Network Connectivity Maintenance via Nonsmooth Control Barrier Functions

Pio Ong[1*] Beatrice Capelli[2*] Lorenzo Sabattini[2] Jorge Cortés[1]

Abstract— **This paper considers the connectivity maintenance problem for multi-robot systems and proposes a continuous optimization-based controller with Nonsmooth Control Barrier Functions to achieve it. The design is based on the concept of algebraic connectivity of the interaction graph. When viewed as a function of the network state, the algebraic connectivity is not continuously differentiable, a fact neglected in previous controller designs. To illustrate the importance of this observation, we present an example of a simple multi-robot system that fails to maintain connectivity under such controllers. The insights gained allow us to synthesize an optimization-based controller that prescribes that all the nontrivial eigenvalues of the Laplacian remain positive. Using tools from nonsmooth analysis and set-valued theory, we show that the proposed controller is continuous, thereby guaranteeing the existence of the robot trajectories for the closed-loop system and ensuring network connectivity is maintained along them.**

I. Introduction

Connectivity maintenance is a crucial requirement for real world applications of multi-robot systems [1]. While important, connectivity is not the sole objective of the multi-robot systems. Real-time optimization-based controller has the flexibility to address systems with multiple objective [2]. The overarching objective of this paper is to develop a set of constraints to represent connectivity maintenance of a multi-robot system, in a way that it may be integrated into an optimization-based controller.

Literature Review: We rely on three bodies of literature: graph theory, safety-critical control, and set-valued analysis. For multi-robot systems, the network graphs are usually defined via proximity graphs [3], which allow the connectivity to change along the robots' trajectories. In the literature, there are two main approaches to connectivity maintenance. The first is the local connectivity [4] concept that requires the maintenance of the initial network. The second and more flexible approach uses the concept in graph theory called algebraic connectivity, also known as Fiedler eigenvalue [5]. Many connectivity maintenance algorithmic solutions [6] and controllers [1] rely on ensuring the Fiedler eigenvalue is greater than zero. Unfortunately, many results do not consider the issue of the eigenvalue having multiplicity higher than one. This issue is known to cause problems and requires special care in the literature. In particular, the issue has been addressed in the estimation of algebraic connectivity. For example, in [7] the proposed method for estimation cannot retrieve the multiplicity of the eigenvalues, and it is well known that the power iteration algorithm can not converge to any value if the algebraic connectivity has multiplicity higher than one [8]. To the best of our knowledge, in the context of feedback control for connectivity maintenance, [6] is the only work that has investigated this issue. The solution provided, however, does not examine the continuity property of the feedback controller, which is necessary for guaranteeing the existence of solutions to the feedback system.

One way to preserve positiveness of a state-dependent function, along the trajectory, is via safety-critical control. The paper [9] uses the concept of barrier certificate to develop Control Barrier Functions (CBFs) for control-affine systems. The concept is improved further in [2] by relaxing the constraint inside the safe set. CBFs are used for connectivity maintenance from a local point of view in [10], and with a global perspective in [11] for robots communicating over a finite radius. The main limitation in the latter is the assumption that network trajectories always evolve in the set of points where the algebraic connectivity is continuously differentiable. As we show later, this might not always be the case as the network tries to maintain connectivity, with problematic implications. To increase the robustness of the method presented in [11], in [12] a heuristic is proposed to consider the presence of time delays, but it does not consider the issue related to higher multiplicity of algebraic connectivity. In this paper we improve the method proposed in [11] by leveraging Nonsmooth Control Barrier Functions (NCBFs) [13] to deal with nonsmoothness as multiplicity of the eigenvalues changes with the network state.

We use set-valued theory to study the continuity of the proposed controller. Continuity is an important property for guaranteeing the existence of solutions to feedback system. The paper [14] studies the smoothness properties of optimization-based controllers with CBFs using perturbation theory. However, the result is limited to continuously differentiable CBF constraints, which is not applicable to our problem. Interpreting control constraint sets as set-valued maps, optimization-based controllers can be viewed as a parametric optimization problem, with the system state as the parameter. An important and well-known result in this regard is [15, Thm. 17.31]. The work [16] studies the continuity of optimization-based controllers using set-valued theory, but the results are restricted to affine constraints.

Statement of Contributions: We present a NCBF-based controller for global connectivity maintenance. We consider

[1] Pio Ong and Jorge Cortés are with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, USA {piong,cortes}@eng.ucsd.edu

[2] Beatrice Capelli and Lorenzo Sabattini are with the Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Italy {beatrice.capelli, lorenzo.sabattini}@unimore.it

[*] Both authors contributed equally.

a multi-robot system with a fully actuated first-order dynamics for which it is desirable for the robots to remain connected. The graph defining the connections between them is proximity-based, meaning that it is state-dependent and can change as the robots move. We design the control input constraint that will ensure connectivity maintenance among the robots. The contributions are threefold. The first is a novel controller design that manages to be well defined everywhere by resorting to the concept of NCBFs to deal with situations in which the algebraic connectivity is not simple. Our second contribution is the generalization of the design to arbitrary continuously differentiable proximity-based graphs. As a result, the controller becomes much more suitable for real multi-robot systems with proximity-based graphs tied to physical meanings. Finally, our third contribution is the characterization of the continuity of the designed constraint. Hence, when the proposed control input constraint is utilized in an optimization-based controller, continuity of the feedback controller is ensured, thereby establishing the existence of a system solution. Proofs are omitted for reasons of space and will appear elsewhere.

II. Continuity of Optimization-based Controllers

In this section we provide a general discussion on optimization-based controllers and their versatility to incorporate safety and stability constraints[1]. This provides the necessary context for the problem formulation described later. Consider the nonlinear system $\dot{x} = f(x,u)$, where $x \in \mathbb{R}^N$, $u \in \mathbb{R}^M$ and $f : \mathbb{R}^N \times \mathbb{R}^M \to \mathbb{R}^N$. Broadly speaking, we seek to synthesize a controller $x \mapsto k(x)$ such that the closed-loop system enjoys some desirable performance and asymptotic guarantees. A commonly used design methodology resorts to the optimization-based controller,

$$k(x) = \underset{u \in \mathcal{U}(x)}{\operatorname{argmin}} \; J(x,u), \tag{1}$$

where $J : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ is a cost function (e.g., minimum-energy control specifications) and $\mathcal{U} : \mathbb{R}^N \rightrightarrows \mathbb{R}^M$ is a set-valued map encoding constraints (e.g., input boundedness, infinitesimal decrease of certificate) on the input.

A key property of the controller is continuity, both from a theoretical and practical viewpoint. Continuity of the controller guarantees the existence of Carathéodory solutions[2] [18, Thm. 5.1]. As one can expect, the continuity of the controller (1) depends on the continuity of the cost J and also on how continuously $\mathcal{U}(x)$ changes with x. Because $\mathcal{U}$ is a set-valued map, the latter requires care in formalizing what continuity means.

Definition 2.1: (Upper and Lower Hemicontinuity [19][3]): A set-valued map $\mathcal{U} : \mathbb{R}^N \rightrightarrows \mathbb{R}^M$ is

- upper hemicontinuous at x if for any neighborhood $\bar{\mathcal{U}}$ of $\mathcal{U}(x)$, there exists $\delta > 0$ such that, if $\|x - x'\| < \delta$ then $\mathcal{U}(x') \subset \bar{\mathcal{U}}$;
- lower hemicontinuous at x if for each $u \in \mathcal{U}(x)$ and for any sequence $\{x^k\}_{k\in\mathbb{N}}$ converging to x, there exists a sequence $\{u^k\}_{k\in\mathbb{N}}$ converging to u with $u^k \in \mathcal{U}(x^k)$.

We simply refer to the map as *upper hemicontinuous (UHC)* (resp. *lower hemicontinuous (LHC)*) if it is upper (resp. lower) hemicontinuous for all x. Unlike for single-valued functions, the two definitions are not equivalent. A set-valued map is *continuous* when it is both UHC and LHC. By Berge Maximum Theorem [15, Thm. 17.31], if the cost function J is continuous, the constraints set map $\mathcal{U}$ is continuous and compact-valued, and the minimization in (1) is single-valued, then, k is continuous.

In the present paper, we employ the set-valued map $\mathcal{U}$ to specify safety constraints via CBFs [2]. Then we restrict the inputs, at each state, to those that ensure the forward invariance of the safe set. Suppose we are given a continuously differentiable function $h : \mathbb{R}^N \to \mathbb{R}$, and the goal is to contain all system trajectories within its zero superlevel set $\mathcal{C} = \{x \in \mathbb{R}^N \mid h(x) \geq 0\}$, which represents the safe set. The idea is to constrain the choices of control input to only the ones that do not drive the state trajectory outside of $\mathcal{C}$. Since we are dealing with continuous trajectories, the only region where one really needs to worry is the boundary $\{x \in \mathbb{R}^N \mid h(x) = 0\}$, where we must ensure the Lie derivative is non-negative, i.e., $\frac{\partial h}{\partial x} f(x,u) \geq 0$, to guarantee h does not decrease from those states. This inequality defines a set of valid inputs as a function of x, i.e., a set-valued map $\mathcal{U}$ specifying constraints for the optimization-based controller. However, because the inequality only appears at the boundary, the set-valued map $\mathcal{U}$, and consequently k, is not continuous. CBFs deal with this issue by prescribing safety constraints everywhere without overconstraining the system. Formally, for all x, one requires $\frac{\partial h}{\partial x} f(x,u) \geq -\alpha(h(x))$, where α is an extended class $\mathcal{K}$ function. With this constraint, h is called a CBF. The constraint does not outright require a positive Lie derivative everywhere, but gradually becomes stricter for states closer to the boundary. This defines a set-valued map $\mathcal{U}$ that specifies a set of valid inputs for every state x by means of an inequality that changes continuously.

[1] We use the following notation. We denote by $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{R}_{>0}$ the set of natural, real, real non-negative, and real positive numbers. For $n \in \mathbb{N}$, we let $[n] = \{1, \dots, n\}$. Given $x \in \mathbb{R}^n$, $\|x\|$ is its Euclidean norm. We let $\mathbb{S}^n = \{v \in \mathbb{R}^n \mid \|v\| = 1\}$ denote the unit ball in $\mathbb{R}^n$. Given matrices $A, B \in \mathbb{R}^{n\times n}$, $A \cdot B = \sum_{i,j} A_{ij} B_{ij}$. Note that for $v \in \mathbb{R}^n$, $vv^\top \cdot A = v^\top A v$. We let $\mathcal{U} : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ denote a set-valued map, also called a correspondence [15, Chap. 17], that assigns a subset of $\mathbb{R}^m$ to each point in $\mathbb{R}^n$. A set-valued map $\mathcal{U}$ is closed-valued, convex-valued, compact-valued and has a nonempty interior if its image at each point of its domain is closed, convex, compact, and has a nonempty interior, respectively. Similarly, all set operations between set-valued maps, such as union and intersection, are performed point-wise. We let $\text{co}(\mathcal{S})$ denote the convex closure of $\mathcal{S}$. A continuous function $\alpha : \mathbb{R} \to \mathbb{R}$ is of extended class $\mathcal{K}$ if $\alpha(0) = 0$, and α is strictly increasing. For a locally Lipschitz scalar-valued function $h : \mathbb{R}^n \to \mathbb{R}$, we let $\partial_x h : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ denote its generalized gradient [17, Chap. 2]. An undirected graph is a pair $\mathcal{G} = (V, E)$, where V is the set of vertexes and $E \subset V \times V$ is the set of unordered edges, i.e., if $(i,j) \in E$ then also $(j,i) \in E$. In a weighted graph, each edge $(i,j) \in E$ has a weight $a_{i,j} \in \mathbb{R}_{\geq 0}$. The Laplacian matrix $L \in \mathbb{R}^{n\times n}$ of a weighted graph $\mathcal{G}$ is defined with entries $L_{ij} = -a_{i,j}$ if $i \neq j$, and $L_{ii} = \sum_{j=1}^n a_{i,j}$ otherwise. Given a Laplacian matrix L of a graph $\mathcal{G}$, its eigenvalues are real and nonnegative, and can be ordered, i.e., $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. For its significance, we refer to λ_2 as the *algebraic connectivity* or *Fiedler eigenvalue* [5]. In fact, $\lambda_2 > 0$ if and only if the graph $\mathcal{G}$ is connected.

[2] A Carathéodory solution is an absolutely continuous trajectory that satisfies the dynamics at almost every time in the sense of Lebesgue measure.

[3] Sometimes referred to as semicontinuity, see e.g. [20].

We point out that it is possible for the function h to not have a well-defined gradient and still carry out a similar discussion using NCBFs [13]. In such case, h being locally Lipschitz is sufficient to reason with its generalized gradient.

III. Problem Statement

We consider a group of n robots moving in a d-dimensional space. Each robot is modeled by a first-order fully-actuated dynamics of the form

$$\dot{x}_i = u_i, \tag{2}$$

where $x_i \in \mathbb{R}^d$ and $u_i \in \mathbb{R}^d$ are the position and the input of the i-th robot, respectively. The network state is represented by $x = \left[x_1^\top, \dots, x_n^\top\right]^\top \in \mathbb{R}^{nd}$, with the position of all robots stacked. The underlying interaction topology is specified by a proximity-based weighted graph $x \mapsto \mathcal{G}(x)$, which changes with the network state as a function of the communication capabilities of the robots, cf. [3]. We assume edge weights are continuously differentiable[4]. Since the graph $\mathcal{G}(x)$ changes with x, we define the state-dependent Laplacian function $L : \mathbb{R}^{nd} \to \mathbb{R}^{n\times n}$. The composition function $\lambda_m \circ L : \mathbb{R}^{nd} \to \mathbb{R}$, for $m \in [n]$, are the ordered eigenvalues of the Laplacian at the corresponding state. For convenience, and with a slight abuse of notation, we use λ_m to refer to $\lambda_m \circ L$. With this in place, network connectivity can be ensured by requiring $\lambda_2(x) \geq \varepsilon$ at all times, for some $\varepsilon \in \mathbb{R}_{>0}$. The threshold parameter provides a robustness margin in ensuring connectivity.

We assume a controller u_{des}, designed to make the network achieve some desirable coordination task, is available. Our goal is then to design a controller that is as close as possible to u_{des} while ensuring network connectivity. This goal can be encoded with the optimization-based controller (1) by using u_{des} to define the cost function J and prescribing the task of defining an appropriate constraint set-valued map $\mathcal{U}$.

Problem 3.1: (Connectivity Maintenance Constraint Design Problem): Consider the multi-robot system (2) operating with a state-feedback optimization-based controller defined by (1). Design a set-valued map constraint $\mathcal{U}$ so that the controller is continuous and the trajectories of the closed-loop system satisfy $\lambda_2(x(t)) \geq \varepsilon$ at all time. •

A. Previous Work

Here we report the solution proposed in [11] for Problem 3.1 and discuss its limitations. This serves as motivation for our developments later. Given the desired controller u_{des}, define the quadratic cost function

$$J(x, u) = \|u - u_{\text{des}}(x)\|^2. \tag{3}$$

For fixed x, the minimizer of J is precisely $u = u_{\text{des}}(x)$. Given the goal of maintaining the algebraic connectivity above a threshold value, one can take $x \mapsto h(x) = \lambda_2(x) - \varepsilon$ as a CBF and resort to the discussion of Section II regarding the specification of set-valued maps via CBFs. This gives rise to the optimization-based controller (1) designed with the following QP problem:

$$k_{QP}(x) = \underset{u\in\mathbb{R}^{nd}}{\operatorname{argmin}} \ \|u - u_{\text{des}}(x)\|^2 \tag{4}$$
$$\text{s.t.} \quad \frac{\partial \lambda_2}{\partial x}(x)u \geq -\alpha(\lambda_2(x) - \varepsilon).$$

This controller is well-defined and maintains connectivity as long as the network state remains in the domain where the algebraic connectivity λ_2 is continuously differentiable. Unfortunately, λ_2 is only Lipschitz, but not continuously differentiable everywhere. In particular, it is not differentiable at states where its multiplicity is higher than one, in which case (4) is not well-defined. Depending on the specific network scenario, such configurations might be precisely the ones the network is drawn to, as shown next.

B. Example of Nonsmoothness of Algebraic Connectivity

We report here a simulation in MATLAB® where the controller (4) steers the network to a state where the multiplicity of λ_2 increases and hence k_{QP} is not well defined, with its implementation actually leading the network to lose connectivity. Consider $n = 4$ robots moving on the plane ($d = 2$), with a weighted r-disk proximity graph and the edge weights defined as in [11]. Let α be the identity function, the threshold value $\varepsilon = 0.1$, and the communication radius $r = 3\text{m}$. The desired controller u_{des} constantly disperses the robots in different directions:

$$u^i_{\text{des}} = \left[0.5\cos\left(\tfrac{2\pi}{n+1}i\right) \ 0.5\sin\left(\tfrac{2\pi}{n+1}i\right)\right]^\top \ \forall i \in [n]. \tag{5}$$

Fig. 1 shows the evolution of λ_2 and λ_3 during the simulation, and it is clear that the two converge to the same value at $t = t^*$. When this happens, the controller (4) is not well defined and the ensuing evolution depends on the implementation choice. To continue the simulation, we use the formula in [11] which assumes the eigenvalue always remains simple. Fig. 2 shows the applied inputs, in which we observe a chattering behavior. We attribute this to the combination of the digital implementation with the fact that the controller neglects higher eigenvalues (e.g., λ_3). By focusing on λ_2, the controller does not make any effort to limit how fast λ_3 may approach the threshold ε. As such, only when λ_3 becomes as small as λ_2, the controller jumps in value to prevent it from going lower. We speculate that the chattering is due to λ_2 and λ_3 interchanging their role. Regardless, it is clear that the controller can jump in value for nearby system states x, i.e., is not continuous. In addition, Fig. 1 shows that the controller (4) proposed in [11] allows λ_2 to reach zero and fails to maintain connectivity. This example reinforces the need to come up with a solution for Problem 3.1 that is well-defined everywhere and can deal with changes in the multiplicity of λ_2. The solution should give rise to a continuous controller that can maintain connectivity without sacrificing system performance.

IV. Continuous Optimization-based Controller for Connectivity Maintenance

In this section, we address the design of an optimization-based controller's constraint that accounts for the possibility

[4]This condition is verified by common weight assignments [6], [21].

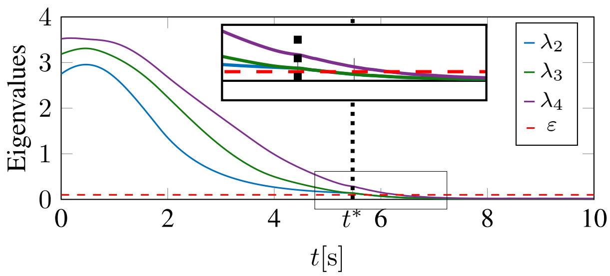

Fig. 1. Eigenvalues ($\lambda_2, \lambda_3, \lambda_4$) during the simulation. The instant t^* when λ_2 and λ_3 converge to the same value is highlighted with the black dotted vertical line.

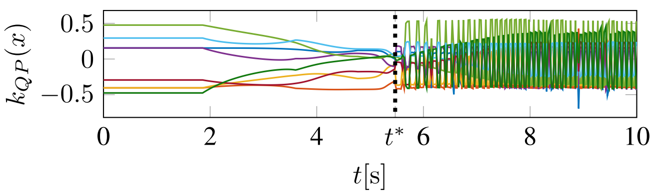

Fig. 2. The applied inputs to the robots. When the multiplicity of λ_2 increases at t^*, the controller $k_{QP}(x)$ is no longer continuous and begins to chatter. We report all the components of $k_{QP}(x)$.

of the Fiedler eigenvalue not being simple. By relying on nonsmooth analysis tools and set-valued theory, we show that the design solves Problem 3.1.

A. Accounting for Nonsmoothness of Algebraic Connectivity Does Not Necessarily Ensure Connectivity Maintenance

Here we revise the optimization-based controller (4) to account for the nonsmoothness of the algebraic connectivity. For arbitrary $m \in [n]$, the eigenvalue function λ_m is not continuously differentiable but fortunately is globally Lipschitz with respect to the Laplacian matrix (cf., [6, Lem. 1] and [22, Thm. 2.4]). In turn, since L is continuously differentiable, $\lambda_m \circ L$ is a Lipschitz function of x, so it qualifies as a NCBF candidate. In fact, the generalized gradient of the eigenvalue λ_m with respect to the matrix L is given by, cf. [6],

$$\partial_L \lambda_m(L) = \text{co}\{v_m v_m^\top \mid v_m \in \mathcal{V}_m\}, \tag{6}$$

where $\mathcal{V}_m = \{v_m \in \mathbb{S}^n \mid L v_m = \lambda_m v_m\}$ is the set of normalized eigenvectors of the Laplacian matrix L associated with λ_m. Note that, in our treatment, λ_m, $\mathcal{V}_m$, and L are all state-dependent, so the generalized gradient ultimately is a set-valued map depending on the system state x. Using the nonsmooth chain rule [17, Thm. 2.3.10], the expression for the weak set-valued Lie derivative [13, Rmk. 2.1] is

$$\mathcal{L}\lambda_m(x,u) = \partial_L \lambda_m(L(x)) \cdot \Big(\sum_{r\in[nd]} \frac{\partial L}{\partial x_r} u_r\Big)$$

Note that with a slight abuse of notation, we use the subscript r to now refer to the entries of the whole state x and u, rather than its robot index. The set-valued Lie derivative represents the possible values the eigenvalue's rate of change can take, i.e., $\frac{d\lambda_m}{dt} \in \mathcal{L}\lambda_m(x(t), u(t))$, along the trajectory. When the eigenvalue's multiplicity is larger than one, the set $\mathcal{L}\lambda_m(x)$ is no longer a singleton. This can explain the chattering behavior seen in Fig. 2 because the formula for controller (4) only uses one of the many possible rate of change, so it does not work with all possible values in $\mathcal{L}\lambda_m(x,u)$. To deal with all possible values, the logical solution is to work with the minimum value of the set-valued Lie derivative $\mathcal{L}\lambda_m(x,u)$, leading to the NCBF constraint

$$\min \mathcal{L}\lambda_m(x,u) \geq -\alpha(\lambda_m(x) - \varepsilon).$$

This is a well-defined constraint on u for all x. Using the constraint for $m = 2$, we can ensure λ_2 (and consequently all nontrivial eigenvalues) is above the threshold ε.

Lemma 4.1: (Connectivity Maintenance): For $m \in [n]$, define $(x,u) \mapsto \mu_m(x,u)$ by

$$\mu_m(x,u) = \min_{v\in\mathcal{V}_m(x)} v^\top \Big(\sum_{r\in[nd]} \frac{\partial L}{\partial x_r} u_r\Big) v. \tag{7}$$

Then $\min \mathcal{L}\lambda_m(x,u) = \mu_m(x,u)$. Furthermore, for the fully actuated multi-robot system (2), if the state-feedback controller $x \mapsto k(x)$ satisfies the inequality constraint

$$\mu_2(x, k(x)) \geq -\alpha(\lambda_2(x) - \varepsilon)$$

with a locally Lipschitz extended class $\mathcal{K}$ function α, then $\lambda_2(x(t)) \geq \varepsilon$ along all Carathéodory solutions (ensuring that network connectivity is maintained). □

Based on Lemma 4.1, we define the following optimization-based controller,

$$\begin{aligned} k_{\text{dis}}(x) = \; & \underset{u\in\mathbb{R}^{nd}}{\text{argmin}} \; J(x,u) \\ & \text{s.t.} \quad \mu_2(x,u) \geq -\alpha(\lambda_2(x) - \varepsilon). \end{aligned} \tag{8}$$

Even though the controller is now well-defined, one can still observe similar results to the ones for controller (4), where the robots eventually become disconnected (cf. Fig. 1 and 2). This behavior stems from the lack of continuity of the controller, which in turn is a consequence of the fact that μ_2 is discontinuous in x wherever the multiplicity of λ_2 changes. In particular, the feedback controller $u = k_{\text{dis}}(x)$ does not guarantee the existence of a Carathéodory solution. To make the connectivity result in Lemma 4.1 meaningful, it is important that the feedback controller is continuous.

B. Synthesis of NCBF Constraints for Connectivity

To design a continuous controller, we focus on the source of the discontinuity for controller (8). The discontinuity in μ_2 is due to the corresponding eigenspace changing dimensions, resulting in a sudden jump in the minimization value. As a result, the constraint set in (8) abruptly changes where the multiplicity changes, i.e., it is not continuous when viewed as a set-valued map. To deal with this, we need to take into account the eigenvectors of those eigenvalues that contribute to the change in multiplicity. To this end, we define

$$\mathcal{V}_{[m]}(x) = \text{span}\{\cup_{2\leq p\leq m} \mathcal{V}_p\} \cap \mathbb{S}^n,$$

which is the collection of all normalized vectors spanned by the eigenvectors associated to eigenvalues smaller than or equal to λ_m. Let

$$\mu_{[m]}(x,u) = \min_{v\in\mathcal{V}_{[m]}(x)} v^\top \Big(\sum_{r\in[nd]} \frac{\partial L}{\partial x_r} u_r\Big) v. \tag{9}$$

Note that $\mu_{[m]}(x,u) \leq \mu_m(x,u)$. One can indeed interpret $\mu_{[m]}$ as the worst-case rate of change possible among all nontrivial eigenvalues up to λ_m. Therefore, $\mu_{[m]}$, $m \in \{2,\dots,n\}$, is a more conservative approximation than μ_2 for the worst-case rate of change of λ_2. Even with such conservativeness, $\mu_{[m]}$ might still not be continuous because of the possibility of $\mathcal{V}_{[m]}$ gaining a dimension. The case $m = n$ is special, as $\mu_{[n]}$ is indeed continuous, because $\mathcal{V}_{[n]}$ is continuous since it is a constant set (the space orthonormal to the vector of all ones). We can then propose a constraint

$$\mu_{[n]}(x,u) \geq -\alpha(\lambda_2(x) - \varepsilon), \tag{10}$$

which replaces μ_2, in the constraint of (8), with $\mu_{[n]}$. Viewing this constraint as a set-valued map, it is clearly continuous. Nonetheless, it is too conservative.

To be less conservative in our constraint design, we propose instead the following controller,

$$\begin{aligned} k_{\text{con}}(x) = \underset{u\in\mathbb{R}^{nd}}{\operatorname{argmin}} \ & J(x,u) \\ \text{s.t.} \quad & \mu_{[m]}(x,u) \geq -\alpha(\lambda_m(x) - \varepsilon), \ \forall m \geq 2. \end{aligned} \tag{11}$$

The idea here is to prescribe multiple NCBF constraints, one for each eigenvalue. The constraint for λ_m will consider the rate of change for all eigenvalues lower than or equal to λ_m. To understand why the constraint here is less conservative than (10), observe that $\mu_{[m]} \leq \mu_{[n]}$ and $\lambda_2 \leq \lambda_m$ for all $m \geq 2$. Consequently, (10) is stricter than each individual constraint here. In addition, the design also utilizes the CBF concept, and hence the eigenspace for λ_m only becomes relevant when the value of λ_m is close to ε. Our ensuing discussion studies the continuity of the controller (11).

C. The Proposed Controller is Continuous

In this section we establish the continuity of the controller using Berge Maximum Theorem. To do so, we assume the following property of the cost function J.

Assumption 4.2: (Regularity of the Cost Function): The cost J is continuous for all (x,u) and strictly convex in u. •

Convexity ensures that the controller is single-valued. Together with the continuity of J, we may apply Berge Maximum Theorem if the constraint set-valued map is continuous and compact-valued. We next analyze the properties of each constraint when considered as a set-valued map.

Lemma 4.3: (Properties of Constraint Sets): Consider the set valued maps $\mathcal{U}_{[m]} : \mathbb{R}^{nd} \rightrightarrows \mathbb{R}^{nd}$ defined by

$$\mathcal{U}_{[m]}(x) = \left\{u \in \mathbb{R}^{nd} \mid \mu_{[m]}(x,u) \geq -\alpha(\lambda_m(x) - \varepsilon)\right\}.$$

Each map $\mathcal{U}_{[m]}$ is closed-valued and convex-valued. In addition, for each x, there exists an interior point common to all $\mathcal{U}_{[m]}(x)$. Consequently, each map $\mathcal{U}_{[m]}$ and every intersection among them has a nonempty interior. □

The properties from Lemma 4.3 are useful when considering the intersection of $\mathcal{U}_{[m]}$. The properties allow for their intersection to preserve continuity of the intersecting set. Particularly, we are interested in the continuity of the constraint set of (11).

Proposition 4.4: (Continuity of the Constraint Set): Consider the set-valued map $\mathcal{U}(x) = \cap_{2\leq m\leq n}\mathcal{U}_{[m]}(x)$ representing the constraint of (11). Define the set-valued map $\mathcal{J} : \mathbb{R}^{nd} \rightrightarrows \mathbb{R}$ by

$$\mathcal{J}(x) = \{u \in \mathbb{R}^{nd} \mid \|J(x,u)\| \leq \|J(x,0)\| + c\}$$

with $c \in \mathbb{R}_{>0}$. Under Assumption 4.2 and assuming the set-valued map $\mathcal{V}_{[m]}$ is continuous for all x where $\lambda_m(x) \neq \lambda_{m+1}(x)$, then $\mathcal{U} \cap \mathcal{J}$ is compact-valued and continuous. □

Despite each constraint not being continuous everywhere, Proposition 4.4 shows that the full constraint set $\mathcal{U}$ is continuous under mild assumptions, see Remark 4.6. Note that the constraint $\mathcal{J}$ is always inactive because $u = 0$ is always a feasible point, so it does not change the controller in any way, but it allows us to consider the constraint as a compact set. With Lemma 4.1 and Proposition 4.4 in place, we are ready to state our main result.

Theorem 4.5: (Continuous Connectivity Maintenance Controller): Consider the fully actuated multi-robot system (2) with a state-feedback control $x \mapsto k_{\text{con}}(x)$ defined by the optimization-based controller (11). Then $\lambda_2(x(t)) \geq \varepsilon$ along all Carathéodory solutions. Moreover, assume the cost function J satisfies Assumption 4.2 and the set-valued map $\mathcal{V}_{[m]}$ is continuous for all x where $\lambda_m(x) \neq \lambda_{m+1}(x)$. Then k_{con} is continuous. □

Theorem 4.5 ensures the continuity of the controller, given that J is properly chosen. The continuity, in turn, guarantees that a Carathéodory solution exists, for which the connectivity of the multi-robot system is maintained in a minimally invasive way. Consequently, the set-valued map $\mathcal{U}$ in (11) solves Problem 3.1. Although we only consider the connectivity maintenance constraint in the controller k_{con}, this can be extended to include other constraints. Notably, more constraints can be added while retaining the continuity of the feedback controller, as long as they are continuous, closed and convex-valued, and their intersection with $\mathcal{U}$ has a nonempty interior with $u = 0$ being a feasible point.

Remark 4.6: (Eigenspace Span Continuity Assumption): We believe that the assumption on the eigenvector span $\mathcal{V}_{[m]}$ in Theorem 4.5 is mild and might indeed hold always. The intuition comes from the fact that the set of eigenvectors changes continuously as long as the corresponding eigenvalue does not change multiplicity. In our case, we believe that the space spanned by the eigenvectors for the first m eigenvalues should also be continuous as long as it does not gain a new dimension. In fact, we can prove this is true if all the eigenvalues higher than λ_m are simple. This is done by noticing how the eigenvector span must be orthogonal to the rest of the eigenvectors. With the possibility of higher multiplicity involved, the formal justification becomes more complicated and we do not include it for space reasons. •

V. Simulation Results

To verify the effectiveness of the proposed controller (11) we replicated the same simulation reported in Section III. We used the same initial positions, the same parameters, and the same desired input u_{des}. Fig. 3 reports the behavior of the three nontrivial eigenvalues of L with the proposed

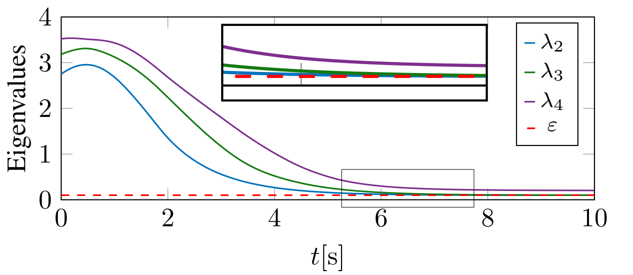

Fig. 3. Eigenvalues (λ_2, λ_3, λ_4) with the proposed controller $u = k_{con}(x)$. All the eigenvalues are constrained above ε, and for λ_2 this means that the multi-robot system remains connected.

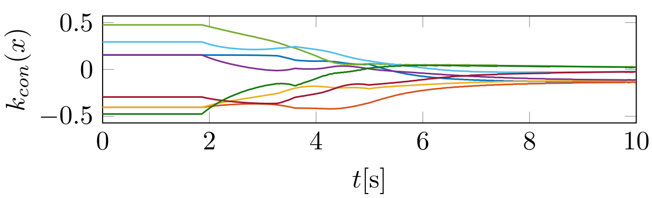

Fig. 4. The continuous controller $k_{con}(x)$, solution of (11). It is clear that even in case of higher multiplicity the controller remains continuous. We report all the components of $k_{con}(x)$.

controller. In particular, λ_2 never drops below the threshold ε, ensuring connectivity of the multi-robot system. Regarding the continuity of the controller, which has been proven in Theorem 4.5, we refer to Fig. 4, which depicts the inputs applied to the robots. Comparing with Fig. 2, it is clear how the proposed method provides a continuous controller.

In addition to the continuity of the controller, the simulation results also demonstrate the effectiveness of the proposed method. Notably, Fig. 3 shows that all eigenvalues slow down smoothly as they approach the threshold. This behavior is expected and intended as we have placed a NCBF constraint on each eigenvalue. Another important aspect to note is that, placing additional constraints does not compensate the performance of the NCBF on λ_2. By this, we mean the additional constraints do not overconstrain λ_2, if any at all. As seen in Fig. 3, λ_2 is still allowed to approach the threshold as long as it is slow in its approach.

VI. Conclusions and Future Work

We have proposed a novel optimization-based controller for connectivity maintenance of multi-robot systems using NCBFs. We have observed that the solution proposed in the literature neglects to account for the lack of smoothness of the algebraic connectivity and identified an example scenario where the controller leads the multi-robot system to problematic configurations where it becomes ill-defined. We have employed tools from nonsmooth analysis and set-valued theory to identify appropriate constraints for the optimization-based controller than can handle the nonsmoothness of the eigenvalues without being overly conservative. We have shown the continuity of the proposed controller. Future work will explore dropping the continuity assumption on eigenspace spans (cf. Remark 4.6) and study the robustness properties of the proposed controller resulting from enforcing NCBFs for each eigenvalue. Particularly, we would like to study the ability of the controller to maintain connectivity under the effect of communication and implementation delays. We also plan to develop suitable methods for implementing the controller in real time, including event-triggered implementations, and explore the design of distributed versions.

References

[1] L. Sabattini, N. Chopra, and C. Secchi, "Decentralized connectivity maintenance for cooperative control of mobile robotic systems," *The Int. Journal of Robotics Research*, vol. 32, no. 12, pp. 1411–1423, 2013.

[2] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *European Control Conference*, Jun. 2019, pp. 3420–3431.

[3] F. Bullo, J. Cortés, and S. Martinez, *Distributed control of robotic networks*, ser. Applied Mathematics Series. Princeton University Press, 2009.

[4] 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.

[5] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak mathematical journal*, vol. 23, no. 2, pp. 298–305, 1973.

[6] M. D. Schuresko and J. Cortés, "Distributed motion constraints for algebraic connectivity of robotic networks," *Journal of Intelligent and Robotic Systems*, vol. 56, no. 1, pp. 99–126, 2009.

[7] M. Franceschelli, S. Martini, M. Egerstedt, A. Bicchi, and A. Giua, "Observability and controllability verification in multi-agent systems through decentralized laplacian spectrum estimation," in *IEEE Conf. on Decision and Control*. IEEE, Dec. 2010, pp. 5775–5780.

[8] 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.

[9] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

[10] M. Egerstedt, J. N. Pauli, G. Notomista, and S. Hutchinson, "Robot ecology: Constraint-based control design for long duration autonomy," *Annual Reviews in Control*, vol. 46, pp. 1–7, 2018.

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

[12] B. Capelli, H. Fouad, G. Beltrame, and L. Sabattini, "Decentralized connectivity maintenance with time delays using control barrier functions," in *IEEE Int. Conf. on Robotics and Automation*, 2021.

[13] 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.

[14] B. J. Morris, M. J. Powell, and A. D. Ames, "Continuity and smoothness properties of nonlinear optimization-based feedback controllers," in *IEEE Conf. on Decision and Control*, Osaka, Japan, Dec 2015, pp. 151–158.

[15] C. D. Aliprantis and K. C. Border, *Infinite Dimensional Analysis: A Hitchhiker's Guide*, ser. Studies in Economic Theory. Springer, 1999.

[16] R. A. Freeman and P. V. Kototovic, *Robust Nonlinear Control Design: State-space and Lyapunov Techniques*. Cambridge, MA, USA: Birkhauser Boston Inc., 1996.

[17] F. H. Clarke, *Optimization and Nonsmooth Analysis*, ser. Canadian Mathematical Society Series of Monographs and Advanced Texts. Wiley, 1983.

[18] J. K. Hale, *Ordinary Differential Equations*. Robert E. Krieger Publishing Company, 1969.

[19] K. C. Border, *Fixed Point Theorems with Applications to Economics and Game Theory*. Cambridge University Press, 1985.

[20] A. Lechicki and A. Spakowski, "A note on intersection of lower semicontinuous multifunctions," *Proc. of the American Mathematical Society*, vol. 95, no. 1, pp. 119–122, 1985.

[21] 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.

[22] A. S. Lewis, "Group invariance and convex matrix analysis," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 4, pp. 927–949, 1996.