Resilient Flocking for Mobile Robot Teams

Kelsey Saulnier, David Saldaña, Amanda Prorok, George J. Pappas, and Vijay Kumar

***Abstract*—We present a method that enables resilient formation control for mobile robot teams in the presence of noncooperative (defective or malicious) robots. Recent results in network science define graph topological properties that guarantee resilience against faults and attacks on individual nodes in static networks. We build on these results to propose a control policy that allows a team of mobile robots to achieve resilient consensus on the direction of motion. Our strategy relies on dynamic connectivity management that makes use of a metric that characterizes the robustness of the communication network topology. Our method distinguishes itself from prior work in that our connectivity management strategy ensures that the network lies above a critical *resilience threshold*, guaranteeing that the consensus algorithm always converges to a value within the range of the cooperative agents' initial values. We demonstrate the use of our framework for resilient flocking, and show simulation results with groups of holonomic mobile robots.**

***Index Terms*—Distributed robot systems, networked robots.**

I. Introduction

CONSENSUS algorithms allow multiple robots to achieve agreement on estimates of variables in a distributed manner. This allows robots to coordinate as a cohesive team, enabling applications such as formation control [1]–[3]. The problem with distributed consensus, however, is that it assumes that all robots are cooperative. A single non-cooperative (defective or malicious) robot can potentially manipulate the whole network and prevent the team of cooperative robots from achieving their goal. As a consequence, the systems are susceptible to failure when one or several robots are non-cooperative and share wrong information. This situation can be due to malicious attacks (e.g., a malicious outsider trying to manipulate the whole network) or due to platform-level faults (e.g., a robot sharing an incorrect location due to a defective GPS sensor). As a consequence, the *resilience* of the communication network is of utmost importance [4]. We note that we distinguish between *robustness* — the ability to cope with errors that can be modeled — and *resilience* — the ability to adapt to tasks in the face of unknown attacks or failures that cannot be modeled.

In this work, we focus on the problem of resilient coordinated motion control of a team of mobile robots. Our coordination strategy builds on the distributed consensus algorithm. In order to provide resilience, we use an alternate version of the linear consensus protocol, termed Weighted-Mean Subsequence-Reduced (W-MSR) algorithm [5], which guarantees that the consensus is achieved and that the consensus value lies within the range of the cooperative robots' initial values if certain network topological criteria are satisfied. While the authors of the former work focus on static networks, in this work we consider time-varying networks formed by mobile robot teams. The key element of our approach is a control policy that maintains the connectivity of a mobile robot team above a given critical threshold, thus ensuring that the necessary topological requirements are met at all times. Since tests for network resilience are known to be co-NP complete [6], we instead resort to the computation of the algebraic connectivity of the network, which we use to construct a lower-bound measure. We then proceed to show how our control law achieves resilient coordinated motion control. As a specific application, we consider the flocking of a swarm of robots. Individual robots follow control policies that guarantee resilience, and use the W-MSR (resilient) consensus algorithm to ensure the correct behavior even in the presence of non-cooperative robots. Though we demonstrate the utility of our framework on flocking, the same concepts may be applied to more concrete applications such as vehicle platooning, a problem that has gained much attention with the onset of intelligent vehicle systems [7].

A. Background

The topic of robustness has received considerable attention, particularly in the domain of complex networks [8], [9]. A main result of this body of work states that resilience can be achieved through sufficiently high connectivity: if the connectivity of the network is $2F$ or less, a subset of F or more malicious or otherwise misbehaving nodes can prevent some of the correctly functioning nodes from receiving legitimate information from other nodes in the network. Conversely, when the network connectivity is $2F + 1$ or higher, there are various algorithms that enable a reliable diffusion of information [10], [11]. This value F defines the maximum number of non-cooperative agents that can be supported by the network. As long the actual number of non-cooperative nodes remains below F the performance of the algorithms is guaranteed. Unfortunately, these algorithms not only depend on high connectivity, but also require non-local information in order to compute updates. As a consequence, Zhang *et al.* and later LeBlanc *et al.* [5], [12] introduced an alternative

Manuscript received September 10, 2016; accepted December 30, 2016. Date of publication January 18, 2017; date of current version February 16, 2017. This paper was recommended for publication by Associate Editor H.-L. Choi and Editor N. Y. Chong upon evaluation of the reviewers' comments. This work was supported in part by the ONR under Grants N00014-15-1-2115 and N00014-14-1-0510, in part by the ARL under Grant W911NF-08-2-0004, in part by the NSF under Grants IIS-1426840 and CNS-1521617, in part by the ARO under Grant W911NF-13-1-0350, and in part by the United Technologies, and TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

The authors are with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: saulnier@seas.upenn.edu; dajulian@gmail.com; prorok@seas.upenn.edu; pappasg@seas.upenn.edu; kumar@cis.upenn.edu).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LRA.2017.2655142

definition of network resilience, termed r-robustness, and the W-MSR algorithm which together provide resilient asymptotic consensus using purely local update rules. In this context, the identities and actual number of non-cooperative nodes remain unknown. In order to build the required topologies, state-of-the art attachment algorithms assume that any node can be connected to any other node in the network [12] (i.e., in absence of sophisticated node placement algorithms, the networks must provide full connectivity). This problem is compounded by dynamic networks and moving nodes. Also, it is worth noting that the prior approach mainly deals with the problem of distributed estimation, and it remains to be explored how it applies to problems that require robust control of shapes and distributions, such as for cooperative exploration and coordination tasks [13], [14].

B. Related Work

The problem of resilience has also been considered in mobile robot systems literature [15]–[17], with a particular focus on rendez-vous (i.e., the application of consensus algorithms to induce a gathering of robots in d-dimensional space). Similar to our work, [16] considers the presence of non-cooperative robots and develops a solution that controls a team of robots so that the goal is achieved without the influence of the non-cooperative robots. The proposed approach exploits concepts from combinatorial geometry to obtain intersecting d-dimensional convex hulls, consequently providing 'safe' rendez-vous points. Since the complexity of the method is exponential in d, the authors propose an approximate variant that scales linearly with the number of robots. Although this work provides an effective solution, it is not easily adapted to more general settings.

In this work, we are interested in achieving resilient flocking behavior. Flocking is a mechanism for achieving velocity synchronization and regulation of relative distances within a group of mobile robots [18]. A number of works derive decentralized controllers for achieving the flocking phenomenon based on the distributed consensus algorithm (see [19] and the references therein). These results critically rely on the assumption that the underlying communication network is either connected throughout time [20], [21], or is jointly connected over infinite sequences of bounded time intervals [1]. The idea of controlling the connectivity of a network of mobile robots is not new. Indeed, Zavlanos *et al.* [22] propose a control law that ensures flocking is achieved, while preserving connectivity. Also, Gennaro *et al.* [23] and Stump *et al.* [24] have proposed controllers that allow robots to move in the direction of increasing algebraic connectivity. Although these methods address the problem of communication quality, they do not address the problem of network resilience and non-cooperating robots. Hsieh *et al.* [25] address the problem of decentralized shape generation while simultaneously maintaining connectivity. Similarly, the authors address the problem of maintaining sufficient communication quality during motion; however, the method is not able to accommodate non-cooperative robots. Our method distinguishes itself from these approaches in that our connectivity management strategy ensures that the network lies above a critical *resilience threshold*. In particular, we differentiate between states where it is safe to compute consensus updates, and states where it is not. Overall, our switching controller guarantees that our consensus algorithm always converges to a value within the range of the cooperative agents' initial values.

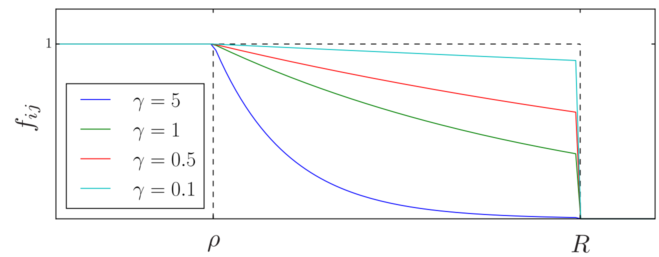

Fig. 1. The effect of design parameter γ on communication function f_{ij}. The x-axis is the distance between agents, $\|\mathrm{x}_i - \mathrm{x}_j\|$.

II. Problem Statement

Consider a robotic network modeled as a undirected graph $\mathcal{G} = (\mathcal{V}, \varepsilon)$, where the vertices $\mathcal{V} = \{1, 2, ..., n\}$ represent the robots, and the edges $\varepsilon \subseteq \mathcal{V} \times \mathcal{V}$ represent the communication links. The location of the robots in the d-dimensional Euclidean space, $\mathbb{R}^d$, is denoted by $\mathcal{X} = \{\mathrm{x}_1, ..., \mathrm{x}_n\}$, $\mathrm{x}_i \in \mathbb{R}^d$. Based on the communication disk model, a function $f : \mathbb{R}^d \times \mathbb{R}^d \to [0, 1]$ is used to model the quality of the link between every pair of nodes (i, j) based on their locations. In the rest of this paper, we use the following exponential-based function [24],

$$f_{ij} = \begin{cases} 1 & \|\mathrm{x}_i - \mathrm{x}_j\| < \rho \\ 0 & \|\mathrm{x}_i - \mathrm{x}_j\| \geq R \\ \exp\left(\frac{-\gamma(\|\mathrm{x}_i - \mathrm{x}_j\| - \rho)}{R - \rho}\right) & \text{otherwise.} \end{cases}, \quad (1)$$

where $\rho > 0$ is a threshold up to which the communication quality is considered optimal, and $R > \rho$ is the maximum communication radius. The design parameter γ trades off the smoothness of f_{ij} and the magnitude of its derivative. This effect can be seen in Fig. 1 where the function f_{ij} is plotted for various γ values. Alternative communication functions are proposed in [26].

We assume that the robots are holonomic and move in the environment based on the control input

$$\dot{\mathrm{x}}_i = \mathbf{u}_i.$$

This work addresses the problem of ensuring resilient coordinated motion control in the presence of internal threats, i.e., non-cooperative robots that can influence the coordination (and motion) of the whole team. In particular, we address the problem of generating a control law that guarantees resilient flocking. Our approach to flocking is based on the distributed consensus algorithm [19]. When performing consensus, each robot i aims at estimating a global variable of interest y_i (in the case of flocking, we consider the direction of motion as the variable of interest). This goal is achieved through local interaction, where each robot i updates its own value at time-step t based on a consensus update rule:

$$y_i[t+1] = g(y_i[t], \{y_j[t] | j \in \mathcal{N}_i\}), \quad (2)$$

where $\mathcal{N}_i$ is the set of neighbors of robot i.

In [1], the authors show that, given a connected, undirected graph $\mathcal{G}$, every node $i \in \mathcal{V}$ reaches consensus on the average of the initial values, $y_i[t] \to \bar{y}[0] = \frac{1}{n}\sum_{i \in \mathcal{V}} y_i[0]$, when $t \to \infty$ by exchanging messages with the local neighborhood and applying an averaging function $y_i[t+1] = \frac{1}{|\mathcal{N}_i|+1}(y_i[t] + \sum_{j \in \mathcal{N}_i} y_j[t])$. Clearly, this strategy only works when all nodes in the network cooperate by executing the update function reliably and communicating truthful values. This insight leads us to the definition of the following threat model.

Definition 1 (Non-Cooperative Robot): A robot is *cooperative* if it applies the consensus update rule (Eq. (2)) at every time-step t and shares the result with its neighbors. It is called *non-cooperative* otherwise.

Definition 2 (Resilient consensus): A group of mobile robots is said to reach *resilient consensus* if the cooperative robots achieve consensus to a value that lies between the maximum and minimum initial values of the cooperative robots, even in the presence of up to F non-cooperative robots.

Non-cooperative robots can be either *(i)* defective (unintentionally non-cooperative, e.g., due to a faulty sensor or actuator), or *(ii)* malicious (intentionally non-cooperative, e.g., an external attacker gains access to a node's communication module, with the goal of manipulating the system). We note that our threat model considers non-cooperation in *communication* only (i.e., non-cooperative robots will execute the agreed upon motion commands, even though they do not necessarily communicate truthful values). Despite this assumption, our threat model is sufficiently powerful, since devious motion can be easily detected and ignored with prior methods [27], whereas devious communication can be executed in such a way that it is hard to detect and ignore. Importantly, our method does not require that non-cooperative nodes send the same (incorrect) value to all their neighbors. Thus, our results also apply to the Byzantine model of adversaries (cf. results in [5]).

We consider a team of mobile robots that coordinate their direction of motion by applying the consensus protocol; in other words, they achieve global agreement upon their direction of motion. The challenge, then, consists of ensuring that the consensus is safely achieved in the presence of non-cooperative robots.

Problem 1 (Resilient Flocking): Given a networked system of N mobile robots, design a control law that guarantees resilient consensus on the direction of motion.

In Section III, we introduce the fundamental concepts of network resilience which underpin our methodology. In Section IV, we develop a control policy that manages the connectivity of the mobile robot team in order to meet the necessary topological robustness requirements. In Section V, we develop a solution to the above-mentioned problem. Finally, in Section VI we present simulations illustrating our approach.

III. Conditions for Resilient Consensus

A *resilient communication network* is defined as a network that is guaranteed to reach resilient consensus. The identity and the strategy of the non-cooperative nodes are assumed to remain unknown. Recent work in the domain of network science introduces the W-MSR algorithm [5], [12], which is a method that achieves consensus to a weighted average of cooperative nodes' values. Yet, for convergence to be assured the network must satisfy certain topological conditions, which we detail below. If these conditions are not satisfied the network may fail to converge even in the presence of fewer than F non-cooperative nodes.

The W-MSR algorithm consists of three steps, executed at each time step t by each node. First, node i creates a sorted list, from smallest to largest, with the values $y_j[t]$ received from its neighbors $j \in \mathcal{N}_i$. Second, node i compares the list to its own value $y_i[t]$. If there are F or more values that are larger than $y_i[t]$, the F largest values are removed. If there are fewer than F larger values then all larger values are removed. The same removal process is applied to the smaller values. After this process is completed, the neighbors whose values remain in the list are denoted by $\mathcal{R}_i[t]$. Finally, node i updates its value with the following rule:

$$y_i[t+1] = w_{ii}[t]y_i[t] + \sum_{j \in \mathcal{R}_i[t]} w_{ij}[t]y_j[t], \tag{3}$$

where $w_{ij} > 0$, and $\sum_j w_{ij}[t] = 1$. In the remainder of this paper, we consider all weights $w_{ij} = 1/(|\mathcal{R}_i[t]| + 1)$. An extended explanation of this algorithm is given in [12]. Using this algorithm, resilient asymptotic consensus is assured as long as the communication graph $\mathcal{G}$ is $(2F+1)$-*robust*, a topological requirement we outline in the following two definitions.

Definition 3 (r-reachable): A nonempty vertex set $\mathcal{A} \in \mathcal{V}$ is said to be r-reachable if $\exists v_i \in \mathcal{A}$ such that $|\delta\mathcal{A}_{v_i}| \geq r, r \in \mathbb{Z}_{\geq 0}$, where for vertex $v_i \in \mathcal{A}$,

$$\delta\mathcal{A}_{v_i} = \{(v_i, v_j) \in \varepsilon \ : \ v_j \in \mathcal{V} \setminus \mathcal{A}\}. \tag{4}$$

is the number of edges leaving subgraph $\mathcal{A}$ from v_i.

Definition 4 (r-robust): A non-trivial graph $\mathcal{G}$ is said to be r-robust if for each pair of disjoint sets $\mathcal{A}_1, \mathcal{A}_2 \subset \mathcal{V}$ at least one is r-reachable.

Based on these definitions, Zhang and Sundaram [5] obtained the following property of r-robust graphs:

Theorem 1 (Th. 1, [5]): Consider a network modeled by a graph $\mathcal{G} = (\mathcal{V}, \varepsilon)$ where each cooperative node updates its value based on the W-MSR algorithm with parameter F. Then, resilient asymptotic consensus is guaranteed if the graph $\mathcal{G}$ is $(2F+1)$-robust.

IV. Maintaining a Resilient Network

Resilient consensus is guaranteed by the W-MSR algorithm in networks that are $(2F+1)$-robust. However, for a given network, computing the level of r-robustness is co-NP complete [6]. For the static networks considered in prior works, the level of r-robustness of a network must only be computed once. In the case of small networks this computational burden may be acceptable. However, in the case of large networks or time-varying networks where the level of r-robustness must be repeatedly computed, the problem becomes intractable. For this reason, we resort to the computation of an alternate metric that lower-bounds the value r. Subsequently, we use this metric to control the resilience of the network.

A. A Lower-Bound on r-Robustness

In the following, we derive a lower bound measure of r-robustness based on the Cheeger constant, also called *isoperimetric constant* which for a graph $\mathcal{G}$ is defined as

$$h(\mathcal{G}) = \min\left\{\frac{|\delta\mathcal{A}|}{|\mathcal{A}|} \ : \ \mathcal{A} \subseteq \mathcal{V}, 0 < |\mathcal{A}| \leq \frac{1}{2}|\mathcal{V}|\right\},$$

where $\delta\mathcal{A}$ is defined as

$$\delta\mathcal{A} = \{(v_i, v_j) \in \varepsilon : v_i \in \mathcal{A}, v_j \in \mathcal{V} \setminus \mathcal{A}\}.$$

Lemma 1 (Lemma 1, [28]): For a given graph $\mathcal{G}$, the Cheeger constant, h($\mathcal{G}$), of the graph lower bounds the value r for which the graph is r-robust.

This lower bound is not very useful however, since finding $h(\mathcal{G})$ for a given graph is NP-Hard [29]. Instead, we resort to another lower bound which is easy to compute.

Theorem 2: For a given graph $\mathcal{G} = (\mathcal{V}, \varepsilon)$, $\lceil \frac{\lambda_2}{2} \rceil$, where λ_2 is the algebraic connectivity of $\mathcal{G}$, lower bounds the r for which the graph is r-robust.

Proof: From Lemma 1, we know that $r \geq h(\mathcal{G})$. The Cheeger constant is itself lower-bounded by $\lambda_2/2$ [30], where λ_2 is the second smallest eigenvalue of the Laplacian matrix of graph $\mathcal{G}$. Thus,

$$r \geq h(\mathcal{G}) \geq \frac{\lambda_2}{2},$$

and since r may take only integer values

$$r \geq \left\lceil \frac{\lambda_2}{2} \right\rceil.$$

■

From this conclusion, and the conditions of convergence of W-MSR, resilient convergence can be guaranteed as long as

$$\lambda_2 > 4F, \tag{5}$$

since for any $\lambda_2 = 4F + \epsilon$, with $\epsilon > 0$, $\lceil \frac{\lambda_2}{2} \rceil = \lceil 2F + \frac{\epsilon}{2} \rceil \geq 2F + 1$. For the remainder of this work we will refer to this threshold as the *resilience threshold*.

Finally, we note that for a given F, the minimum number of nodes required for resilient consensus is $4F + 1$ [31]. Likewise, the minimum number of nodes required by a connectivity controller based on Eq. (5) is $4F + 1$ (since $N \geq \lambda_2$). Therefore using the lower bound measure to guarantee resilience does not require an increase in the number of robots, but may require more connections between them.

Remark 1: The bound $r \geq \lceil \frac{\lambda_2}{2} \rceil$ is tight. This can be seen in the case of the complete graph on N nodes. Here, $\lambda_2 = N$ and $r = \lceil N/2 \rceil$ and, thus, $r = \lceil \frac{\lambda_2}{2} \rceil$.

B. Control of Algebraic Connectivity

To create a controller for the algebraic connectivity, λ_2, we first note that λ_2 is a concave function of the graph Laplacian L. Although the function $\lambda_2(L)$ is non-smooth, its derivative where λ_2 is unique (i.e. $\lambda_2 \neq \lambda_3$) can be shown to be [24]

$$\frac{\partial \lambda_2(L)}{\partial L} = \frac{\mathbf{v}_2 \mathbf{v}_2^T}{\mathbf{v}_2^T \mathbf{v}_2}.$$

It is shown in [23] that even when λ_2 is not unique this derivative serves as a supergradient and can be used for the purpose of gradient climbing. Using a weighted graph Laplacian

$$[L]_{ij} = \begin{cases} -f_{ij} & i \neq j \\ \sum_j f_{ij} & i = j \end{cases},$$

where the function f_{ij} is defined as in Eq. (1), allows us to take the derivative $\frac{\partial L}{\partial x_\alpha}$ for each dimension, α, of the position. Linearizing and using the chain rule we arrive at

$$\frac{\partial \lambda_2(L(\mathbf{x}))}{\partial L(\mathrm{x})} \frac{\partial L(\mathrm{x})}{\partial x_{i,\alpha}} = \text{Trace}\left\{ \left[\frac{\mathbf{v}_2 \mathbf{v}_2^T}{\mathbf{v}_2^T \mathbf{v}_2} \right]^T \left[\frac{\partial L(\mathrm{x})}{\partial x_{i,\alpha}} \right] \right\}. \tag{6}$$

The expression for $\frac{\partial L(\mathrm{x})}{\partial x_{i,\alpha}}$ can be easily computed from Eq.(1). This gradient indicates to each robot the directional derivative which will increase the algebraic connectivity of the graph.

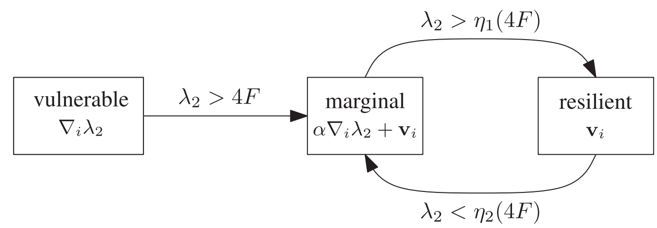

Fig. 2. Proposed three-state hybrid controller for resilient flocking. The switching signal is based on the algebraic connectivity of the network.

V. Resilient Flocking

As summarized in Section I-B, none of the existing approaches address the question of *resilient* flocking, in the presence of non-cooperative communication from anonymous robot team members. Similar to the approach in [22], we propose a controller that maintains a critical level of connectivity, hereby ensuring a resilient network topology, and allowing us to apply the W-MSR algorithm to compute consensus updates that lead to resilient asymptotic consensus.

A. Robot Controller

For the purpose of resilient flocking we propose a three-state controller similar to the one proposed in [32]. In our proposed controller, robots switch from one state to another as a function of the current algebraic connectivity of the network. A diagram of the controller can be seen in Fig. 2. The three states are termed the *vulnerable state*, the *marginal state*, and the *resilient state*, and are elaborated below.

In the vulnerable state, the system is not guaranteed to achieve resilient consensus. In this state, the robots apply a control law which increases the algebraic connectivity of the system

$$\mathbf{u}_i = \nabla_i \lambda_2, \tag{7}$$

where $\nabla \lambda_2 \in \mathbb{R}^{dN}$, determined from Eq. (6), and $\nabla_i \lambda_2$ is a d-dimensional gradient that steers robot i to a resilient formation. Once λ_2 reaches the resilience threshold, the system switches over to the marginal state.

In the marginal state, the system is guaranteed to be $2F + 1$-robust and can safely run consensus dynamics. The system is still close to the resilience threshold, so in this state we consider a control law which includes the team's objective while also ensuring the system remains connected

$$\mathbf{u}_i = \beta \nabla_i \lambda_2 + \mathbf{v}_i, \tag{8}$$

where $\mathbf{v}_i$ represents a navigation function indicating a goal direction for each robot, i, and β is chosen such that $\frac{\mathrm{d}\lambda_2}{\mathrm{d}t} > 0$ at each time step. In the case of flocking, $\mathbf{v}_i = \mathbf{y}_i$ is agreed upon by the robots via the resilient consensus dynamics shown in Eq. (3). Since our control law guarantees an r-robust network topology, each robot i can compute $\mathcal{R}_i$ locally and apply the resilient consensus update safely.

When the system is far enough beyond the resilience threshold as specified by a parameter $\eta_1 > 1$ so that $\lambda_2 > \eta_1(4F)$, the system switches to the resilient state where the controller

$$\mathbf{u}_i = \mathbf{v}_i \tag{9}$$

is used. The robots continue to run the consensus dynamics, Eq. (3), since the system is guaranteed to be $2F + 1$-robust.

The system returns to the marginal state from the resilient state if $\lambda_2 < \eta_2(4F)$ for a parameter $1 < \eta_2 < \eta_1$ which provides some hysteresis for the switching between the marginal and resilient states.

B. Convergence Properties

To show convergence, we initially show that using the controller in Eq. (7) the system goes from the vulnerable state to the marginal state. From there, it must be shown that the system never reenters the vulnerable state. Once the robots are in the marginal state, it is guaranteed that they will agree on a direction of motion $\bar{\mathrm{v}}$ through the consensus dynamics even in the presence of up to F non-cooperative robots. In finite time, switching between the marginal and resilient states will end and the robots will remain in the resilient state.

Proposition 1: If the robots are in the vulnerable state, the controller in Eq. (7) will drive the system to the marginal state.

Proof: As long as the value of λ_2 is below the resilience threshold, we apply the control in Eq. (7), hence the marginal state will be reached in finite time. ■

Proposition 2: If the robots are in the marginal state, following the controller in Eq. (8), they will never reenter the vulnerable state. Furthermore, if there are no more than F non-cooperative robots, the robots' orientation will asymptotically converge to the same value.

Proof: The condition to reenter the vulnerable state is that λ_2 drops below the resilience threshold. Therefore, it suffices to show that

$$\frac{\mathrm{d}\lambda_2}{\mathrm{d}t} \geq 0, \quad \forall t$$

in the marginal state. The derivative can be rewritten as

$$\frac{\mathrm{d}\lambda_2}{\mathrm{d}t} = \sum_{i=1}^{N} \frac{\partial \lambda_2}{\partial \mathbf{x}_i} \frac{\partial \mathbf{x}_i}{\partial t} = \sum_{i=1}^{N} \frac{\partial \lambda_2}{\partial \mathbf{x}_i} \mathbf{u}_i. \tag{10}$$

using the chain rule. Therefore, as long as

$$\frac{\partial \lambda_2}{\partial \mathbf{x}_i} \mathbf{u}_i \geq 0, \quad \forall i, \ \forall t, \tag{11}$$

the system will never leave the marginal state to the vulnerable state. The value β is chosen to ensure that this is the case. Using the controller from Eq. (8) and solving for β in Eq. (11) we arrive at the the following condition:

$$\beta \geq \frac{-\nabla_i \lambda_2^T \mathbf{v}_i}{\|\nabla_i \lambda_2\|^2}.$$

Using Eq. (6), the value of β can be computed at each time step by each robot, thus ensuring that the system does not return to the vulnerable state for any time step t. If β is chosen such that $\frac{\mathrm{d}\lambda_2}{\mathrm{d}t} > 0$ the system will eventually reach the resilient state.

The system switches out of the vulnerable state when $\lambda_2 > 4F$ which guarantees $2F + 1$-robustness. Thus, in the marginal and resilient states, each robot i will converge to a common navigation vector, $\mathbf{v}_i \rightarrow \bar{\mathbf{v}}$ as long are there are no more than F non-cooperative robots. ■

Proposition 3: Once in the marginal state, the system will switch between marginal and resilient states. If there are no more than F non-cooperative robots, switching will stop with the system in the resilient state after a finite time interval.

Proof: By design, once the system is in the marginal state it will move to the resilient state since we choose β such that $\frac{\mathrm{d}\lambda_2}{\mathrm{d}t} > 0$. To show that the system will remain in the resilient state, we show that the derivative $\frac{\mathrm{d}\lambda_2}{\mathrm{d}t}$ converges exponentially to zero in the resilient state as consensus on the heading direction converges. From Eq. (10) and the controller in Eq. (9), in the resilient state we have

$$\frac{\mathrm{d}\lambda_2}{\mathrm{d}t} = \sum_i \frac{\partial \lambda_2}{\partial \mathrm{x}_i} \mathrm{v}_i. \tag{12}$$

The derivatives of f_{ij} are such that the off-diagonal entries of $\left[\frac{\partial L}{\partial x_{k,\alpha}}\right]_{ij}$ can only be non-zero for $k \in \{i, j\}$. Since $\frac{\partial f_{ij}}{\partial x_{i,\alpha}} = -\frac{\partial f_{ij}}{\partial x_{j,\alpha}}$, the off diagonal entries of $\sum_k \frac{\partial L}{\partial x_{k,\alpha}}$ for each spatial dimension $\alpha \in 1 \ldots d$ are

$$\begin{aligned}\sum_k -\frac{\partial f_{ij}}{\partial x_{k,\alpha}} v_{k,\alpha} &= -\frac{\partial f_{ij}}{\partial x_{i,\alpha}} v_{i,\alpha} - \frac{\partial f_{ij}}{\partial x_{j,\alpha}} v_{j,\alpha} \\ &= \frac{\partial f_{ij}}{\partial x_{i,\alpha}} \left(v_{j,\alpha} - v_{i,\alpha}\right)\end{aligned}$$

Furthermore, the diagonal entries are

$$\begin{aligned}\sum_k \sum_j \frac{\partial f_{ij}}{\partial x_{k,\alpha}} v_{k,\alpha} &= \sum_j \sum_k \frac{\partial f_{ij}}{\partial x_{k,\alpha}} v_{k,\alpha} \\ &= \sum_j \frac{\partial f_{ij}}{\partial x_{i,\alpha}} v_{i,\alpha} + \frac{\partial f_{ij}}{\partial x_{j,\alpha}} v_{j,\alpha} \\ &= \sum_j \frac{\partial f_{ij}}{\partial x_{i,\alpha}} \left(v_{i,\alpha} - v_{j,\alpha}\right).\end{aligned}$$

The quantity $|v_{i,\alpha} - v_{j,\alpha}|$ approaches zero asymptotically due to the agents running consensus on their heading direction. Therefore

$$\lim_{t \to \infty} \sum_i \frac{\partial L(\mathrm{x})}{\partial x_{i,\alpha}} v_{i,\alpha} = \mathbf{0}_{N \times N},$$

and the rate of convergence is proportional to the rate of convergence to consensus. Since, it is shown in [33] that the convergence rate of W-MSR is exponential, the difference $|v_{i,\alpha} - v_{j,\alpha}|$ decreases exponentially. It can be concluded that since $\frac{\partial f_{ij}}{\partial x_{i,\alpha}}$ are bounded, all elements of $\sum_i \frac{\partial L(\mathrm{x})}{\partial x_{i,\alpha}} v_{i,\alpha}$ go to zero exponentially.

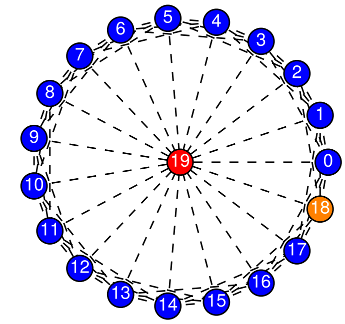

Fig. 3. Initial configuration of the team of robots for the simulations. The red disk represents the non-cooperative robot (19) for all simulations. The orange disk (18) represents the non-cooperative robot for simulations with more than one non-cooperative robot. The dashed lines represent the communication links.

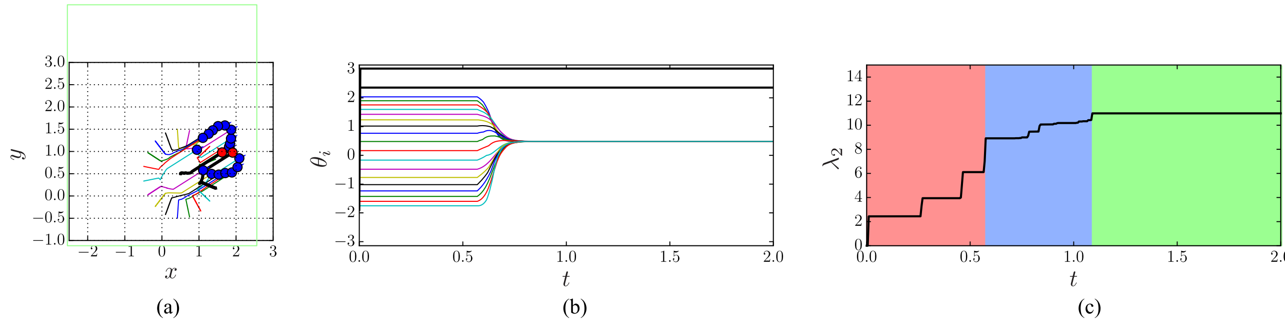

Fig. 4. Simulation results using the proposed controller with two non-cooperative robots. The non-cooperative robots are shown in red in (a) and in (b) the bold black lines show the values they are sharing with their neighbors. In (c) the vulnerable, marginal, and resilient states are shown in red, blue, and green. (a) trajectories. (b) shared heading value. (c) λ_2.

From Eq. (6), we have

$$\begin{aligned}\sum_i \frac{\partial \lambda_2}{\partial x_{i,\alpha}} v_{i,\alpha} &= \sum_i \text{Trace}\left\{ \left[\frac{\mathbf{v}_2 \mathbf{v}_2^T}{\mathbf{v}_2^T \mathbf{v}_2}\right]^T \left[\frac{\partial L(\mathrm{x})}{\partial x_{i,\alpha}}\right] \right\} v_{i,\alpha} \\ &= \text{Trace}\left\{ \left[\frac{\mathbf{v}_2 \mathbf{v}_2^T}{\mathbf{v}_2^T \mathbf{v}_2}\right]^T \sum_i \left[\frac{\partial L(\mathrm{x})}{\partial x_{i,\alpha}}\right] v_{i,\alpha} \right\}.\end{aligned}$$

Thus

$$\lim_{t\to\infty} \sum_i \frac{\partial \lambda_2}{\partial \mathrm{x}_i} \mathrm{v}_i = \mathbf{0},$$

and from Eq. (12)

$$\lim_{t\to\infty} \frac{\mathrm{d}\lambda_2}{\mathrm{d}t} = 0,$$

and the convergence rate is exponential. The exponential convergence of $\frac{\mathrm{d}\lambda_2}{\mathrm{d}t}$ to zero means that for any $\epsilon > 0$ there exists a finite $T > 0$ such that $\forall t > T$, $|\lambda_2(T) - \lambda_2(t)| \le \epsilon$. Choosing ϵ smaller than the hysteresis allows us to conclude that after a finite time the system will stop switching and remain in the resilient state forever. ■

VI. Simulation Results

In our simulations we consider four different scenarios in which the team of robots strive to achieve resilient flocking under a variety of conditions. For each simulation, $N = 20$ robots were initialized in a circular pattern with a non-cooperative node in the center as depicted in Fig. 3. This configuration was chosen to give the non-cooperative robot the most potential to influence the cooperative robots. The switching parameters were chosen as $\eta_1 = 8/6$ and $\eta_2 = 7/6$. The parameter γ in Eq. (1) was chosen to be 0.5 which allowed the robots to maintain λ_2 close to the boundary of the resilient state. Larger values of γ tend to cause the robots to reach a final λ_2 that is much larger than the threshold due to the larger gradients from f_{ij} (See Fig. 1). Since this initial configuration is 4-robust, the conditions are not met to guarantee resilient consensus using the W-MSR algorithm with parameter $F \ge 2$. Simple collision avoidance was included for more realistic simulations. The collision avoidance was implemented to only influence nodes when $\|\mathrm{x}_i - \mathrm{x}_j\| \ll \rho$. In this range $f_{ij} = 1$ and therefore the collision avoidance does not affect $\frac{\mathrm{d}\lambda_2}{\mathrm{d}t}$.

In the fist scenario, we use our proposed method for resilient flocking with $F = 2$. Since the initial graph starts off with $r = 4$, W-MSR does not guarantee resilient convergence until the robustness is increased. Fig. 4 shows the results when there are two non-cooperative robots, shown in red and orange in Fig. 3. The robots initially start in the vulnerable state, shown in red in Fig. 4(c). The vulnerable state controller quickly increases the value of λ_2, as depicted in Fig. 4(c), until it enters the marginal state shown in blue. Every robot updates its velocity, v_i, based on its current consensus value $\mathbf{y}_i$; however, during the update step the non-cooperative robots share a constant value with their neighbors instead of the actual $\mathbf{y}_i$. Fig. 4(b) shows the nodes reaching consensus on the components of $\mathbf{y}_i$, which for clarity is shown as a heading value $\theta_i = \text{atan2}(y_{i,y}, y_{i,x})$. The constant values shared by the non-cooperative robots are shown in bold black. As expected, resilient consensus is achieved.

In the second scenario, a non-cooperative robot shares a time-varying heading, $\theta_i = \theta_{i,0} + \cos(t)$, with its neighbors. The simulation shown in Fig. 5 shows that robots will still achieve resilient consensus. In this case and in the remaining simulations the controller was designed with $F = 1$. It can be seen in Fig. 5(b) that the cooperative nodes are still able to quickly achieve resilient consensus.

In the third scenario the connectivity controller is used, but the classical linear consensus protocol is used instead of W-MSR. The results are shown in Fig. 6. Consensus is achieved, but it is not resilient consensus as the cooperative robots converge to the value of the single non-cooperative robot. We can see that W-MSR is clearly required for resilient consensus. However, if the algebraic connectivity is not kept above the resilience threshold, resilient consensus is not guaranteed. To illustrate this, we present the next scenario.

In the forth scenario, we show a version of the controller where the switching threshold was decreased below the resilience threshold to $2 < 4F = 4$. This simulation can be seen in Fig. 7. In this case, W-MSR is used to attempt to achieve resilient consensus in the presence of a single non-cooperative robot. Although the initial network is such that $r = 4 > 2F + 1$ and is thus resilient to a single non-cooperative robot, the network does not remain sufficiently resilient as the robots move and so the robots fail to reach a resilient consensus. Fig. 7(b) shows the heading values for each robot showing that again the cooperative robots converge to the value shared by the non-cooperative robot.

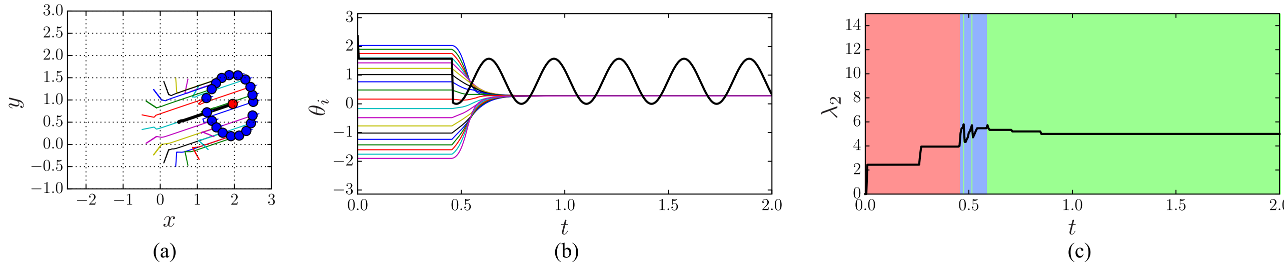

Fig. 5. Simulation results using the proposed controller in which a non-cooperative robot shares a time-varying signal with its neighbors. The bold black line in (b) shows the shared value of the non-cooperative robot. In (c) the vulnerable, marginal, and resilient states are shown in red, blue, and green. (a) trajectories. (b) shared heading value. λ_2.

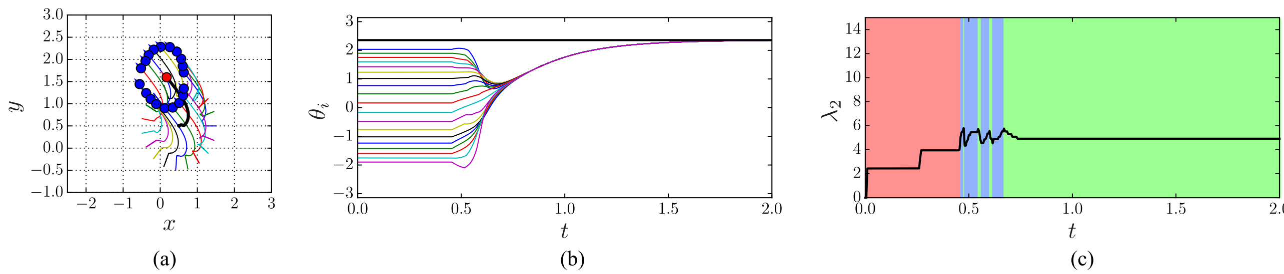

Fig. 6. Simulation results using the proposed controller without W-MSR. The bold black lines in (b) show the shared values of the non-cooperative robots. In (c) the vulnerable, marginal, and resilient states are shown in red, blue, and green. (a) trajectories. (b) shared heading value. λ_2.

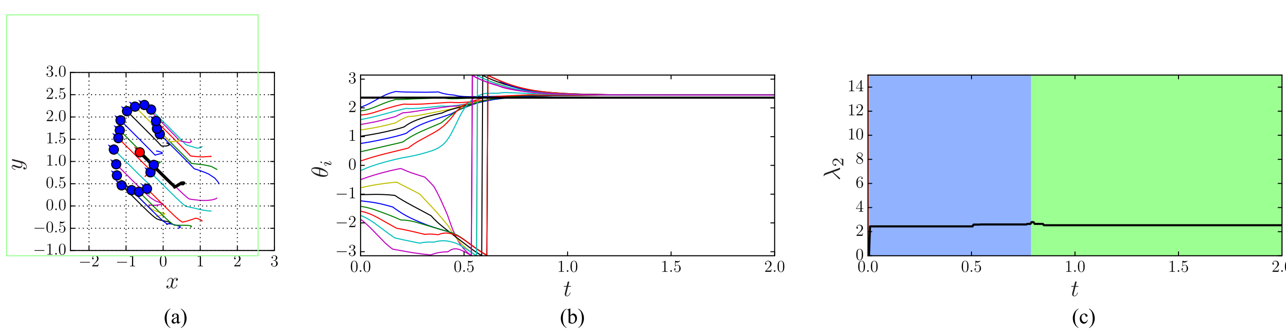

Fig. 7. Simulation results when robots maintain $\lambda_2 > 2$ instead of the resilience threshold in the presence of one non-cooperative robot. The bold black line in (b) shows the shared value of the non-cooperative robot. (a) trajectories. (b) shared heading value. λ_2.

Overall, we observe that the switching controller, when used with the W-MSR consensus algorithm, ensures that the robots achieve resilient consensus. If the robots use linear consensus protocol in place of W-MSR, or do not use the correct resilience threshold, there is no guarantee of resilient consensus.

VII. Conclusion & Future Work

We present a method that enables resilient flocking for mobile robot teams in the presence of non-cooperative robots. Our method builds on the concept of robust network topologies that guarantee resilient consensus. Since determining the exact robustness properties of the network is hard, we make use of a lower bound metric that can be computed efficiently. Combining these results, we propose a dynamic connectivity management strategy that ensures that the communication network topology remains above a critical resilience threshold. We propose a switching control policy that allows a team of mobile robots to achieve resilient consensus on the direction of motion. Finally, we demonstrate the use of our framework for resilient flocking, and show simulation results with groups of holonomic mobile robots.

Our work has the limitation that we have to assume the robots have access to the quantities λ_2 and $\mathbf{v}_2$, which are global properties of the communication graph. The communication graph is defined by the robots' locations. This observation can be made centrally, via an 'eye-in-the-sky' (e.g., GPS) and communicated back to the team. Alternatively, relative locations can be measured locally and in a distributed manner (e.g., via range-and-bearing sensors [34]), allowing the team to compute locations via network localization strategies. This method requires cooperation among robots as they agree on a common set of locations (and consequently, common values for λ_2 and $\mathbf{v}_2$). Similarly, decentralized methods that allow λ_2 and $\mathbf{v}_2$ to be computed directly also rely on truthful, cooperative communication [35], [36]. In the current work, we do not explore the resilient computation of λ_2 and $\mathbf{v}_2$ in a decentralized way. The methods used here to arrive at consensus can be extended to provide other estimates. Alternatively, if we can make the assumption that the robot team starts in a resilient state, we believe that the algorithm presented in [36] can be augmented with a resilient consensus strategy, such as W-MSR, and hence used in our work to provide reliable estimates of λ_2 and $\mathbf{v}_2$ in a decentralized manner. However, a detailed investigation of such approaches is left for future work.

Another direction for future work is an extension of our framework to non-holonomic vehicles, such as in [37], and with higher order dynamics, such as in [20], [38].

References

[1] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.

[2] R. O. Saber and R. M. Murray, "Consensus protocols for networks of dynamic agents," in *Proc. Amer. Controls Conf.*, 2003, pp. 951–956.

[3] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Syst.*, vol. 27, no. 2, pp. 71–82, Apr. 2007.

[4] J. Rodin, *The Resilience Dividend: Being Strong in a World Where Things Go Wrong*. New York, NY, USA: PublicAffairs, 2014.

[5] H. Zhang and S. Sundaram, "Robustness of information diffusion algorithms to locally bounded adversaries," in *IEEE Proc. Amer. Controls Conf.*, 2012, pp. 5855–5861.

[6] H. Zhang, E. Fata, and S. Sundaram, "Robustness of complex networks: Reaching consensus despite adversaries," *CoRR*, vol. abs/1203.6119, 2012, 35 pages.

[7] M. di Bernardo, A. Salvi, and S. Santini, "Distributed consensus strategy for platooning of vehicles in the presence of time-varying heterogeneous communication delays," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 102–112, Feb. 2015.

[8] N. A. Lynch, *Distributed Algorithms*. San Mateo, CA, USA: Morgan Kaufmann, 1996.

[9] J. Hromkovič, *Dissemination of Information in Communication Networks: Broadcasting, Gossiping, Leader Election, and Fault-Tolerance*. New York, NY, USA: Springer, 2005.

[10] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 90–104, Jan. 2012.

[11] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Trans. Autom. Control*, vol. 56, no. 7, pp. 1495–1508, Jul. 2011.

[12] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 4, pp. 766–781, Apr. 2013.

[13] F. Zhang and N. E. Leonard, "Cooperative filters and control for cooperative exploration," *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 650–663, Mar. 2010.

[14] P. Yang, R. A. Freeman, and K. M. Lynch, "Multi-agent coordination by decentralized estimation and control," *IEEE Trans. Autom. Control*, vol. 53, no. 11, pp. 2480–2496, Dec. 2008.

[15] H. Park and S. Hutchinson, "A distributed robust convergence algorithm for multi-robot systems in the presence of faulty robots," in *Proc. IEEE Conf. Intell. Robots Syst.*, 2015, pp. 2980–2985.

[16] H. Park and S. Hutchinson, "An efficient algorithm for fault-tolerant rendezvous of multi-robot systems with controllable sensing range," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 358–365.

[17] J. Cortés, S. Martínez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Trans. Autom. Control*, vol. 51, no. 8, pp. 1289–1298, Aug. 2006.

[18] R. Olfati-Saber , "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401–420, Mar. 2006.

[19] W. Ren and R. W. Beard, *Distributed Consensus in Multi-Vehicle Cooperative Control*. New York, NY, USA: Springer, 2008.

[20] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 863–868, May 2007.

[21] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[22] M. M. Zavlanos, A. Jadbabaie, and G. J. Pappas, "Flocking while preserving network connectivity," in *Proc. IEEE Conf. Decis. Control*, 2007, pp. 2919–2924.

[23] M. C. De Gennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *Proc. IEEE Conf. Decis. Control*, Dec. 2006, pp. 3628–3633.

[24] E. Stump, A. Jadbabaie, and V. Kumar, "Connectivity management in mobile robot teams," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 1525–1530.

[25] M. A. Hsieh, V. Kumar, and L. Chaimowicz, "Decentralized controllers for shape generation with robotic swarms," *Robotica*, vol. 26, no. 05, pp. 691–701, Sep. 2008. [Online]. Available: http://journals.cambridge.org/article_S0263574708004323

[26] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian," *IEEE Trans. Autom. Control*, vol. 51, no. 1, pp. 116–120, Jan. 2006.

[27] M. Franceschelli, M. Egerstedt, and A. Giua, "Motion probes for fault detection and recovery in networked control systems," in *Proc. IEEE Amer. Controls Conf.*, 2008, pp. 4358–4363.

[28] E. M. Shahrivar, M. Pirani, and S. Sundaram, "Robustness and algebraic connectivity of random interdependent networks," *CoRR*, vol. abs/1508.03650, 12 pages, 2015. [Online]. Available: http://arxiv.org/abs/1508.03650

[29] V. Kaibel, "On the expansion of graphs of 0/1-polytopes," in *The Sharpest Cut: The Impact of Manfred Padberg and His Work (Series MPS-SIAM Series on Optimization)*, vol. 4, M. Grötschel, Ed. Philadelphia, PA, USA: SIAM, 2004, pp. 199–216. [Online]. Available: http://arxiv.org/pdf/math.CO/0112146

[30] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Math. J.*, vol. 23, no. 2, pp. 298–305, 1973.

[31] L. Guerrero-Bonilla, A. Prorok, and V. Kumar, "Formations for resilient robot teams," *IEEE Robot. Autom. Lett.*, pp. 841–848, Jan. 2017, doi: 10.1109/LRA.2017.2654550.

[32] G. A. S. Pereira, V. Kumar, and M. F. M. Campos, "Closed loop motion planning of cooperating mobile robots using graph connectivity," *Robot. Auton. Syst.*, vol. 56, no. 4, pp. 373–384, Apr. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2007.08.003

[33] H. J. LeBlanc, "Resilient cooperative control of networked multi-agent systems," Ph.D. dissertation, Electrical Engineering Department, Vanderbilt Univ., Nashville, TN, USA, Dec. 2012.

[34] A. Prorok, A. Bahr, and A. Martinoli, "Low-cost collaborative localization for large-scale multi-robot systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 4236–4241.

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

[36] L. Zareh, M. Sabattini, and C. Secchi, "Distributed laplacian eigenvalue and eigenvector estimation in multi-robot systems," in *Proc. Int. Symp. Distrib. Auton. Robot. Syst.*, 2016, 7 pages.

[37] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in teams of nonholonomic agents," in *Cooperative Control*. New York, NY, USA: Springer, 2005, pp. 229–239.

[38] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Stable flocking of mobile agents, part i: Fixed topology," in *Proc. IEEE Conf. Decis. Control*, 2003, vol. 2, pp. 2010–2015.