Robotics: Science and Systems 2022
New York City, NY, USA, June 27-July 1, 2022

Multi-Robot Adversarial Resilience using Control Barrier Functions

Matthew Cavorsi*, Beatrice Capelli†, Lorenzo Sabattini†, and Stephanie Gil*
* Harvard University
Email: mcavorsi@g.harvard.edu, sgil@seas.harvard.edu
†University of Modena and Reggio Emilia
Email: beatrice.capelli@unimore.it, lorenzo.sabattini@unimore.it

***Abstract*—In this paper we present a control barrier function-based (CBF) resilience controller that provides resilience in a multi-robot network to adversaries. Previous approaches provide resilience by virtue of specific linear combinations of multiple control constraints. These combinations can be difficult to find and are sensitive to the addition of new constraints. Unlike previous approaches, the proposed CBF provides network resilience and is easily amenable to multiple other control constraints, such as collision and obstacle avoidance. The inclusion of such constraints is essential in order to implement a resilience controller on realistic robot platforms. We demonstrate the viability of the CBF-based resilience controller on real robotic systems through case studies on a multi-robot flocking problem in cluttered environments with the presence of adversarial robots.**

I. Introduction

The problem of consensus, where robots must agree on a communicated value of interest, is at the heart of many multi-robot coordination tasks from rendezvous to flocking, and beyond [2, 8, 23, 24, 28, 31]. Unfortunately, it is well-known that consensus is vulnerable to failure whenever the robots are not cooperative, for example due to malfunctioning or adversarial action by the agents. Thus, the problem of deriving *resilient* forms of consensus has recently been the focus of increased attention in the literature [10, 18, 33]. Many recent works arrive at enhanced resilience by increasing the network connectivity and thus establishing redundant paths for the information to traverse the network [9, 18, 21, 27, 29, 33]. These controllers improve connectivity by driving the robots closer together, thereby changing the underlying network topology. While this actuation of the topology enhances resilience, it could make difficult the satisfaction of supplementary, possibly contradictory constraints, such as avoiding obstacles or inter-robot collisions. A common resolution is to employ a weighted linear combination of multiple control laws, but this results in trade-offs that are very sensitive to the choice of weights. Control barrier functions (CBFs) offer a simple, compact way to incorporate multiple control objectives and constraints while preserving certain performance guarantees. Because of this, CBFs have gained popularity in many robot applications such as environmental monitoring [22], constrained navigation [36, 37], biped robots [13], robot swarms [19], as well as autonomous vehicles [14, 16]. However, the benefits of CBFs for problems of resilience in multi-robot systems have only recently been investigated [12, 35] and this constitutes the objective of the current paper.

In this paper we apply CBFs to achieve high levels of network connectivity in order to achieve sufficient resilience to adversaries. Unlike previous works that rely on the careful linear combinations of control constraints, our CBF inherits the generality that is typical of CBF-based approaches of simultaneously considering any combination of realistic constraints that can be formulated as an inequality that is controllable by the input. For example, the authors in [4, 6] implement collision and obstacle avoidance constraints while the authors in [11] impose constraints such as coverage, energy, and battery charging using CBFs. This is done without the need to tune control weights that trade-off the different constraints, making our controller more practical to implement in real-world settings.

An inherent challenge exists, however, both for CBF-based approaches and general controllers that are resilience-aware, in that *constraints imposed by the environment often put making progress on the mission at inherent odds with maintaining resilience.* Unfortunately, these situations come up very often in practice. The presence of simple obstacles or other structures such as corridors, will limit the reachable topologies, and thus the connectivity that is attainable by the multi-robot team. For example, large obstacles in the space may force the team to split apart, whereas resilience might require agents to form more edges in the network by congregating closer together. Navigation through tight spaces, such as corridors, also often constrains robot formations to narrow chain-like configurations which can force limits on the number of connections between agents. Thus in scenarios such as these, which are commonly encountered in real-world settings, it becomes increasingly challenging to satisfy both the resilience constraint and the primary navigation objective. Understanding *what situations force this trade-off* is thus important for controller design and for understanding fundamental limits on the performance and resilience attainable by these controllers.

To this end we present two versions of the resilience CBF controller where i) resilience is treated as a *hard constraint* or where ii) resilience is treated as a *soft constraint*. Whether resilience should be treated as a hard or soft constraint is often times determined by the problem scenario. For example, treating resilience as a hard constraint would be useful in safety-critical or military applications where adversarial intervention could have catastrophic consequences. Treating

resilience as a soft constraint could be useful in search and rescue or exploration applications where resilience is helpful, but only when convenient, and should not come at the expense of navigation. However, in these cases it is still necessary to understand *at what point the environment will necessitate a tradeoff between team progress and resilience*. To address this, we derive properties of the environment, specifically the maximum obstacle size, such that agents can provably navigate and make progress on their primary task *without sacrificing resilience*. Secondly, in the soft constraint setting, we characterize a critical gain value for our controller that ensures that the *primary navigation task is provably prioritized* by the controller even at the expense of briefly sacrificing resilience. We verify these derived bounds, both in theory and in an actual multi-robot implementation, in complex settings with obstacles and narrow corridors. These scenarios represent real-world settings that a multi-robot team would commonly encounter.

The main contributions are 1) A CBF-based controller that can simultaneously consider any general combination of realistic control objectives and any constraints that can be expressed as inequalities, while guaranteeing resilience if the constraint set admits a feasible solution. 2) A CBF that can trade-off resilience for progress when it is necessary to prioritize the main mission objective. 3) Theoretical analysis of difficult scenarios, such as environments with large obstacles and narrow corridors, with closed form bounds for when such environments will necessitate trade-offs between mission progress and resilience. 4) Verification of the CBFs with aggregate results from 20 trials of a hardware experiment where robots maintain a resilient formation in order to reach a resilient heading consensus and satisfy a flocking objective while simultaneously avoiding collisions with obstacles and other robots, 2 case studies involving a large obstacle placed in the environment and a narrow corridor, and 4 simulations that verify the derived theoretical bounds.

II. Problem Formulation

We seek to develop a control law that creates sufficiently connected multi-robot networks in order to achieve resilient consensus in the presence of up to F adversaries.

Definition 1 (F-Resilient Consensus)**.** *A multi-robot network achieves* F-resilient consensus *if the legitimate robots achieve consensus to a value between the maximum and minimum of the legitimate robots' initial values, even in the presence of up to* F *adversaries.*

An *adversary* is a robot that chooses to not cooperate with the group in order to attempt to break or change the team's consensus. A *legitimate* robot is one that is not adversarial.

Consider a team of N robots moving in n-dimensional space with positions $\mathbf{x}_i \in \mathbb{R}^n$. Define the vector $\mathbf{x}$ to be an $nN \times 1$ vector of the stacked robot positions: $\mathbf{x} = \left[\mathbf{x}_1^\top, \ldots, \mathbf{x}_N^\top\right]^\top$. The robots have dynamics governed by $\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}$, where $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^q$ is the control input within some admissible set of inputs $\mathcal{U}$. For this paper, we assume single integrator dynamics: $\dot{\mathbf{x}} = \mathbf{u}$. The communication between robots is modeled by an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where the set of vertices, $\mathcal{V} = \{1, 2, \ldots, N\}$, represents the robots in the system, and the edge set, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, represents communication between robots. We consider a distance-based communication model where there exist edges $e_{i,j} \in \mathcal{E}$ and $e_{j,i} \in \mathcal{E}$ between robots i and j if they are within a distance R. We define the neighborhood of a robot i to be the robots that robot i can communicate with, i.e., $\mathcal{N}_i = \{j \in \mathcal{V} | e_{i,j} \in \mathcal{E}\}$.

The algebraic connectivity of the graph, denoted by $\lambda_2(\mathcal{G})$ [7], has the property that $\lambda_2(\mathcal{G}) > 0$ when the graph is connected, and $\lambda_2(\mathcal{G})$ increases as the graph becomes more connected. The algebraic connectivity is the second smallest eigenvalue of the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is the degree matrix and $\mathbf{A}$ is the adjacency matrix. We denote by $a_{i,j}$ the entries of the adjacency matrix, and we assume that these equal the edge weights of the edges $e_{i,j} \in \mathcal{E}$. The degree matrix is a diagonal matrix whose i^{th} diagonal element, $\psi_{i,i}$, corresponds to the sum of the i^{th} row of $\mathbf{A}$, i.e., $\psi_{i,i} = \sum_{j=1}^{N} a_{i,j}$. The Laplacian matrix has eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$, and corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N$. We define a communication function which corresponds to the elements of $\mathbf{A}$ as

$$a_{i,j} = \begin{cases} \exp\left[(R^2 - d_{i,j}^2)^2/\sigma\right] - 1, & \text{if } d_{i,j} \leq R, \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where $d_{i,j} = \|x_i - x_j\|$ is the Euclidean distance between robot i and robot j, and $\sigma > 0$ is a constant to normalize the edge weights. We choose the communication function used in [3, 4] because it is continuously differentiable with respect to the distance between robots, decreases with increasing distance, and has positive edge weights $a_{i,j} \geq 0$ when $d_{i,j} \leq R$.

In this paper, we seek to achieve resilient consensus. The authors in [18] introduced the W-MSR algorithm which can guarantee a resilient consensus is reached by the legitimate robots. In the W-MSR algorithm, each robot i forms a list from its own value, $y_i[t]$, and the values $y_j[t]$ of its neighbors $j \in \mathcal{N}_i$. Then, each robot sorts the values in ascending order and discards up to the F highest values that are greater than its own, and up to the F lowest values that are less than its own, where F is the maximum number of adversaries that the network is resilient to. Finally, each robot i updates its value using those of its neighbors which it did not discard.

The W-MSR algorithm guarantees that a resilient consensus is reached by the legitimate robots, in the presence of up to F adversarial robots, when the communication network is $(2F+1)$-robust. The r-robustness of a graph is defined below.

Definition 2 (r-reachable)**.** *A nonempty vertex set* $\mathcal{A} \in \mathcal{V}$ *is* r*-reachable if there exists a node* $i \in \mathcal{A}$ *such that* $|\delta\mathcal{A}_i| \geq r$*, where*

$$\delta\mathcal{A}_i = \{(i, j) \in \mathcal{E} : j \in \mathcal{V} \backslash \mathcal{A}\}. \tag{2}$$

Definition 3 (r-robust)**.** *A graph is* r*-robust if for each pair of disjoint sets* $\mathcal{A}_1, \mathcal{A}_2 \subset \mathcal{V}$ *at least one is* r*-reachable.*

In other words, a set is r-reachable if there is at least one node in the set with at least r edges to nodes outside the set. Then, a graph is r-robust if every pair of disjoint sets contains an r-reachable set.

The authors in [29] note that determining the r-robustness of a graph is an NP-Hard problem, and instead propose to use a lower bound on the r-robustness as a function of the algebraic connectivity of the graph. Specifically, they determine that

$$r \geq \lceil \frac{\lambda_2(\mathcal{G})}{2} \rceil, \tag{3}$$

where $\lceil \cdot \rceil$ represents the ceiling function.

We desire a CBF that uses the lower bound in (3) to maintain the sufficient connectivity required for F-resilience. We seek to express the desired resilience constraint as a superlevel set of a continuously differentiable function $h(\mathbf{x}) : \mathbb{R}^{nN} \to \mathbb{R}$. We consider the safe set to be the superlevel set $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^{nN} | h(\mathbf{x}) \geq 0\}$. The goal of the CBF is to render the set $\mathcal{C}$ forward invariant, such that for every $\mathbf{x}(0) \in \mathcal{C}$ we have that $\mathbf{x}(t) \in \mathcal{C}$ for all time $t > 0$. In particular, the function $h(\mathbf{x}) : \mathcal{D} \subset \mathbb{R}^{nN} \to \mathbb{R}$ is a CBF if there exists an extended class $\mathcal{K}$ function $\alpha(h(\mathbf{x}))$ [1] such that [1]

$$\sup_{\mathbf{u} \in \mathcal{U}} [L_f h(\mathbf{x}) + L_g h(\mathbf{x}) \mathbf{u}] \geq -\alpha(h(\mathbf{x})), \quad \forall \mathbf{x} \in \mathcal{D}, \tag{4}$$

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

In this paper, we aim to define a CBF such that the system remains within the forward invariant set $\mathcal{C}$, for all time $t \geq 0$ while trying to achieve some other desired control objective, $\mathbf{u}_{des}$. This can be done by solving a Quadratic Program (QP) subject to constraints:

$$\begin{aligned} \mathbf{u}(\mathbf{x}) = \underset{\mathbf{u} \in \mathcal{U}}{\operatorname{argmin}} \quad & \frac{1}{2} \left\| \mathbf{u} - \mathbf{u}_{des}(\mathbf{x}) \right\|^2 \\ \text{s.t.} \quad & L_f h(\mathbf{x}) + L_g h(\mathbf{x}) \mathbf{u} \geq -\alpha(h(\mathbf{x})). \end{aligned} \tag{5}$$

Problem 1. *Given a connected team of N robots, a desired control input, $\mathbf{u}_{des}$, and a resilience threshold, F, design a CBF, $h(\mathbf{x})$, to be used in the QP* (5) *such that the network remains resilient to up to F adversaries while minimally deviating from the desired control input, $\mathbf{u}_{des}$.*

In the next section, we define CBFs that solve Problem 1.

III. CBFs for F-Resilience

In this section we define a CBF that takes F-resilience as a hard constraint, such that the network must remain resilient to up to F adversaries at all times. Then, we note limitations on the maneuverability of the robot team when subject to this constraint. One specific example is when the team is tasked with navigating through a narrow corridor. The level of connectivity needed to maintain resilience cannot always be achieved when moving through a tight corridor where the team's formation is vastly constrained by the walls of the corridor, especially when robots must maintain a certain minimum distance from each other. For scenarios such as this, we propose a nested CBF approach which takes the F-resilience constraint as a soft constraint that is satisfied when possible, but can be broken in order to allow the team to navigate difficult areas such as narrow corridors. This case is treated in Section III-B.

A. F-Resilience as a Hard Constraint

First, we must determine the sufficient connectivity of a graph in order for the network to have resilience to up to F adversaries. If each legitimate robot updates its value based on the W-MSR algorithm with parameter F, then, resilient consensus is guaranteed if the graph is $(2F+1)$-robust [40, Theorem 1]. Recall the lower bound on r-robustness in (3) [29]. Let $\lambda_2(\mathbf{x}) = 4F + \epsilon$, where $\epsilon > 0$. Then, we have that $r \geq \lceil \frac{\lambda_2(\mathbf{x})}{2} \rceil = \lceil 2F + \frac{\epsilon}{2} \rceil \geq 2F + 1$. Therefore, a sufficient condition for a graph to be $(2F+1)$-robust is for its algebraic connectivity to be at least $4F + \epsilon$.

Observation 1 (Resilience to Adversaries [29])**.** *If a robot network modeled by a graph $\mathcal{G}$ has an algebraic connectivity $\lambda_2(\mathbf{x}) > 4F$, then the network can achieve resilient consensus in the presence of up to F adversaries using the W-MSR algorithm.*

The authors in [3] propose the CBF $h(\mathbf{x}) = \lambda_2(\mathbf{x}) - \epsilon$, for some $\epsilon > 0$, which guarantees that the network will remain connected by enforcing $\lambda_2(\mathbf{x}) > \epsilon$. Motivated by [3] and Observation 1, we propose the F-resilience CBF

$$h_{res}(\mathbf{x}) = \lambda_2(\mathbf{x}) - 4F - \epsilon, \tag{6}$$

with $\epsilon > 0$.

One characteristic required of (6) to be a valid CBF is that it is continuously differentiable. This is satisfied when the Laplacian matrix has simple eigenvalues, i.e., $\lambda_1 < \lambda_2 < \cdots < \lambda_N$ (see the proof of [3, Lemma 2]). The Laplacian matrix must have simple eigenvalues because $h_{res}(\mathbf{x})$ is discontinuous in $\mathbf{x}$ whenever the multiplicity of $\lambda_2(\mathbf{x})$ changes. Unfortunately, when we constrain the eigenvalues to be greater than a potentially high value such as $4F + \epsilon$, the range of values that the eigenvalues can take decreases, thus making changes in multiplicity more prominent. To handle this issue, the authors in [25, 26] propose a constraint that takes into account the eigenvectors of the eigenvalues that contribute to the discontinuities. Define the collection of normalized eigenvectors

$$\hat{\mathbf{v}}_{[m]}(\mathbf{x}) = \operatorname{span} \left\{ \bigcup_{2 \leq p \leq m} \hat{\mathbf{v}}_p \right\}, \tag{7}$$

to be the vectors spanned by the eigenvectors associated to the eigenvalues $\lambda_p \leq \lambda_m$. Furthermore, let

$$\mu_{[m]}(\mathbf{x}, \mathbf{u}) = \min_{\mathbf{v} \in \hat{\mathbf{v}}_{[m]}(\mathbf{x})} \mathbf{v}^\top \left(\sum_{r \in [nN]} \frac{\partial \mathbf{L}}{\partial \mathbf{x}_r} \mathbf{u}_r \right) \mathbf{v}, \tag{8}$$

[1] A continuous function $\alpha : \mathbb{R} \to \mathbb{R}$ is an extended class $\mathcal{K}$ function if $\alpha(0) = 0$ and α is strictly increasing.

where $r \in [nN]$ refers to the entries in the whole state $\mathbf{x}$ and $\mathbf{u}$. Here, $\mu_{[m]}(\mathbf{x}, \mathbf{u})$ can be interpreted as the worst-case rate of change possible among all nontrivial eigenvalues up to eigenvalue m.

We also wish to include further constraints to satisfy any objectives necessary for a real system implementation. These constraints include but are not limited to inter-robot collision avoidance, where robots must remain a minimum distance $d_{min,c}$ from each other, and obstacle avoidance. We denote this set of constraints by $h_{sys}(\mathbf{x})$ and derive specific constraints for collision and obstacle avoidance in Section IV. Finally, the QP in (5) can be written as

$$\begin{aligned} \mathbf{u}(\mathbf{x}) = \underset{\mathbf{u}\in\mathcal{U}}{\operatorname{argmin}} \quad & \frac{1}{2}\left\|\mathbf{u}-\mathbf{u}_{des}(\mathbf{x})\right\|^2 \\ \text{s.t.} \quad & \mu_{[m]}(\mathbf{x},\mathbf{u}) \geq \\ & -\alpha(\lambda_m(\mathbf{x}) - 4F - \epsilon), \forall m \geq 2, \\ & h_{sys}(\mathbf{x}) \geq 0. \end{aligned} \tag{9}$$

Theorem 2. *Given the CBF in* (6) *and* $F < \frac{N}{4}$*, if, for each* $\mathbf{x}$*, there exists an input* $\mathbf{u} \in \mathcal{U}$ *that satisfies the constraints in* (9)*, then the QP in* (9) *is continuous on* $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^{nN} | \lambda_2(\mathbf{x}) \geq 4F + \epsilon\}$*, and the input* $\mathbf{u}(\mathbf{x})$ *renders* $\lambda_2(\mathbf{x}) \geq 4F + \epsilon$ *for all time* $t \geq 0$*.*

Proof: The proof follows directly from the proof of [26, Theorem 5.2] by replacing ϵ with $4F + \epsilon$.

Since the algebraic connectivity is upper bounded by $\lambda_2(\mathbf{x}) \leq N$, there does not exist an input that can render $\lambda_2(\mathbf{x}) \geq 4F + \epsilon$ when $F \geq \frac{N}{4}$. Therefore, the number of adversaries is limited to $F < \frac{N}{4}$. ∎

Note that CBFs for connectivity such as (6) also work as Control Lyapunov functions (see [3] for details). This means that if the state is initially less connected than the resilience threshold requires, then the team will be driven toward the resilient state.

Adding obstacles to the environment may limit the ability of the team to maneuver in the space while remaining in a sufficiently connected formation. For example, robots that encounter a large obstacle may start to split around the obstacle to continue forward. If the obstacle is large enough, the team may not be able to pass around it from opposite sides while maintaining the resilience threshold. In this case, robots from one side of the obstacle would choose to backtrack, and join the robots on the other side to pass around the obstacle together. This could cause the team to get stuck and not successfully pass the obstacle. To avoid cases like this, it is desirable to determine a bound on the obstacle size for which the robot team can successfully navigate around it without having to backtrack.

Lemma 1. *Given a resilience threshold* $4F+\epsilon$*, communication range* R*, and team of* N *robots that split into two groups* $\mathcal{A}_{left}$ *and* $\mathcal{A}_{right}$*, forming sub-graphs* $\mathcal{G}_l$ *and* $\mathcal{G}_r$*, respectively, to move around a convex obstacle, if the obstacle has a diameter of*

$$d_{o,max} \leq \sqrt{R^2 - \left(\sigma \log\left[\frac{4F+\epsilon}{N} + 1\right]\right)^{1/2}}, \tag{10}$$

then there exists a formation where the robot team can continue around the obstacle while maintaining an algebraic connectivity of at least $4F + \epsilon$ *without having to backtrack.*

Proof: Consider the network shown in Fig. 1 where $N - i$ robots form a sub-graph ($\mathcal{G}_r$) to one side of a large obstacle of diameter $d_{o,max}$ and i robots form a sub-graph ($\mathcal{G}_l$) to the other side of the obstacle, maintaining edges to the $N - i$ robots in $\mathcal{G}_r$ with weight a_{min}. This represents any possible combination of robots choosing the left and right sides, from ($|\mathcal{A}_{left}| = 1, |\mathcal{A}_{right}| = N - 1$) to any general ($|\mathcal{A}_{left}| = i, |\mathcal{A}_{right}| = N - i$). We lower bound these sets of graphs by considering complete sub-graphs $\mathcal{G}_l$ and $\mathcal{G}_r$ where each edge weight is a_{min}, forming a fully complete graph, $\mathcal{K}_N(a_{min})$, with edge weights a_{min}. From [30], we know that the algebraic connectivity of the complete graph with edge weights of 1 is $\lambda_2(\mathcal{K}_N(1)) = N$, and when we scale all the edges to a_{min} this becomes $\lambda_2(\mathcal{K}_N(a_{min})) = Na_{min}$. The algebraic connectivity of any general graph in Fig. 1 can be lower bound by $\lambda_2(\mathbf{x}) \geq Na_{min}$ as long as the algebraic connectivity of each sub-graph is at least that of their complete sub-graph counterparts with edge weights a_{min}, i.e., $\lambda_2(\mathcal{G}_l) \geq ia_{min}$, $\lambda_2(\mathcal{G}_r) \geq (N - i)a_{min}$. This holds for most general graphs $\mathcal{G}_l$ and $\mathcal{G}_r$.

Let

$$a_{min} = \exp\left[(R^2 - d_{o,max}^2)^2/\sigma\right] - 1, \tag{11}$$

be the communication function when that robot is a distance $d_{o,max} < R$ from the complete sub-graph. Given the resilience condition $\lambda_2(\mathbf{x}) \geq 4F + \epsilon$, we can see that a sufficient condition for resilience to be maintained is that

$$\lambda_2(\mathbf{x}) \geq Na_{min} \geq 4F + \epsilon. \tag{12}$$

Substituting (11) for a_{min} into (12), we have

$$N\left(\exp\left[(R^2 - d_{o,max}^2)^2/\sigma\right] - 1\right) \geq 4F + \epsilon. \tag{13}$$

Rearranging and solving for $d_{o,max}$ gives us

$$d_{o,max} \leq \sqrt{R^2 - \left(\sigma \log\left[\frac{4F+\epsilon}{N} + 1\right]\right)^{1/2}}. \tag{14}$$

∎

Lemma 1 provides the possibility of a given robot team to split around obstacles. If larger obstacles are to be encountered, the robot team may get stuck trying to navigate around them. From (10), it can be seen that improving the communication range of the robots, or increasing the number of robots in the team can help increase the size of the obstacle that the controller can handle.

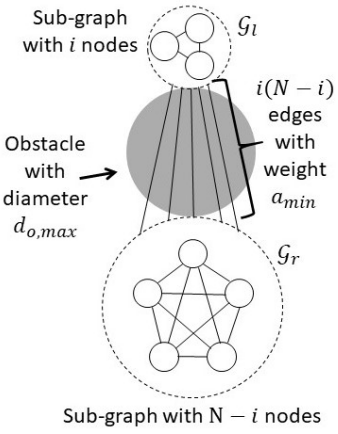

Fig. 1. Two sub-graphs $\mathcal{G}_l$ with i nodes and $\mathcal{G}_r$ with $N-i$ nodes split around a convex obstacle with diameter $d_{o,max}$ and maintain edges between each other of weight a_{min}.

B. F-Resilience as a Soft Constraint

Another example of obstacles in the environment limiting the ability of the team to maneuver the space while remaining in a sufficiently connected formation is the narrow corridor example. In this example, robots tasked with passing through a narrow corridor may need to break formation and form a line in order to fit through, thus making it impossible to maneuver through the corridor while maintaining a high connectivity. Motivated by this, we propose a nested CBF that considers the F-resilience requirement as a soft constraint that can be broken if necessary to achieve an otherwise impossible objective:

$$\begin{aligned}\mathbf{u}(\mathbf{x}) = \underset{\mathbf{u}\in\mathcal{U}}{\operatorname{argmin}} \quad & \frac{1}{2}\left\|\mathbf{u}-\tilde{\mathbf{u}}_{des}(\mathbf{x})\right\|^2 \\ \text{s.t.} \quad & \mu_{[m]}(\mathbf{x},\mathbf{u}) \geq -\alpha(\lambda_m(\mathbf{x})-\epsilon), \forall m \geq 2, \\ & h_{sys}(\mathbf{x}) \geq 0, \\ & \tilde{\mathbf{u}}_{des}(\mathbf{x}) = \tilde{\mathbf{u}}(\mathbf{x}),\end{aligned} \tag{15}$$

where $\tilde{\mathbf{u}}(\mathbf{x})$ is the solution to a nested CBF:

$$\begin{aligned}\tilde{\mathbf{u}}(\mathbf{x}) = \underset{\tilde{\mathbf{u}}\in\mathcal{U}}{\operatorname{argmin}} \quad & \frac{1}{2}\left\|\tilde{\mathbf{u}}-\gamma\mathbf{u}_{des}(\mathbf{x})\right\|^2 \\ \text{s.t.} \quad & \mu_{[m]}(\mathbf{x},\tilde{\mathbf{u}}) \geq \\ & -\alpha(\lambda_m(\mathbf{x})-4F-\epsilon), \forall m \geq 2,\end{aligned} \tag{16}$$

with a gain $\gamma \geq 1$ assigned to the desired input. The intuition here is that the nested CBF determines a control input that satisfies the F-resilience constraint while minimally deviating from the desired control input. Then, the solution of that CBF is set as the desired input for the external CBF that satisfies the hard constraints (system constraints and minimum connectivity $\lambda_2 \geq \epsilon$). In this way, the CBF will achieve all constraints if possible, but will deviate from the desired control input, which includes achieving resilience, if needed in order to satisfy the system constraints such as obstacle avoidance. The gain γ on $\mathbf{u}_{des}(\mathbf{x})$ acts as a trade-off to determine how much the resilience can be sacrificed in order to satisfy a contradictory objective. A low gain ($\gamma \approx 1$) may cause the controller to refuse to compromise the resilience enough for robots to navigate through difficult environments, such as narrow corridors. Therefore, we derive an upper bound for which the gain will always allow the CBF to compromise resilience in order to navigate the tightest corridors, without causing the robots to get stuck and stall in the corridor.

First, we need to present a few results that we will use in the analysis.

Lemma 2 (Path Graph [30])**.** *Let $\mathcal{G}_p = (\mathcal{V}, \mathcal{E}_p)$ where $\mathcal{E}_p = \{(i, i+1) : 1 \leq i < n\}$. The Laplacian, $L(\mathcal{G}_p)$, has eigenvalue $\lambda_2(\mathcal{G}_p) = 2(1-\cos(\frac{\pi}{N}))$, and corresponding eigenvector, $\mathbf{v}_{2,p} = \begin{bmatrix} v_{2,p}(1) & v_{2,p}(2) & \dots & v_{2,p}(N)\end{bmatrix}^\top$, where*

$$v_{2,p}(i) = \cos\left(\frac{i\pi}{N} - \frac{\pi}{2N}\right). \tag{17}$$

Lemma 3. *Given a team of N robots, communication function $a_{i,j}$ defined in* (1)*, and a minimum safe collision distance $d_{min,c}$. Let $\mathbf{x} = \mathbf{x}_p$ be the state of the robot team when they are in a straight line forming a path graph at a distance $d_{min,c}$ from each other. Then, $\frac{\partial\lambda_2(\mathbf{x}_p)}{\partial\mathbf{x}_p} = \begin{bmatrix}\frac{\partial\lambda_2(\mathbf{x}_p)}{\partial\mathbf{x}_{p,1}} & \frac{\partial\lambda_2(\mathbf{x}_p)}{\partial\mathbf{x}_{p,2}} & \dots & \frac{\partial\lambda_2(\mathbf{x}_p)}{\partial\mathbf{x}_{p,N}}\end{bmatrix}$, with*

$$\frac{\partial\lambda_2(\mathbf{x}_p)}{\partial\mathbf{x}_{p,i}} = \begin{cases} 4\frac{\partial a_{i,i+1}}{\partial\mathbf{x}_{p,i}}\sin^2\left(\frac{i\pi}{N}\right)\sin^2\left(\frac{\pi}{2N}\right), & \text{if } i=1, \\ 4\frac{\partial a_{i,i-1}}{\partial\mathbf{x}_{p,i}}\sin^2\left(\frac{(i-1)\pi}{N}\right)\sin^2\left(\frac{\pi}{2N}\right), & \text{if } i=N, \\ 4\frac{\partial a_{i,i+1}}{\partial\mathbf{x}_{p,i}}\sin^2\left(\frac{\pi}{2N}\right)\cdot & \\ \quad\left(\sin^2\left(\frac{i\pi}{N}\right)-\sin^2\left(\frac{(i-1)\pi}{N}\right)\right), & \text{otherwise.}\end{cases} \tag{18}$$

Proof: The authors in [38] show that

$$\frac{\partial\lambda_2(\mathbf{x}_p)}{\partial\mathbf{x}_{p,i}} = \sum_{j\in\mathcal{N}_i}\frac{\partial a_{i,j}}{\partial\mathbf{x}_{p,i}}(v_{2,p}(i)-v_{2,p}(j))^2. \tag{19}$$

From Lemma 2, we have that $v_{2,p}(i) = \cos\left(\frac{i\pi}{N}-\frac{\pi}{2N}\right)$. For a path graph, note that $\mathcal{N}_1 = \{2\}, \mathcal{N}_N = \{N-1\}$, and $\mathcal{N}_i = \{i-1, i+1\}$ for all $i \neq \{1, N\}$. Therefore, using (19), Lemma 2, and the trig identity $\cos A - \cos B = -2\sin\left(\frac{A+B}{2}\right)\sin\left(\frac{A-B}{2}\right)$, we can write $\frac{\partial\lambda_2(\mathbf{x}_p)}{\partial\mathbf{x}_{p,i}}$ as

$$\frac{\partial\lambda_2(\mathbf{x}_p)}{\partial\mathbf{x}_{p,i}} = \begin{cases} 4\frac{\partial a_{i,i+1}}{\partial\mathbf{x}_{p,i}}\sin^2\left(\frac{i\pi}{N}\right)\sin^2\left(\frac{\pi}{2N}\right), & \text{if } i=1, \\ 4\frac{\partial a_{i,i-1}}{\partial\mathbf{x}_{p,i}}\sin^2\left(\frac{(i-1)\pi}{N}\right)\sin^2\left(\frac{\pi}{2N}\right), & \text{if } i=N, \\ 4\frac{\partial a_{i,i+1}}{\partial\mathbf{x}_{p,i}}\sin^2\left(\frac{\pi}{2N}\right)\cdot & \\ \quad\left(\sin^2\left(\frac{i\pi}{N}\right)-\sin^2\left(\frac{(i-1)\pi}{N}\right)\right), & \text{otherwise.}\end{cases} \tag{20}$$

where we use the fact that $\frac{\partial a_{i,i+1}}{\partial\mathbf{x}_{p,i}} = -\frac{\partial a_{i,i-1}}{\partial\mathbf{x}_{p,i}}$ to simplify the terms $\frac{\partial\lambda_2(\mathbf{x}_p)}{\partial\mathbf{x}_{p,i}}$ for $i \neq \{1, N\}$. ■

As the algebraic connectivity deviates farther from the resilience threshold, the desire of the CBF in (16) to violate the desired input, $\mathbf{u}_{des}$, in order to improve the connectivity also increases. Therefore, if the team reaches a point where the CBF in (16) chooses an input with a desire to improve

connectivity that outweighs the team's desire to continue through the corridor, then the team will stall.

Lemma 4. *Given a team of N robots, desired control $\mathbf{u}_{des}$, resilience threshold $4F+\epsilon$, desired safe collision distance $d_{min,c}$, communication function $a_{i,j}$ defined in* (1)*, and a corridor extending in the direction θ_c. If the team uses the nested CBFs in* (15) *and* (16)*, and*

$$\gamma > \gamma_{max} = \frac{\max_i \left\{ -\left[\frac{\partial \lambda_2(\mathbf{x}_p)}{\partial \mathbf{x}_{p,i}}\right]_{\theta_c} \right\} (4F+\epsilon-\lambda_2(\mathbf{x}_p))}{\left\| \frac{\partial \lambda_2(\mathbf{x}_p)}{\partial \mathbf{x}_p} \right\|^2 \min_i \left\{ [\mathbf{u}_{des,i}]_{\theta_c} \right\} + \frac{\partial \lambda_2(\mathbf{x}_p)}{\partial \mathbf{x}_p} \mathbf{u}_{des} \max_i \left\{ -\left[\frac{\partial \lambda_2(\mathbf{x}_p)}{\partial \mathbf{x}_{p,i}}\right]_{\theta_c} \right\}}, \tag{21}$$

where $\mathbf{x}_p$ is a state describing the robots when forming a path graph inside a narrow corridor, and $[\cdot]_{\theta_c}$ is the component of a vector along the θ_c direction. Then, the robot team will always break the soft resilience constraint to pass through any narrow corridor without stalling.

Proof: Consider a corridor that is narrow enough such that one robot perfectly fits the width of the corridor. Assume that the desired input $\mathbf{u}_{des}$ tries to drive the robots through the corridor, i.e., $[\mathbf{u}_{des,i}]_{\theta_c} > 0$. The solution to the nested CBF in (16) for each robot i will be [34]

$$\tilde{\mathbf{u}}_i(\mathbf{x}) = \gamma \mathbf{u}_{des,i} + \mu \left(\frac{\partial \lambda_2(\mathbf{x})}{\partial \mathbf{x}_i} \right)^\top, \tag{22}$$

where

$$\mu = \begin{cases} 0, & \text{if } -\frac{\partial \lambda_2(\mathbf{x})}{\partial \mathbf{x}} \gamma \mathbf{u}_{des} \\ & -\lambda_2(\mathbf{x}) + 4F + \epsilon \leq 0, \\ \frac{-\frac{\partial \lambda_2(\mathbf{x})}{\partial \mathbf{x}} \gamma \mathbf{u}_{des} - \lambda_2(\mathbf{x}) + 4F + \epsilon}{\left\| \frac{\partial \lambda_2(\mathbf{x})}{\partial \mathbf{x}} \right\|^2}, & \text{otherwise.} \end{cases} \tag{23}$$

The robot team will continue moving through the corridor as long as

$$\gamma \left[\mathbf{u}_{des,i}\right]_{\theta_c} > -\mu \left[\frac{\partial \lambda_2(\mathbf{x})}{\partial \mathbf{x}_i} \right]_{\theta_c}, \tag{24}$$

for all $i \in \{1, 2, \ldots, N\}$, and all $\mathbf{x} \in \mathbb{R}^{nN}$. This is achieved if

$$\gamma \min_i \left\{ [\mathbf{u}_{des,i}]_{\theta_c} \right\} > \max_{\mathbf{x}} \left\{ \mu \max_i \left\{ -\left[\frac{\partial \lambda_2(\mathbf{x})}{\partial \mathbf{x}_i} \right]_{\theta_c} \right\} \right\}. \tag{25}$$

Since the desire to improve connectivity increases as the overall algebraic connectivity decreases, the maximum of $\mu \max_i \left\{ -\left[\frac{\partial \lambda_2(\mathbf{x})}{\partial \mathbf{x}_i} \right]_{\theta_c} \right\}$ over the states $\mathbf{x} \in \mathbb{R}^{nN}$ is achieved at the minimum possible algebraic connectivity. The minimum algebraic connectivity will occur when the team is forced to form a line, following one behind another at a distance $d_{min,c}$, in order to pass through the corridor. Without loss of generality, label the robots in the line with indices $1, 2, \ldots, N$ from left to right. This configuration can be lower bounded by that of a path graph when $d_{min,c} < R < d_{min,c}/2$. We denote the path graph configuration by the state $\mathbf{x}_p$. Therefore, it is sufficient to show that if

$$\gamma \min_i \left\{ [\mathbf{u}_{des,i}]_{\theta_c} \right\} > \mu \max_i \left\{ -\left[\frac{\partial \lambda_2(\mathbf{x}_p)}{\partial \mathbf{x}_{p,i}} \right]_{\theta_c} \right\}, \tag{26}$$

then the robot team will never stall. Plugging μ in (23) into (26) and solving for γ yields the expression in (21), where from Lemma 2 we have that $\lambda_2(\mathbf{x}_p) = 2(1-\cos(\frac{\pi}{N}))$, and $\frac{\partial \lambda_2(\mathbf{x}_p)}{\partial \mathbf{x}_p}$ is derived in Lemma 3 (18). ∎

Note that the bound on γ applies to the most restrictive connectivity situations, where the robots must form a path graph in order to pass. This corresponds to scenarios where the corridor is tight enough that only one robot can fit through it at a time, causing them to form a line, and the communication radius $R < 2d_{min,c}$ such that robots can only communicate to robots directly adjacent to them in the line. In scenarios where the connectivity is less restricted, the team has to compromise resilience much less in order to navigate through, and thus choosing a gain $\gamma \approx 1$ will likely suffice. The bound in (21) serves as a worst-case, such that the robots can navigate the environment under even the most restrictive circumstances.

IV. Defining System Constraints

We wish to implement the controller in (9) on realistic robot platforms with more complicated dynamics, such as a differential drive robot, and demonstrate the effectiveness of our CBF through multiple case studies. We apply our methods to a resilient flocking problem where a team of robots must agree upon a desired heading direction, and then move in that direction together as a flock. The team must achieve a resilient heading consensus in the presence of up to F adversaries. Along with maintaining the network connectivity constraints necessary for F-resilience, the robots must avoid collisions with each other as well as other obstacles that are laid out throughout the environment.

One of the benefits of CBF-based approaches is the simplicity for which additional constraints can be added to the QP without sacrificing much performance. In order to employ collision or obstacle avoidance, different CBFs can be used. The CBF used in [4, 6] can be used for collision avoidance:

$$h_{coll}(\mathbf{x}_i, \mathbf{x}_j) = d_{i,j}^2 - d_{min,c}^2, \tag{27}$$

where $d_{min,c} > 0$ is the minimum distance robots are allowed to have between each other. Upon assigning a minimum safe distance, a good choice for σ in (1) could be $\sigma = (R^2 - d_{min,c}^2)^2 / \log(2)$, so that $a_{i,j} = 1$ when $d_{i,j} = d_{min,c}$. Similarly, the CBF used in [11] can be used for obstacle avoidance:

$$h_{obs}(\mathbf{x}_i, \mathbf{x}_o) = d_{i,o}^2 - d_{min,o}^2, \tag{28}$$

where $d_{min,o} > 0$ is the minimum distance robots are allowed to have between themselves and the obstacle o, modeled as a point mass, with position $\mathbf{x}_o$. More details about the composition of multiple CBFs are provided in [6, 11].

It is important to note that the addition of collision avoidance can limit the level of network connectivity that the team can achieve. For example, in environments with $n = 2$ or $n = 3$ dimensions it becomes difficult to arrange more than a few robots in a formation such that each robot can be close to a distance $d_{min,c}$ from all others. This limitation is most prevalent when the connectivity of the network approaches

a complete graph, and can be minimized with smaller safe distances, $d_{min,c}$, or larger communication ranges, R.

To handle robots with different dynamics, the authors in [4] propose a distance heuristic, denoted by $\Delta_{i,j}$. This distance heuristic considers worst-case distances between robots corresponding to how much the distance could have changed over a small interval of time. For example, the distance heuristic for connectivity, which involves computing the eigenvalues and eigenvectors, would look at a worst-case scenario where any two robots have moved away from each other at maximum velocity over the small time interval. This is computed by

$$\Delta_{i,j}^{conn} = d_{i,j} + 2u_{max}(\kappa + \tau), \tag{29}$$

where u_{max} is the maximum velocity of the robots. The parameter $\kappa > 0$ is a correction factor to take into account the real robot's kinematics as well as additional disturbing factors. The parameter $\tau > 0$ takes into account communication delays. Similarly, the distance heuristics for collision and obstacle avoidance are computed by

$$\Delta_{i,j}^{coll} = d_{i,j} - 2u_{max}(\kappa + \tau), \tag{30}$$

and

$$\Delta_{i,o}^{obs} = d_{i,o} - u_{max}(\kappa + \tau), \tag{31}$$

respectively.

With the collision and obstacle avoidance constraints presented, we define the system constraints for the CBFs in (9) and (15) as follows:

$$h_{sys}(\mathbf{x}) = \begin{bmatrix} h_{coll}(\mathbf{x}_i, \mathbf{x}_j) = (\Delta_{i,j}^{coll})^2 - d_{min,c}^2 \\ h_{obs}(\mathbf{x}_i, \mathbf{x}_o) = (\Delta_{i,o}^{obs})^2 - d_{min,o}^2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{32}$$

In a decentralized implementation the robots need to compute their components distributedly. The CBFs for collision avoidance and obstacle avoidance are inherently determined locally since the boundary of the safe sets occurs when a robot i is close to $d_{min,c}$ or $d_{min,o}$ with a robot j or obstacle o. Due to the distance-based communication function in (1), a robot j that is close to $d_{min,c}$ with a robot i will be its neighbor, i.e., if $d_{i,j} \approx d_{min,c}$, then $j \in \mathcal{N}_i$. The CBF for connectivity requires each robot i to determine the eigenvalues of the Laplacian as well as the components of the eigenvectors corresponding to robot i and robots $j \in \mathcal{N}_i$. More information on how the eigenvalues and eigenvector components can be computed distributedly can be found in [5, 15, 17, 20, 32, 38, 39].

V. Results

A. Experimental Setup

We demonstrate the proposed CBFs (9), (15)-(16) in hardware experiments where local information is used to compute each robot's control. The robots used in the hardware experiments were GoPiGo 3 differential drive robots designed by Dexter Industries equipped with Raspberry Pi 3 B+ computers and OptiTrack tracking orbs. The robots shared information and were controlled using the Robot Operating System (ROS). The control inputs from the CBFs were handled by the robots as waypoints, where they then used waypoint control to drive to their next desired location, necessitating a nonzero correction factor κ in (29)-(31) that was experimentally chosen to be $\kappa = 0.5$.

An OptiTrack motion capture system tracks the position and orientation of each robot, however, each robot is only given access to its own position and orientation, as well as the position and desired heading of robots within its communication range R at each time-step. The OptiTrack system also provides the robots with the positions of upcoming obstacles. In the following experiments, the communication range was manually set to $R = 3$ meters within a 4×8 meter testbed and we used $\epsilon = 0.1$, $\alpha(h_{res}(\mathbf{x})) = h_{res}(\mathbf{x})$, $\alpha(h_{coll}(\mathbf{x}_i, \mathbf{x}_j)) = 1000h_{coll}^3(\mathbf{x}_i, \mathbf{x}_j)$, and $\alpha(h_{obs}(\mathbf{x}_i, \mathbf{x}_o)) = 1000h_{obs}^3(\mathbf{x}_i, \mathbf{x}_o)$.

B. Experimental Results

1) Case Study - Resilient Flocking: The first set of experiments demonstrate the ability of the proposed controller (9) to provide F-resilience in a flocking application. In these experiments, robots were randomly placed on the left side of the testbed, in a connected, but not necessarily resilient formation. The robots were not required to start in a resilient formation since the controller in (9) can drive non-resilient robot teams to resilient formations. The robots used the controller in (9) to drive to a resilient configuration while avoiding collisions with each other and the obstacles placed in the testbed. The desired input for each robot was given as $u_{des,i} = u_{max} \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \end{bmatrix}^\top$, where θ_i is the desired heading of robot i. The desired headings of the robots were initially evenly distributed between $(-\pi/2, \pi/2)$, and remained as such until the robots used W-MSR to achieve a resilient consensus on a heading, and therefore achieved resilient flocking. The robots used the W-MSR update rule:

$$y_i[t+1] = w_{i,i}[t]y_i[t] + \sum_{j \in \mathcal{R}_i[t]} w_{i,j}[t]y_j[t], \tag{33}$$

where $w_{i,j} > 0$ are weights assigned to each value, $\mathcal{R}_i[t]$ is the set of sorted neighboring values not discarded by robot i, and $\sum_{j \in \mathcal{R}_i[t]} w_{i,j}[t] = 1$. During the experiments, an attack was staged by $F = 1$ randomly chosen adversary. The adversary chooses a desired heading of $\pi/2$ and refused to update its heading with the consensus rule in (33) in order to try to drive the team to its value. The adversary still chose to move in the direction of the team since changing its input could reveal its identity as an adversary. A snapshot from one of the experiments can be seen in Fig. 2.

A total of 20 experiments were run using this setup, and aggregate results for the algebraic connectivity as well as the collision and obstacle avoidance constraints are plotted in Fig. 3. The plotted lines represent the mean of the 20 experiments, while the shaded area represents the standard deviation. In the left plot, the true algebraic connectivity, plotted in green, is based on the ground truth data from OptiTrack, while the algebraic connectivity plotted in blue was computed by the robots using the distance heuristic in (29). While the computed heuristic algebraic connectivity drops below the

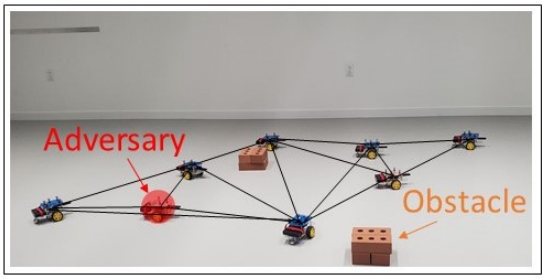

Fig. 2. A snapshot from during one of the resilient flocking experiments. The adversary in this example is highlighted in red, and two obstacles were present in the testbed. The experiments used a communication range of $R = 3$ meters, with safe collision distances of $d_{min,c} = d_{min,o} = 0.3048$ meters.

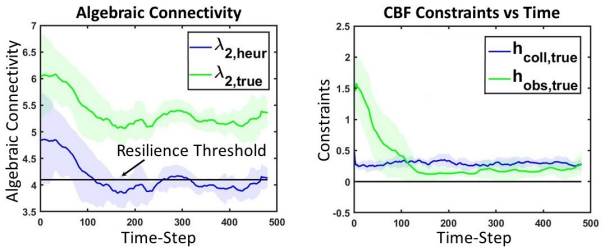

Fig. 3. Aggregate plots of the algebraic connectivity and the collision and obstacle avoidance constraints through 20 resilient flocking experiments. Mean data is shown by lines while the standard deviations are shaded.

$4F + \epsilon$ resilience threshold, the true algebraic connectivity does not. The collision avoidance CBF and obstacle avoidance CBF remain greater than 0 in the right plot, showing that obstacle and inter-robot collisions were successfully avoided. For reference, one sample experiment is shown in Fig. 4. The left plot shows the trajectories of the robots throughout the experiment, with initial positions marked by $\times$'s, final positions marked by triangles, and obstacles marked by circles. The heading consensus using W-MSR is plotted in the right plot. In both plots the adversary is shown with red. Fig. 4 shows that the team is able to reach a resilient heading consensus, despite the adversary, and the team moves together across the testbed as desired.

For comparison, an experiment was also conducted using the CBF proposed in [3] with $h(\mathbf{x}) = \lambda_2(\mathbf{x}) - \epsilon$. The

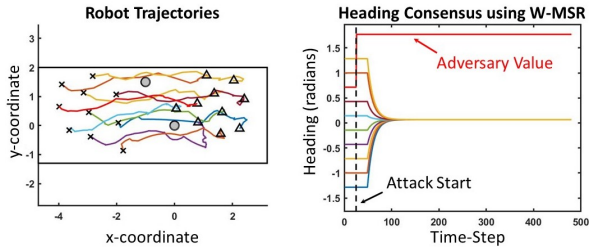

Fig. 4. Trajectory and heading consensus plots from one sample resilient flocking experiment. Initial positions are depicted by $\times$'s on the trajectory plot, while final positions are depicted by triangles, and obstacles are depicted by circles. The robots agree on a heading despite the adversary, and move together across the testbed while avoiding obstacle and inter-robot collisions.

trajectory and heading consensus plots for this experiment are shown in Fig. 5. The CBF proposed in [3] guarantees that the team remains connected, but does not guarantee sufficient connectivity for adversarial resilience. In this experiment, the blue and yellow robots circled in the trajectory plot venture far enough from the rest of the team that their connectivity is insufficient for W-MSR. After discarding outlying information using W-MSR they are only left with their own heading value, and therefore are unable to reach consensus with the rest of the team.

2) Case Study - Large Obstacle: In this experiment, $N = 6$ robots were randomly placed on the left side of the testbed and tasked with moving to the right side of the testbed, while maintaining the sufficient $4F + \epsilon$ algebraic connectivity for F-resilience using the controller in (9). However, an obstacle is placed in the middle of the testbed that is too large for the team to split apart and drive around without violating the F-resilience condition, i.e., the width is greater than $d_{o,max}$. Plots of the robot trajectories and the network algebraic connectivity during this experiment are shown in Fig. 6. The trajectory plot shows that initially the robots were driven to opposite sides of the obstacle. However, a splitting did not occur because it would violate the F-resilience threshold. This caused the robots on the left side of the obstacle to turn around and pass

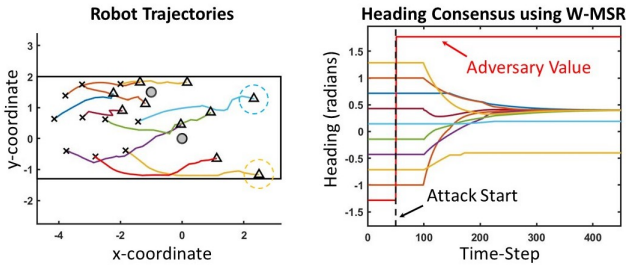

Fig. 5. Trajectory and heading consensus plots from a flocking experiment where the F-resilience controller in (9) is not used. Initial positions are depicted by $\times$'s on the trajectory plot, while final positions are depicted by triangles, and obstacles are depicted by circles. The robots do not agree on one heading due to the insufficient connectivity of the network when W-MSR is run. The circled blue and yellow robots venture far from the team and are unable to achieve consensus with the others.

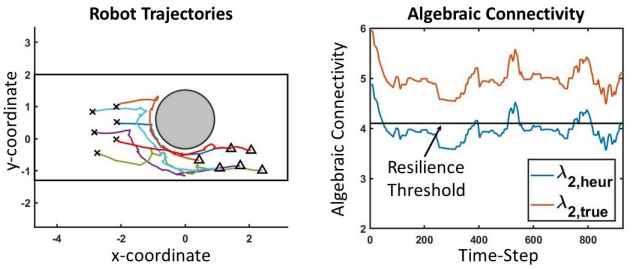

Fig. 6. Trajectory and algebraic connectivity plots from a scenario where the robot team must navigate around a large obstacle while maintaining an algebraic connectivity of $\lambda_2(\mathbf{x}) \geq 4F+\epsilon$. The team initially attempts to split around the obstacle, but eventually maneuvers around the obstacle from one side in order to maintain sufficient connectivity. Initial positions are depicted by $\times$'s, final positions by triangles, and obstacles by circles in the trajectory plot.

the obstacle from the right side with the others in order to maintain sufficient connectivity. This example demonstrates the ability of the proposed controller in (9) to guarantee that sufficient connectivity is maintained.

3) Case Study - Narrow Corridor: As mentioned in Section IV, the addition of collision and obstacle avoidance may limit the ability of the team to navigate in tight environments while maintaining the sufficient connectivity required for F-resilience. Next, we test the ability of the nested CBFs in (15) and (16) to overcome this pitfall. In this experiment, $N = 6$ robots were placed on the left side of the testbed and tasked with moving to the right side of the testbed, while passing through a narrow corridor. The corridor is sufficiently narrow such that two robots cannot pass through it side-by-side, thus forcing the team into a linear formation in order to navigate through it. The gain γ was set to $\gamma_{max} = 3.7$ for this experiment. Fig. 7 shows the robot trajectories as well as the F-resilience, collision, and obstacle avoidance CBF constraints throughout the experiment. The trajectory plot shows that the team was able to successfully navigate through the narrow corridor using the nested CBFs. The right plot shows that the collision and obstacle avoidance constraints were not violated during the experiment. Due to the construction of the nested CBFs, the F-resilience constraint is viewed as a soft constraint, and can be broken if necessary in order to continue satisfying a desired trajectory. This is evident by the yellow line dropping below 0. The team still remains connected ($\lambda_2(\mathbf{x}) > 0$) due to the strict connectivity constraint in (15).

C. Simulations Verifying Lemma 1 and Lemma 4

We run simulations to verify the bounds in Lemma 1 for the size of a large obstacle that allows splitting, $d_{o,max}$, and the gain, γ_{max}, in Lemma 4 that guarantees the robots can pass through environments with obstacles and narrow corridors. The results are shown in Fig. 8. Both simulations used $\epsilon = 0.1$.

In the large obstacle simulation, the communication range was set to $R = 2$ meters, with a safe inter-robot collision distance of $d_{min,c} = 0.1$ meters, $N = 10$ robots, and resilience to $F = 2$ adversaries. This corresponds to a maximum obstacle width of $d_{o,max} = 0.54$ meters. When the obstacle width is set to greater than $d_{o,max}$ the robots need to backtrack in order to satisfy the resilience threshold, and may stall in certain situations as shown by the top left plot. When the obstacle width in simulation is set to $d_{o,max}$ the team is capable of splitting around the obstacle while maintaining resilience.

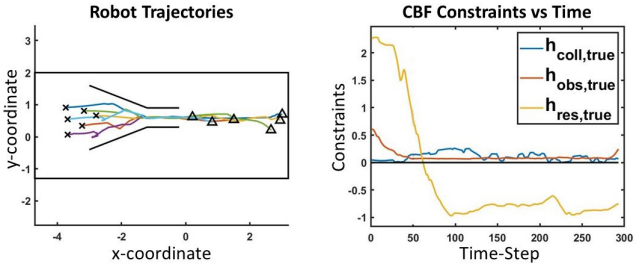

Fig. 7. Trajectory and CBF constraint plots from a scenario where the robot team must navigate through a narrow corridor. The nested CBFs in (15) and (16) allow the F-resilience to be viewed as a soft constraint that can be broken if necessary. Initial positions are depicted by $\times$'s, final positions by triangles, and the corridor by black lines in the trajectory plot.

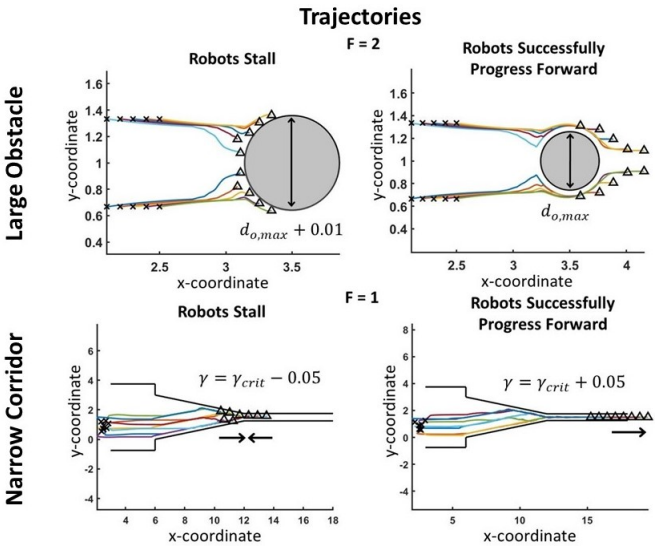

Fig. 8. Trajectory plots for the large obstacle and narrow corridor simulations with $\epsilon = 0.1$. The critical bound for $d_{o,max}$ that determines whether the team splits or stays together was $d_{o,max} = 0.54$ meters. The bottom plots exhibit the existence of a γ_{crit}, where $\gamma < \gamma_{crit}$ would cause the team to stall, but $\gamma > \gamma_{crit}$ would allow the team to progress forward. The robot team formation provided resilience to $F = 2$ adversaries during the large obstacle simulations, and $F = 1$ adversary during the narrow corridor simulations.

In the narrow corridor simulation, the communication range was set to $R = 0.8$ meters, with a safe inter-robot collision distance of $d_{min,c} = 0.6$ meters, and resilience to $F = 1$ adversary. The team consisted of $N = 8$ robots, and the robots have a desired velocity in the direction of the corridor with speed 1 m/s. We study the critical gain value, γ_{crit}, in simulation, where the behavior of the team switches from stalling in the corridor to progressing forward. The critical gain was found to be $\gamma_{crit} = 1.3$. The upper bound was computed to be $\gamma_{max} = 2.6$ by (21). We note that the theoretical upper bound of 2.6 is higher because it is conservative and considers the worst possible configuration, and also does not consider the effects of collision avoidance in the external CBF in (15). Robots to the far right in the line formation will be 'pushed' to the right in order to maintain a safe distance from robots to their left, further helping the team pass through the corridor even when the desired input from the nested CBF, $\tilde{\mathbf{u}}_{des}$ in (16), may make them stall. The critical gain found in simulation was determined by trying different gains and observing the behavior of the team, to see when they switched from stalling in the corridor to passing through it.

VI. Conclusion

In this paper we construct a CBF that can maintain sufficiently high network connectivity for F-resilience while considering additional control objectives. Previous approaches

use linear combinations of various control objectives, and are therefore sensitive to changes or additions to the objectives. Conversely, our CBF approach is trivially adaptable to different control objectives while maintaining guarantees on the satisfaction of each desired control constraint. The adaptability of our approach enables the addition of practical constraints such as collision and obstacle avoidance that are necessary for implementing the controller on real multi-robot systems in cluttered environments. We demonstrate the ability of our approach to simultaneously provide F-resilience to adversaries, collision avoidance, and obstacle avoidance, on distributed robot platforms while achieving a flocking objective. We also test our approach in a case study with large obstacles that try to split apart the team. Finally, we propose an alternative nested CBF that takes the F-resilience condition as a soft constraint, and we test its effectiveness with an experiment where robots need to navigate through a narrow corridor that forces the team to break formation.

While the CBFs and results presented in this paper focus on providing F-resilience to adversaries, it is worth mentioning that they can be applied to fault resilience as well. If a robot drops communication with the rest of the network, whether it is an adversary purposely cutting off connection, or a robot experiencing a communication failure, similar CBFs can be applied to safeguard from such scenarios. In these scenarios, the network must be sufficiently connected in order to remain connected after the removal of any F robots from the network. This corresponds to a network with a vertex connectivity of $F + 1$.

Acknowledgments

The authors gratefully acknowledge partial support through the Office of Naval Research (ONR) Young Investigator Program (YIP) [grant number: N00014-21-1-2714].

References

[1] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European Control Conference (ECC)*, pages 3420–3431. IEEE, 2019.

[2] Gianluca Antonelli, Filippo Arrichiello, and Stefano Chiaverini. Flocking for multi-robot systems via the null-space-based behavioral control. *Swarm Intelligence*, 4 (1):37, 2010.

[3] Beatrice Capelli and Lorenzo Sabattini. Connectivity maintenance: Global and optimized approach through control barrier functions. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5590–5596. IEEE, 2020.

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

[5] Maria Carmela De Gennaro and Ali Jadbabaie. Decentralized control of connectivity for multi-agent systems. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 3628–3633. IEEE, 2006.

[6] Magnus Egerstedt, Jonathan N Pauli, Gennaro Notomista, and Seth Hutchinson. Robot ecology: Constraint-based control design for long duration autonomy. *Annual Reviews in Control*, 46:1–7, 2018.

[7] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[8] Anurag Ganguli, Jorge Cortés, and Francesco Bullo. Multirobot rendezvous with visibility sensors in nonconvex environments. *IEEE Transactions on Robotics*, 25 (2):340–352, 2009.

[9] Cinara Ghedini, Carlos Ribeiro, and Lorenzo Sabattini. Toward fault-tolerant multi-robot networks. *Networks*, 70 (4):388–400, 2017.

[10] Stephanie Gil, Cenk Baykal, and Daniela Rus. Resilient multi-agent consensus using wi-fi signals. *IEEE control systems letters*, 3(1):126–131, 2018.

[11] Paul Glotfelter, Jorge Cortés, and Magnus Egerstedt. Nonsmooth barrier functions with applications to multi-robot systems. *IEEE control systems letters*, 1(2):310–315, 2017.

[12] Luis Guerrero-Bonilla and Vijay Kumar. Realization of r-Robust Formations in the Plane Using Control Barrier Functions. *IEEE Control Systems Letters*, 4(2):343–348, 2019.

[13] Shao-Chen Hsu, Xiangru Xu, and Aaron D Ames. Control barrier function based quadratic programs with application to bipedal robotic walking. In *2015 American Control Conference (ACC)*, pages 4542–4548. IEEE, 2015.

[14] Yiwen Huang, Sze Zheng Yong, and Yan Chen. Guaranteed vehicle safety control using control-dependent barrier functions. In *2019 American Control Conference (ACC)*, pages 983–988. IEEE, 2019.

[15] David Kempe and Frank McSherry. A decentralized algorithm for spectral analysis. *Journal of Computer and System Sciences*, 74(1):70–83, 2008.

[16] Mohammad Khajenejad, Matthew Cavorsi, Ruochen Niu, Qiang Shen, and Sze Zheng Yong. Tractable Compositions of Discrete-Time Control Barrier Functions with Application to Driving Safety Control. In *2021 European Control Conference (ECC)*, pages 1303–1309. IEEE, 2021.

[17] A. Y. Kibangou and C. Commault. Decentralized Laplacian Eigenvalues Estimation and Collaborative Network Topology Identification. *3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys'12)*, 2012.

[18] Heath J LeBlanc, Haotian Zhang, Xenofon Koutsoukos, and Shreyas Sundaram. Resilient asymptotic consensus

in robust networks. *IEEE Journal on Selected Areas in Communications*, 31(4):766–781, 2013.

[19] Manao Machida and Masumi Ichien. Consensus-based control barrier function for swarm. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8623–8628. IEEE, 2021.

[20] Frederik Mallmann-Trenn, Matthew Cavorsi, and Stephanie Gil. Crowd Vetting: Rejecting Adversaries via Collaboration With Application to Multirobot Flocking. *IEEE Transactions on Robotics*, pages 1–20, 2021. doi: 10.1109/TRO.2021.3089033.

[21] Marco Minelli, Jacopo Panerati, Marcel Kaufmann, Cinara Ghedini, Giovanni Beltrame, and Lorenzo Sabattini. Self-optimization of resilient topologies for fallible multi-robots. *Robotics and Autonomous Systems*, 124: 103384, 2020.

[22] Gennaro Notomista and Magnus Egerstedt. Constraint-Driven Coordinated Control of Multi-Robot Systems. In *2019 American Control Conference (ACC)*, pages 1990–1996, 2019. doi: 10.23919/ACC.2019.8814594.

[23] Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on automatic control*, 51(3):401–420, 2006.

[24] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[25] Pio Ong, Beatrice Capelli, Lorenzo Sabattini, and Jorge Cortés. Network connectivity maintenance via nonsmooth control barrier functions. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 4786–4791. IEEE, 2021.

[26] Pio Ong, Beatrice Capelli, Lorenzo Sabattini, and Jorge Cortes. Nonsmooth Control Barrier Function Design of Continuous Constraints for Network Connectivity Maintenance. *arXiv preprint arXiv:2112.05935*, 2021.

[27] Fabio Pasqualetti, Antonio Bicchi, and Francesco Bullo. Consensus computation in unreliable networks: A system theoretic approach. *IEEE Transactions on Automatic Control*, 57(1):90–104, 2011.

[28] Wei Ren, Randal W Beard, and Ella M Atkins. Information consensus in multivehicle cooperative control. *IEEE Control systems magazine*, 27(2):71–82, 2007.

[29] K. Saulnier, D. Saldaña, A. Prorok, G. J. Pappas, and V. Kumar. Resilient Flocking for Mobile Robot Teams. *IEEE Robotics and Automation Letters*, 2(2):1039–1046, April 2017.

[30] Daniel A. Spielman. *Spectral and Algebraic Graph Theory*. Yale University, 2019.

[31] Housheng Su, Xiaofan Wang, and Zongli Lin. Flocking of multi-agents with a virtual leader. *IEEE transactions on automatic control*, 54(2):293–307, 2009.

[32] S. Sundaram and C. N. Hadjicostis. Distributed function calculation and consensus using linear iterative strategies. *IEEE Journal on Selected Areas in Communications*, 26 (4):650–660, 2008.

[33] S. Sundaram and C. N. Hadjicostis. Distributed Function Calculation via Linear Iterative Strategies in the Presence of Malicious Agents. *IEEE Transactions on Automatic Control*, 56(7):1495–1508, 2011.

[34] Xiao Tan and Dimos V Dimarogonas. Distributed Implementation of Control Barrier Functions for Multi-agent Systems. *IEEE Control Systems Letters*, 6:1879–1884, 2021.

[35] James Usevitch and Dimitra Panagou. Adversarial Resilience for Sampled-Data Systems using Control Barrier Function Methods.

[36] Li Wang, Aaron Ames, and Magnus Egerstedt. Safety barrier certificates for heterogeneous multi-robot systems. In *2016 American Control Conference (ACC)*, pages 5213–5218. IEEE, 2016.

[37] Li Wang, Aaron D Ames, and Magnus Egerstedt. Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, 33(3):661–674, 2017.

[38] Peng Yang, Randy A Freeman, Geoffrey J Gordon, Kevin M Lynch, Siddhartha S Srinivasa, and Rahul Sukthankar. Decentralized estimation and control of graph connectivity for mobile sensor networks. *Automatica*, 46 (2):390–396, 2010.

[39] Mehran Zareh, Lorenzo Sabattini, and Cristian Secchi. Distributed laplacian eigenvalue and eigenvector estimation in multi-robot systems. *Distributed autonomous robotic systems*, pages 191–204, 2018.

[40] Haotian Zhang and Shreyas Sundaram. Robustness of information diffusion algorithms to locally bounded adversaries. In *2012 American Control Conference (ACC)*, pages 5855–5861. IEEE, 2012.