kinematic behavior of several types of mobile robots, like wheeled mobile robots [28], and UAVs [29], with (4). We can now instantiate the general affine control system, reported in (2), with  $f(x) = \mathbb{0} \in \mathbb{R}^{nN \times nN}$ ,  $g(x) = \mathbb{I} \in \mathbb{R}^{nN \times nN}$ .  $\mathbb{0}$  and  $\mathbb{I}$  represent, respectively, the null and the identity matrix of opportune dimension.

Along the lines of [12], [30], we calculate the edge weights of the graph  $\mathcal{G}$  as a function of the Euclidean distance between the  $i$ -th and the  $j$ -th robot,  $d_{i,j} = \|x_i - x_j\|$ . Considering the communication distance  $R \in \mathbb{R}^+$ , and introducing a constant  $\sigma \in \mathbb{R}^+$  for normalization purpose<sup>1</sup>, we define:

$$a_{i,j} = \begin{cases} e^{(R^2 - d_{i,j}^2)^2 / \sigma} - 1 & \text{if } d_{i,j} \leq R \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

In [12], the following CBF was introduced for connectivity maintenance:

$$h(x) = \lambda_2(x) - \varepsilon \quad (6)$$

where  $\lambda_2(x)$  is the algebraic connectivity of the system and  $\varepsilon \in \mathbb{R}^+$  is an adjustable threshold to allow different levels of action of the constraint, namely the higher  $\varepsilon$ , the higher the desired global connectivity value. The proposed CBF (6) was verified in simulation to be effective in multiple scenarios, that is considering different desired inputs  $u_{des}$ .

The general solution for this problem is obtained solving the following Quadratic Program (QP):

$$\begin{aligned} u(x) = \underset{u \in \mathbb{R}^{nN}}{\text{argmin}} & \quad \frac{1}{2} \|u - u_{des}(x)\|^2 \\ \text{s.t.} & \quad \frac{\partial \lambda_2}{\partial x} u \geq -(\lambda_2 - \varepsilon) \\ & \quad u \in U \end{aligned} \quad (7)$$

<sup>1</sup>One possible choice for  $\sigma$  is  $\frac{R^4}{\log(2)}$ , in order to obtain a value  $a_{i,j} \leq 1$ .

where we set the extended class  $\mathcal{K}$  function  $\alpha(\cdot)$  equal to the identity, i.e.,  $\alpha(h(x)) = h(x)$ .

The QP problem (7) can be solved in a decentralized fashion if each robot solves the problem considering the  $i$ -th component of the first constraint. To compute  $\frac{\partial \lambda_2}{\partial x_i}$  in a distributed way, we use the results of [8], from which:

$$\frac{\partial \lambda_2}{\partial x_i} = \sum_{j \in \mathcal{N}_i} \frac{\partial a_{i,j}}{\partial x_i} (v_2^i - v_2^j)^2 \quad (8)$$

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 enable the robots to communicate their reciprocal distances to the whole group, in such a way that they can build the Laplacian matrix, and from it they can calculate the components needed in (8).

To avoid collisions, a necessary condition from [12], we used the CBF proposed in [31]:

$$h_{safety}(x_i, x_j) = d_{i,j}^2 - d_{min}^2 \quad (9)$$

where  $d_{min} \in \mathbb{R}^+$  is the minimum safe distance between robots. The reader can refer to [31], [32] for a detailed description of the composition of CBFs.

#### B. Problem statement: decentralized implementation

We consider the problem of implementing, in a fully distributed setting, the CBF-based control strategy on a team of robots. In particular, we consider the following issues:

- 1) *Decentralized implementation*: each robot computes its control input independently, based on the data available on board;
- 2) *Imperfect state knowledge*: each robot can measure its own state, and can communicate it to the rest of the robots in the group. Each robot has then access to the full state of the multi-robot system, but the information is subject to communication delays.

### IV. HEURISTIC FOR CBF IMPLEMENTATION IN THE PRESENCE OF DELAYS

As reported in Section V-B, we performed initial experiments to evaluate the behavior of the control strategy proposed in [12]. When using a decentralized multi-robot setup, the presence of communication delays cause a degradation of the system performance.

Considering a constant delay  $\tau \in \mathbb{R}^+$ , the general model of the system given in (2) can be rewritten as

$$\dot{x}(t) = f(x(t), x(t-\tau)) + g(x(t), x(t-\tau))u(t) \quad (10)$$

to explicitly take into account the delayed state information used by each robot.

As reported in [24], the presence of such delay modifies the safety set of the CBF. However, the explicit computation of this modified safety set is challenging, and not practical in the general case. To avoid the need to explicitly compute the modified safety set, we propose a heuristic.

Each robot aggregates the state  $x$  from all other robots in the network through multi-hop communication, a process that introduces time delay due to communication latency.

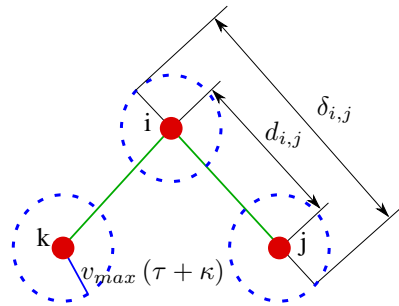


Fig. 1. Schematic for the proposed heuristic. In red the robots, in blue the possible area in which each robot can be after the time  $\tau$ , and in green the current edges.

From the state, each robot can calculate the value of  $\lambda_2$ , which, in turn, is affected by the delay. More specifically, the computation taken at time  $t$  is based on the state  $x(t-\tau)$ . Hence, the computed value of  $\lambda_2$  does not represent the *real* connectivity condition, which should use the state  $x(t)$ .

It is worth remarking that  $\lambda_2$  is a non-decreasing function of each edge weight. Considering the definition given in (5), the value of each edge weight  $a_{i,j}$  decreases, as the inter-robot distance  $d_{i,j}$  increases. Hence, we propose a heuristic based on the worst-case scenario that takes place when, during the time  $\tau$ , the increase of the distance  $d_{i,j}$  is maximum. Namely, when robots  $i$  and  $j$  move in opposite directions at the maximum of their velocity. Fig. 1 shows a schematic representation of the idea behind the proposed heuristic.

Hence, the proposed heuristic consists of redefining the edge weights in (5), replacing the measured distance  $d_{i,j}$  with a modified distance  $\delta_{i,j}$ , defined as

$$\delta_{i,j} = d_{i,j} + 2v_{max}(\tau + \kappa) \quad (11)$$

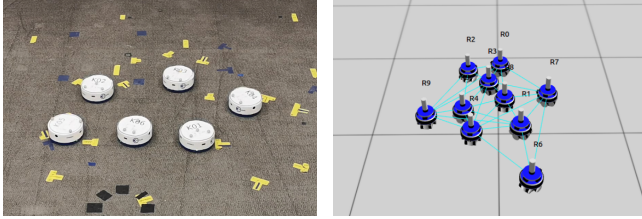
where  $v_{max}$  is the maximum velocity of the robots. We also introduce a correction factor  $\kappa \in \mathbb{R}^+$  to take into account additional disturbing factors that can affect the computation of  $\lambda_2$  from the communicated state. These disturbing factors include all the unmodelled elements that characterize the real system with respect to the ideal one, such as real robot kinematics with limited inputs, model errors, etc.

### V. EXPERIMENTAL VALIDATION

In this section we describe the experiments we carried out to demonstrate the effectiveness of the proposed control strategy, in the presence of time delays.

#### A. Hardware and software implementation

The experimental validation of the proposed connectivity maintenance strategy had been carried out on a group of K-Team Khepera IV (KH4) robots deployed inside a  $2 \times 2$  m arena equipped with an OptiTrack tracking system. The Khepera robot is equipped with a 800MHz ARM Cortex-A8 processor, mounting the Yocto operating system. The robots communicate through Wi-Fi and a software hub, `blabbermouth` which emulates range and bearing sensors, namely each robot has the (simulated) ability to measure the



(a) Experimental setup. (b) ARGoS simulator.

Fig. 2. Experimental validation.

distance and orientation between itself and its neighbors. In addition, `blabbermouth` can emulate limited communication range, packet drops, etc.

The control algorithm is implemented in Buzz [33], which is a programming language specific for swarm behaviors. In particular, during the experiments, we use the virtual stigmergy [34], a mechanism to share information through the multi-robot system. This mechanism allowed us to distribute the relative positions among all the robots. From this information, each robot could compute the value of  $\lambda_2$  and of the corresponding eigenvector of the Laplacian matrix ( $v_2$ ). These values were used to define the QP problem (7) solved on board of each robot, with the `alglib` library.

For our simulations, we used ARGoS [35], a physics-based multi-robot simulator. ARGoS supports Buzz and allowed us to use the same scripts in simulation and on the real robots.

We did not use single-integrator dynamics (4) directly: the robots have differential-drive kinematics that need an additional transformation of the velocity inputs. We used input-output state feedback linearization [36] to transform the velocity input  $u$  into the suitable velocity commands for the robots, simulated and real.

We tested the CBF for connectivity maintenance with different desired behaviors. Due to space limitations, in the following we report only the results of the two most relevant experiments. Some representative runs of the experiments are shown in the attached video. For both the simulations and the real implementation we set:  $R = 1\text{m}$ ,  $d_{min} = 0.25\text{m}$ , and  $u_{max} = 0.2\text{m/s}$ .

### B. Initial experiments

Our first experiments implement the control strategy proposed in [12] on the robots. It is worth noting that the simulation results reported in [12] were carried out in an ideal case, while the experiments in this paper consider a more realistic situation. Table I summarizes the differences.

The first desired behavior was the most challenging for the connectivity issue: the robots were controlled to disconnect (disconnecting behavior). Namely, the robots run away from each other, with a desired controller defined for the  $i$ -th robot, where  $i \in [1, \dots, N]$ , as:

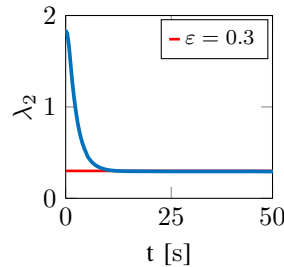
$$u_{des}^i = \left[ k \cos\left(\frac{2\pi}{N+1}i\right) \quad k \sin\left(\frac{2\pi}{N+1}i\right) \right]^T \quad (12)$$

where  $k \in \mathbb{R}^+$  is a tuning parameter.

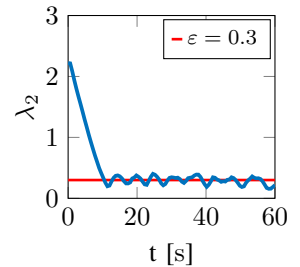
TABLE I

MAIN DIFFERENCES BETWEEN PREVIOUS SIMULATIONS, REPORTED IN [12], AND THE EXPERIMENTS OF THIS PAPER.

	Previous simulations [12]	Experiments
Implementation	Centralized	Decentralized
Positions	Absolute	Relative
Kinematics	Omni-directional	Differential-drive
Input	$u \in \mathbb{R}^{nN}$	$u \in U$



(a) Ideal conditions, as in the previous work [12] ( $N = 10$ ,  $\epsilon = 0.3$ ).



(b) Real conditions in initial experiments on real robots ( $N = 3$ ,  $\epsilon = 0.3$ ).

Fig. 3. Algebraic connectivity in ideal and real conditions with disconnecting behavior. In the real conditions, the threshold  $\epsilon$  is crossed due to the presence of delay.

Fig. 3a illustrates the behavior of  $\lambda_2$  in an ideal simulation, similar to the ones presented in [12].

This control would have led to disconnection, but, as can be seen from Fig. 3b, the proposed method reacts and prevents disconnections also in the real experiments. However, the threshold value is exceeded, namely  $\lambda_2 < \epsilon$ . It is worth noting that the oscillating behavior is caused by the non-holonomic kinematics of the robots. To reduce this issue we can add an artificial damping to the dynamics of the robots.

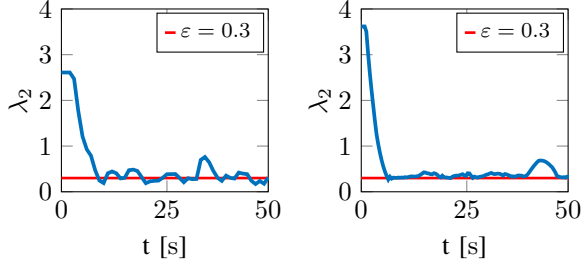
### C. Investigative simulations

To understand which disturbing factor caused the decreased performance of the control law, we reproduced the same experimental conditions in simulation with ARGoS. In particular, we add a delay in the calculation of  $\lambda_2$  and of  $v_2$ . Fig. 4a reports the trend of  $\lambda_2$  in a simulation with the disconnecting behavior as the desired behavior for the robots, and the similarity is clear with the trend in the physical robots experiment (Fig. 3b). Instead, Fig. 4b reports the trends of  $\lambda_2$  without the addition of the delay, and the behavior is approaching the ideal case reported in Fig. 3a. The small differences are probably caused by the different kinematics of the robots and the limited input.

From these simulations we deduce that the main issue is the delay, which corroborates the introduction of the heuristic proposed in Section IV.

### D. Experiments with the heuristic for delay

In this section, we test the efficacy of the heuristic proposed in (11) for the disconnecting behavior.



(a) With delay addition (delay = 0.1s, variable delay = 0). (b) Without delay addition (delay = 0s, variable delay = 0).

Fig. 4. Algebraic connectivity in the simulations for delay analysis with disconnecting behavior ( $N = 10$ ,  $\varepsilon = 0.3$ ).

1) *Simulations*: We investigated different combinations of variables: number of robots, threshold, and value of delay (possibly different for each robot). The diversification is aimed at verifying the effectiveness in a wide range of cases. For each combination of the variables we performed 20 experiments with random initial positions of the robots. Fig. 5 reports the behavior of  $\lambda_2$  in every trial for one combination of the variables.

Table II summarizes the results of the simulations in an aggregate form. In particular, for each trial we recorded the minimum value of  $\lambda_2$ : the last three columns of Table II report the mean value (*Mean*), the standard deviation (*Std*), and the minimum value (*Min*), for each combination of the parameters, reported in the first four columns of the table. In particular, the third column (*Max delay*) was the maximum delay injected in the system. In addition, if the value in the fourth column (*Delay variable*) is equal to zero, then all the robots had the same delay, if the value is one, the robots worked with random delay, uniformly chosen between the maximum value and the null value (at least one robot had delay equal to the maximum value).

It is worth noting that in just few cases (highlighted in grey in Table II) the minimum value of  $\lambda_2$  fell below the given threshold, and this happened when using variable delays. However, even in these particular cases, the proposed heuristic compensates the effect introduced by the delay, and the minimum value of  $\lambda_2$  is comparable to the one obtained in absence of delay (highlighted in light blue in Table II).

In the simulations without delay, the mean value is guaranteed to be above the threshold by the presence of the parameter  $\kappa$ , which allows to compensate the aforementioned other disturbing factors. During the simulations, the parameter  $\kappa$  was empirically tuned for the actual environment: we used the value  $\kappa = 0.04$ , which provided good results with all parameter combinations.

2) *Real experiments*: For testing the effectiveness of the proposed heuristic, we replicated the experiments of the disconnecting behavior in the real setup (described in Section V-A). We empirically tuned  $\kappa$  at the value of 0.05, and we measured that  $\tau \approx 0.3$  s. We performed several experiments with 3 robots starting in random initial positions. Fig. 6a shows the value of  $\lambda_2$  for a representative trial, with starting

TABLE II  
DATA OF THE SIMULATIONS WITH THE HEURISTIC CORRECTION  
( $\kappa = 0.04$ ).

$N$	$\varepsilon$	Delay max [s]	Delay variable	Mean	Std	Min
5	0.1	0	0	0.104	0.003	0.099
5	0.1	0.1	0	0.177	0.023	0.134
5	0.1	0.1	1	0.126	0.027	0.081
5	0.1	0.05	0	0.126	0.013	0.109
5	0.1	0.05	1	0.123	0.079	0.082
5	0.3	0	0	0.310	0.008	0.295
5	0.3	0.1	0	0.458	0.046	0.400
5	0.3	0.1	1	0.370	0.083	0.265
5	0.3	0.05	0	0.359	0.025	0.315
5	0.3	0.05	1	0.321	0.026	0.277
10	0.1	0	0	0.103	0.005	0.088
10	0.1	0.1	0	0.213	0.030	0.149
10	0.1	0.1	1	0.129	0.041	0.075
10	0.1	0.05	0	0.136	0.015	0.114
10	0.1	0.05	1	0.105	0.016	0.078
10	0.3	0	0	0.317	0.009	0.299
10	0.3	0.1	0	0.603	0.059	0.492
10	0.3	0.1	1	0.433	0.100	0.272
10	0.3	0.05	0	0.340	0.031	0.343
10	0.3	0.05	1	0.326	0.042	0.262

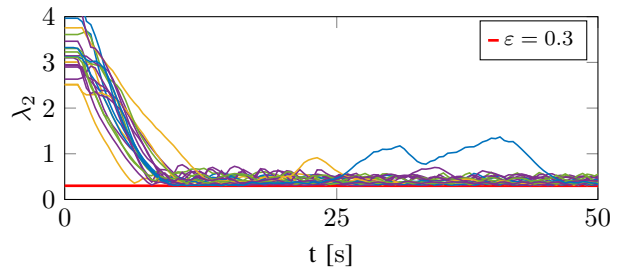
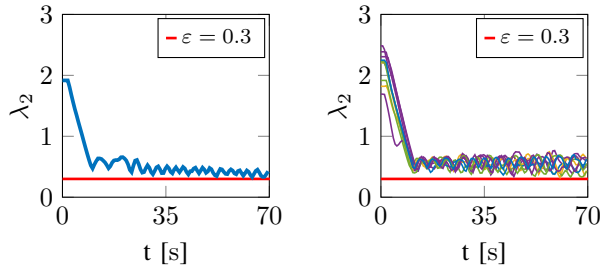


Fig. 5. Algebraic connectivity in 20 simulations with the proposed heuristic and disconnecting behavior ( $N = 10$ ,  $\varepsilon = 0.3$ , delay = 0.05 s, variable delay = 1).

positions of the robots similar to the ones of the experiment reported in Fig. 3b. With the introduction of the heuristic, the threshold value is never exceeded, differently from what happens without the correction for the presence of delay (Fig. 3b). To further validate the efficacy of the heuristic, we replicated the experiment with different starting positions. Fig. 6b reports the behavior of  $\lambda_2$  in the ten different trials. The effect of the proposed heuristic on the calculation of  $\lambda_2$  on the robots is highlighted in Fig. 7. The actual value of  $\lambda_2$  is calculated with the ground truth data, namely the data extrapolated from the Optitrack system, which report the exact distance among the robots. Instead, the value of  $\lambda_2$  calculated on the robots is lower as a consequence of



(a) Initial positions similar to experiment reported in Fig. 3b. (b) Random initial positions (10 replications).

Fig. 6. Algebraic connectivity in the experiments on real robots with the proposed heuristic and disconnecting behavior ( $N = 3$ ,  $\varepsilon = 0.3$ ).

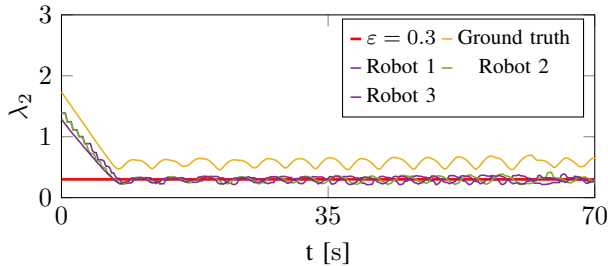


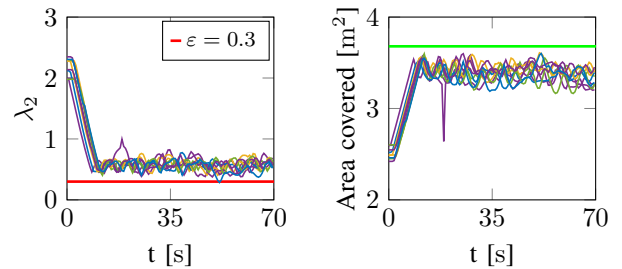
Fig. 7. Difference between the actual algebraic connectivity and the values used by the robots. ( $N = 3$ ,  $\varepsilon = 0.3$ ).

considering a worst-case distance. Finally, these experiments confirm the delay-compensation capacity of the proposed heuristic.

### E. Performance analysis

The second desired behavior is a coverage task [37], a standard problem of multi-robot systems, that we tested to investigate the effect on performance of choosing the worst-case scenario in (11). We performed a set of ten experiments, in which three Khepera IV robots used the available neighbours' positions to calculate a Voronoi tessellation using Fortune's algorithm [38], and then each robot carried out a Lloyd relaxation by chasing the center of its cell, in a manner similar to [37]. The area to be covered was  $5.29 \text{ m}^2$  and the maximum covered area achievable was  $3.68 \text{ m}^2$ , with the given threshold  $\varepsilon = 0.3$ , and with a sensing range  $R_{\text{sensing}} = 0.75 \text{ m}$ . We need to refer to a particular  $\varepsilon$  because a different choice of  $\varepsilon$  causes different performance of the system. However, the analysis of the influence of  $\varepsilon$  on the performances of the system goes beyond the scope of this paper.

Hence, to analyse the impact of the proposed heuristic, we report, in Fig. 8b, the area covered during the experiments with respect to the maximum achievable. It is clear how the maximum value is never reached, but the performance is not drastically decreased. Considering all the experiments, we obtain a mean deviation of 12.71% from the maximum area, with standard deviation equal to 3.03%. In addition, it is important to remark that, without the proposed heuristic, algebraic connectivity goes below the desired value, and the robots usually disconnect.



(a) Algebraic connectivity during the experiments. (b) Area covered with respect to the maximum area achievable with the chosen  $\varepsilon$  (reported in green).

Fig. 8. Coverage experiments with  $N = 3$ ,  $\varepsilon = 0.3$ ,  $\kappa = 0.05$  (10 replications).

## VI. CONCLUSIONS

In this paper we have presented a first implementation on real robots of the proposed Control Barrier Function for connectivity maintenance, first introduced in [12]. The experiments show that the CBF is effective in a more complex, non-ideal, decentralized scenario, but with decreased performance. We found out that the main issue is the presence of delay in the system, in particular the fact that each robot works with a delayed state of the system. To solve this problem we have proposed a heuristic approach and we have demonstrated its effectiveness in simulation and on real robots.

The proposed heuristic was tuned for the system and for the experiments considered in Section V-D, but it may give an idea on how to deal with the presence of delay in a system when using CBFs. As reported in [24], computing explicitly the safety domain in the presence of delays is challenging. Hence, we propose a heuristic approach that maintains comparable performance to the ideal case, as stated in Section V-E. The proposed heuristic was developed explicitly considering the system detailed in Section V-D, but the concept can be easily extended to more general scenarios, in which the control law is computed based on relative positions among the robots. In addition, the proposed heuristic does not increase the computation time with respect to the nominal case, since the modified inter-robot distance is trivially computed with constant factors.

It is worth noting that our implementation represents only an initial step towards real-world implementation: the Optitrack system allows to obtain very precise positions, and hence the relative measures among the robots are not affected by noise, as they would be in a real application. Including noisy measurements is part of our future work.

Moreover, besides the presence of delays, we considered an ideal communication channel (no packet drops or link failures). As future work, we aim at investigating the effect of these issues on the performance of the system. Finally, to refine an algorithm for tuning the parameters of the proposed heuristic, we want to investigate the effects of the specific communication topologies on the information propagation in the presence of delays, and possibly introduce a machine learning based algorithm for tuning the parameters.

## REFERENCES

- [1] L. Sabattini, M. Aikio, P. Beinschob, M. Boehning, E. Cardarelli, V. Digani, A. Krengel, 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. Sukkariéh, G. Wyeth, and P. Corke, "Robotics for sustainable broad-acre agriculture," in *Field and Service Robotics*. Springer, 2015, pp. 439–453.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] B. Capelli and L. Sabattini, "Connectivity maintenance: Global and optimized approach through control barrier functions," in *Proc. 2020 Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2020.
- [13] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. 18th European Control Conf. (ECC)*. IEEE, 2019, pp. 3420–3431.
- [14] 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.
- [15] L. Sabattini, C. Secchi, and C. Fantuzzi, "Arbitrarily shaped formations of mobile robots: artificial potential fields and coordinate transformation," *Autonomous Robots*, vol. 30, no. 4, p. 385, 2011.
- [16] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A. E. Eiben, and T. Vicsek, "Optimized flocking of autonomous drones in confined environments," *Science Robotics*, vol. 3, no. 20, p. eaat3536, 2018.
- [17] 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.
- [18] G. Notomista, S. F. Ruf, and M. Egerstedt, "Persistification of robotic tasks using control barrier functions," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 758–763, 2018.
- [19] 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.
- [20] L. Wang, A. D. Ames, and M. Egerstedt, "Multi-objective compositions for collision-free connectivity maintenance in teams of mobile robots," in *Proc. 2016 IEEE Conf. on Decision and Control (CDC)*. IEEE, 2016, pp. 2659–2664.
- [21] L. Wang, A. Ames, and M. Egerstedt, "Safety barrier certificates for heterogeneous multi-robot systems," in *Proc. 2016 American Control Conf. (ACC)*. IEEE, 2016, pp. 5213–5218.
- [22] C. Secchi, S. Stramigioli, and C. Fantuzzi, *Control of interactive robotic interfaces: A port-Hamiltonian approach*. Springer Science & Business Media, 2007, vol. 29.
- [23] M. Jankovic, "Control barrier functions for constrained control of linear systems with input delay," in *Proc. 2018 American Control Conf. (ACC)*. IEEE, 2018, pp. 3316–3321.
- [24] G. Orosz and A. D. Ames, "Safety functionals for time delay systems," in *Proc. 2019 American Control Conf. (ACC)*. IEEE, 2019, pp. 4374–4379.
- [25] H. K. Khalil, "Nonlinear systems," *Upper Saddle River*, 2002.
- [26] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak mathematical journal*, vol. 23, no. 2, pp. 298–305, 1973.
- [27] 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.
- [28] R. Soukieh, I. Shames, and B. Fidan, "Obstacle avoidance of robotic formations based on fluid mechanical modeling," in *Proc. 2009 European Control Conf. (ECC)*. IEEE, 2009, pp. 3263–3268.
- [29] D. Lee, A. Franchi, H. I. Son, C. Ha, H. H. Bühlhoff, and P. R. Giordano, "Semiautonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 4, pp. 1334–1345, 2013.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] C. Pinciroli and G. Beltrame, "Swarm-oriented programming of distributed robot networks," *Computer*, vol. 49, no. 12, pp. 32–41, 2016.
- [34] C. Pinciroli, A. Lee-Brown, and G. Beltrame, "A tuple space for data sharing in robot swarms," in *Proc. 9th EAI Int. Conf. on Bio-inspired Inform. and Commun. Technologies (formerly BIONETICS)*, 2016, pp. 287–294.
- [35] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, *et al.*, "Argos: a modular, parallel, multi-engine simulator for multi-robot systems," *Swarm intelligence*, vol. 6, no. 4, pp. 271–295, 2012.
- [36] B. Siciliano, L. Sciacivico, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.
- [37] 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.
- [38] S. Fortune, "A sweepline algorithm for voronoi diagrams," *Algorithmica*, vol. 2, no. 1–4, p. 153, 1987.