Collision-Free Formation Control with Decentralized Connectivity Preservation for Nonholonomic-Wheeled Mobile Robots

Hasan A. Poonawala, *Member, IEEE*, Aykut C. Satici, Hazen Eckert, and Mark W. Spong, *Fellow, IEEE*

***Abstract*—The preservation of connectivity in mobile robot networks is critical to the success of most existing algorithms designed to achieve various goals. The most basic method to preserve connectivity is to have each agent preserve its set of neighbors for all time. More advanced methods preserve a (minimum) spanning tree in the network. Other methods are based on increasing the algebraic graph connectivity, which is given by the second smallest eigenvalue $\lambda_2(\mathcal{L})$ of the graph Laplacian $\mathcal{L}$ that represents the network. These methods typically result in a monotonic increase in connectivity until the network is completely connected. In previous work by the authors, a continuous feedback control method had been proposed which allows the connectivity to decrease, that is, edges in the network may be broken. This method requires agents to have knowledge of the entire network. In this paper, we modify the controller to use only local information. The connectivity controller is based on maximization of $\lambda_2(\mathcal{L})$ and artificial potential functions and can be used in conjunction with artificial potential-based formation controllers. The controllers are extended for implementation on nonholonomic-wheeled mobile robots, and the performance is demonstrated in an experiment on a team of wheeled mobile robots.**

***Index Terms*—Connectivity control, decentralized control, multi-robot systems.**

I. Introduction

MOBILE ROBOT networks afford a robust and inexpensive method for achieving certain coverage tasks or cooperative missions. Many algorithms for achieving tasks using mobile robot networks require that the network maintains connectivity. When the network is connected, any two robots can communicate and share information, even if through several "hops." The problem of maintaining connectivity in mobile robot networks has thus been receiving increasing attention. This problem becomes further complicated when the connectivity depends on the state of the system.

A good review of different methods to control the connectivity can be found in [1]. These methods may be either centralized or decentralized. The key advantage of a decentralized method is that it can scale to a large numbers of robots. An obvious method of maintaining connectivity is to preserve the edges present in the network for all time [2]–[6]. Most decentralized methods of preserving connectivity utilize a variation of this idea. A notable exception is found in [7], where the authors propose algorithms to decide whether edges may be deleted while still ensuring that a spanning subgraph exists, based on local estimates of the network topology. The edges are usually preserved using unbounded artificial potential functions, which suffer from the phenomenon of the overall potential becoming unbounded (generating a large control effort due to the gradient) whenever a new edge is added. This is overcome by using a hysterisis protocol. A method that uses bounded control to tackle this phenomenon is given in [8]; however, edges are never broken.

The presence of a spanning subgraph in the network can be inferred from the spectral properties of the Graph Laplacian $\mathcal{L}$ [9]. Thus, another method for maintaining connectivity amongst a group of mobile robots is to maximize the second smallest eigenvalue of the graph Laplacian [10]. In this method, the edge strengths are nonincreasing functions of the distance between robots. The resulting graph is always completely connected, as seen in the simulation results in [10]. This method is effective for solving rendezvous problems, which was a primary goal in [10]. It can also can be extended to some other applications [1], [10], such as tracking a leader.

When the task to be achieved is formation control or area coverage, the tendency of the network to become completely connected is undesirable. Instead, the goal is to prevent disconnection during the execution of such tasks. Hence, we would wish to allow link deletions when suitable, without relying on higher level planning or decision making. In a previous paper [11], we modify the approach in [10] so that the resulting behavior is such that global connectivity is maintained, but not increased until the network is completely connected. The result is that links may be broken under the influence of additional control objectives (such as exploration or coverage) without losing global connectivity. The method is easier to implement than the one in [7]; however, our controller requires each agent to have access to the positions of all other nodes.

Manuscript received February 10, 2014; revised September 5, 2014; accepted October 3, 2014. Date of publication December 8, 2014; date of current version June 16, 2015. This work was supported in part by the National Science Foundation Grants ECCS 07-25433 and CMMI-0856368, in part by the University of Texas STARs program, and in part by the DGIST R&D Program of the Ministry of Education, Science and Technology of Korea under Grant 12-BD-0101. Recommended by Associate Editor C. Canudas de Wit.

H. A. Poonawala, H. Eckert, and M. W. Spong are with the Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: hasanp@utdallas.edu; hxe110130@utdallas.edu; mspong@utdallas.edu).

A. C. Satici is with MathWorks Inc., Natick, MA 01760 USA (e-mail: Aykut.Satici@mathworks.com).

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

Digital Object Identifier 10.1109/TCNS.2014.2378876

To overcome this, we present a decentralized version of the connectivity controller which only requires local network information. This decentralized controller is the main contribution of this paper. It relies on the connectivity estimation algorithm presented in [12]. A further contribution in this work is the experimental validation of this decentralized controller on a network of nonholonomic wheeled mobile robots (WMRs).

Connectivity controllers that prevent edge deletions or converge to complete networks limit the set of formations that can be commanded. The controllers we present allow a larger set of formations that can be achieved, which can be modified using a parameter in the control. This method is decentralized and, hence, can scale to networks with a large number of robots.

II. Background

In this section, we give a brief recount of concepts from graph theory used to model the connectivity of a mobile robot network.

The team of mobile robots consists of $N \in \mathbb{N}$ mobile robots. The configuration of the kth mobile robot is $\mathbf{x}_k \in \mathbb{R}^2$, where $k \in \{1, 2, \ldots, N\}$. The configurations of each mobile robot can be combined in the obvious way to form the configuration of the team of mobile robots $\mathbf{x} \in \mathbb{R}^{2N}$.

A weighted graph G is a tuple consisting of a set of vertices V (also called nodes) and a function W, that is

$$G = (V, W)$$

where $V = \{1, \ldots, N\}$ denotes the set of nodes. The function $W : V \times V \times R_+ \to R_+$ is used to compute the weights of the edges in G, such that

$$w_{ij}(t) = W(i, j, t). \tag{1}$$

If $w_{ij}(t)$ =0, then there is no connection between nodes i and j. We obtain the edge weights using bump functions, commonly used as gluing objects of differential geometry

$$\psi(x) = \begin{cases} 1 & \text{if } x \leq \rho_1 \\ \frac{\exp\left(-\frac{1}{\rho_2 - x}\right)}{\exp\left(-\frac{1}{\rho_2 - x}\right) + \exp\left(\frac{1}{\rho_1 - x}\right)} & \text{if } \rho_1 \leq x \leq \rho_2 \\ 0 & \text{if } \rho_2 \leq x. \end{cases}$$

One of the advantages of bump functions is that they are smooth objects and can thus be differentiated as many times as required. If we take the distance d_{ij} between two robots as the domain of $\psi(x)$, we obtain a smooth weighting $w_{ij} = \psi(d_{ij})$ from full connectivity to no connectivity for any two robots.

The edge weights give rise to the graph Laplacian $\mathcal{L}(G) \in \mathbb{R}^{N \times N}$ defined as

$$\mathcal{L}_{ij}(t) = \begin{cases} -w_{ij}(t) & \text{if } i \neq j \\ \sum_{k \neq i} w_{ik}(t) & \text{if } i = j. \end{cases}$$

The Laplacian gives us a measure of the connectivity of the graph G since the number of connected components in the graph is equal to the number of zero eigenvalues of $\mathcal{L}(G)$ [9]. Thus, for the graph to be connected, only one eigenvalue of $\mathcal{L}(G)$ must be zero. The second smallest eigenvalue $\lambda_2(\mathcal{L}(G))$ thus becomes an indicator of connectivity in the graph. Note that since the w_{ij} depends on the distance between agents, the Laplacian is a function of $\mathbf{x}$.

The Laplacian $\mathcal{L}(G)$ can be converted to a matrix $\mathcal{M}(G) \in \mathbb{R}^{N-1 \times N-1}$, whose eigenvalues are the largest $N-1$ eigenvalues of $\mathcal{L}(G)$. Thus, the determinant of $\mathcal{M}(G)$ vanishes if and only if $\lambda_2(\mathcal{L}(G))$ vanishes. This means that $\det \mathcal{M}$ is also an indicator of the connectivity of the graph. The matrix $\mathcal{M}(G)$ [13] is given by

$$\mathcal{M}(G) = P^T \mathcal{L}(G) P \tag{2}$$

where $P \in \mathbb{R}^{N \times N-1}$ satisfies $P^T \mathbf{1} = 0$ and $P^T P = I_{N-1}$.

For each $k \in V$, we can define the time-varying neighbor set $\mathcal{N}_k \subset V$ as

$$\mathcal{N}_k = \{j \in V | w_{jk} \neq 0\} \tag{3}$$

and its closure $\overline{\mathcal{N}_k}$ given by

$$\overline{\mathcal{N}_k} = k \cup \mathcal{N}_k. \tag{4}$$

Note that $\overline{\mathcal{N}_k} \subseteq V$ has order $N_k + 1$, where $N_k \in \mathbb{N}$ is the number of neighbors of k in G. Each member of $\overline{\mathcal{N}_k}$ can be assigned a position in $V_k = \{1, 2, \ldots, N_k + 1\}$. This is achieved through the map $\pi_k : V \to V_k$. We can now define the subgraph $G_k = (V_k, W)$, which has Laplacian $\mathcal{L}_k = \mathcal{L}(G_k)$ and reduced Laplacian $\mathcal{M}_k = \mathcal{M}(G_k)$. The sets V_k change with time; however, the dependence on time is suppressed for convenience.

For the rest of this paper, $\mathcal{L}$ refers to the Laplacian of the graph G, and $\mathcal{L}_k$ refers to that of the graph G_k for each k with similar convention applied to $\mathcal{M}$. The dependence of these graphs and matrices on the configuration $\mathbf{x}$ has been suppressed for convenience.

III. Control Design

Consider the scenario where we would like a team of robots to stay connected to each other while performing some other task. This task might be, for example, arranging themselves in a formation or exploration, etc. These two requirements can be mathematically restated as bounding the second smallest eigenvalue of the Laplacian away from zero while each robot tracks either (possibly time varying) absolute or relative positions.

A connectivity preserving controller that achieves this behavior was presented in [11]. The method can be used with formation and collision-avoidance controllers satisfactorily. A chief drawback of the method was that it was centralized, due to the presence of the term $\mathcal{M}^{-1}$ in each agent's control law. The main result of this paper is a decentralized version of that connectivity controller, presented in Section III-B. We also analyze the performance of the decentralized connectivity controller when used in conjunction with formation or collision-avoidance control laws in Section III-C and D, respectively.

The controllers in the following sections assume that each agent has the following dynamics:

$$\dot{\mathbf{x}}_k = \tau_k \tag{5}$$

where $\tau_k \in \mathbb{R}^2$ is the control input of the kth mobile robot. In the next subsection, we recount the centralized control law from [11] and its properties.

A. Centralized Connectivity Control Law

The connectivity control law in [13] was shown to maintain the connectivity of a network; however, it restricts the possible equilibrium configurations of the team of mobile robots to those where the network is completely connected. The control was based on the gradient of a potential function whose argument is $\det \mathcal{M}$, where $\mathcal{M}$ is the reduced Laplacian of the graph. In [11], the following potential function was proposed:

$$D(\mathbf{x}) := \det\left(\mathcal{M}(\mathbf{x})\right)$$
$$V_c(D) = \left(\min\left\{0, \frac{D^2 - \bar{\alpha}^2}{D^2 - \underline{\alpha}^2}\right\}\right)^2$$

where $\underline{\alpha} \in \mathbb{R}$ is a desired lower bound for $\det \mathcal{M}$, and $\bar{\alpha} \in \mathbb{R}$ is a threshold for $\det \mathcal{M}$ above which the connectivity control is inactive. The function $V_c(D)$ and its gradient are unbounded whenever D approaches $\underline{\alpha}$ and is zero whenever the determinant is greater than $\bar{\alpha}$. Thus, using the gradient of the potential function above results in a control law which guarantees that $\det \mathcal{M} > \underline{\alpha}$, which implies that the graph is always connected. Moreover, since $\det \mathcal{M}$ is bounded from above, this ensures that $\lambda_2(\mathcal{L})(t)$ has a nonzero lower bound which we can select. The performance of consensus-based algorithms improves with an increase in $\lambda_2(\mathcal{L})$; hence, this feature would be beneficial in such a scenario.

Another feature of the control law is that the connectivity control law is inactive if $\det \mathcal{M} > \bar{\alpha}$. Thus, by choosing $\bar{\alpha}$ and $\underline{\alpha}$ appropriately, we can make the control law unresponsive to changes in connectivity until the connectivity becomes lower than desired. This reduces the interference of the connectivity controller with the primary tasks that the team of agents is supposed to achieve, yet guarantees that connectivity will be maintained.

Partial derivatives of V_c with respect to the coordinates x_k and y_k of the kth robot are

$$\frac{\partial V_c}{\partial x_k} = \begin{cases} 0 & \text{if } D(\mathbf{x}) \le \underline{\alpha} \\ \beta(\mathbf{x})\mathrm{tr}\left(\mathcal{M}^{-1}\frac{\partial \mathcal{M}}{\partial x_k}\right) & \text{if } \underline{\alpha} < D(\mathbf{x}) < \bar{\alpha} \\ 0 & \text{if } \bar{\alpha} \le D(\mathbf{x}) \end{cases}$$
$$\frac{\partial V_c}{\partial y_k} = \begin{cases} 0 & \text{if } D(\mathbf{x}) \le \underline{\alpha} \\ \beta(\mathbf{x})\mathrm{tr}\left(\mathcal{M}^{-1}\frac{\partial \mathcal{M}}{\partial y_k}\right) & \text{if } \underline{\alpha} < D(\mathbf{x}) < \bar{\alpha} \\ 0 & \text{if } \bar{\alpha} \le D(\mathbf{x}) \end{cases}$$

where

$$\beta(\mathbf{x}) = 4\frac{\left(\bar{\alpha}^2 - \underline{\alpha}^2\right)\left(D(\mathbf{x})^2 - \bar{\alpha}^2\right)}{\left(D(\mathbf{x})^2 - \underline{\alpha}^2\right)^3} D(\mathbf{x})^2 < 0.$$

The following result was shown in [11].

Proposition III.1: Under the control law

$$\tau_k = -\nabla_{\mathbf{x}_k} V_c(\mathbf{x}) = -\beta(\mathbf{x}) \begin{bmatrix} \mathrm{tr}\left(\mathcal{M}^{-1}\frac{\partial \mathcal{M}}{\partial x_k}\right) \\ \mathrm{tr}\left(\mathcal{M}^{-1}\frac{\partial \mathcal{M}}{\partial y_k}\right) \end{bmatrix} \tag{6}$$

the first-order agents with dynamics (5) converge to the set $E = \{\mathbf{x} \in \mathbb{R}^{2N} : \det(\mathcal{M}(\mathbf{x})) \ge \bar{\alpha}\}$, and the graph G is connected for all time.

B. Decentralized Connectivity Control Law

The connectivity control (6) causes the robots to move along the gradient of $\det \mathcal{M}$ with a "gain" that is determined by the magnitude of $\det \mathcal{M}$. When computing the control law, each agent must know the matrix $\mathcal{M}$, which makes the control law centralized. The gradient $\partial \mathcal{M}/\partial x_k$ consists of terms of the form $\partial w_{kj}/\partial \mathbf{x}_k$ which vanish when robot j is not a neighbor of robot k and, hence, each robot may compute it by knowing information from neighbors only.

Thus, when attempting to form a decentralized control version of (6), we must eliminate the use of $\mathcal{M}^{-1}$ and the use of $\det \mathcal{M}$ to determine the "gain." The first elimination is achieved by making the agents move based on the gradient of the "local" matrix $\mathcal{M}_k$ instead of the matrix $\mathcal{M}$. The second elimination is achieved by using the algebraic connectivity $\lambda_2(\mathcal{L})$ in order to calculate the "gain."

Thus, the decentralized connectivity controller becomes

$$\tau_{k,c} = -k_c \beta_k\left(\hat{\lambda}_2^k\right) \begin{bmatrix} \mathrm{tr}\left(\mathcal{M}_k^{-1}\frac{\partial \mathcal{M}_k}{\partial x_k}\right) \\ \mathrm{tr}\left(\mathcal{M}_k^{-1}\frac{\partial \mathcal{M}_k}{\partial y_k}\right) \end{bmatrix} \tag{7a}$$
$$\beta_k\left(\hat{\lambda}_2^k\right) = \min\left\{\left(\left(\lambda_2^k\right)^2 - \bar{\alpha}^2\right), 0\right\} \tag{7b}$$

where $k_c > 0$, $\hat{\lambda}_2^k$ is the kth agent's estimate of $\lambda_2(\mathcal{L})$, $\beta_k \le 0\ \ \forall k \in V$, and $\bar{\alpha}$ is the upper threshold defined in Section III-A.

The estimates $\hat{\lambda}_2^k$ are obtained by using the method proposed in [12]. The kth agent estimates the kth component of v_2 (where $\mathcal{L}v_2 = \lambda_2 v_2$) by the update law

$$\dot{\eta}^k = -k_1 z^{k,1} - k_2 \sum_{j \in \mathcal{N}_k} A_{kj}(\eta^k - \eta^j) - k_3(z^{k,2} - 1)\eta^k \tag{8}$$

where η^k is the estimate of the kth component of v_2; k_1, k_2, and k_3 are estimator gains, and $z^{k,1}$ and $z^{k,2}$ are the kth agent's estimate of the average of $\{\eta^k\}$ and $\{(\eta^k)^2\}$, respectively. Note that all terms on the right-hand side of (8) consist of information known by agents in $\overline{\mathcal{N}}_k$. Each agent obtains the estimates of the average of $\{\eta^k\}$ and that of $\{(\eta^k)^2\}$ using two copies of the following PI consensus estimators:

$$\dot{z}^k = \gamma(\alpha^k - z^k) - K_P \sum_{j \in \mathcal{N}_k}(z^k - z^j) + K_I \sum_{j \in \mathcal{N}_k}(w^k - w^j)$$
$$\dot{w}^k = -K_I \sum_{j \in \mathcal{N}_k}(z^k - z^j) \tag{9}$$

where z^k is an estimate which converges to $(1/N)\sum_{i=1}^{N}\alpha^i$; $\gamma > 0$ is the rate new information replaces old information; and K_P, K_I are estimator gains. One of the estimators runs with input $\alpha^{k,1} = \eta^k$ and states $(z^{k,1}, w^{k,1})$, and another with input $\alpha^{k,2} = (\eta^k)^2$ and states $(z^{k,2}, w^{k,2})$.

Finally, each agent computes its estimate of $\lambda_2(\mathcal{L})$ as

$$\hat{\lambda}_2^k = \frac{k_3}{k_2}(1 - z^{k,2}).$$

If the adjacency matrix A is constant, then for appropriate estimator gains, each agent's estimate will converge to $\lambda_2(\mathcal{L})$, as shown in [12].

The control law (7) can be reduced to a linear combination of the relative position vectors from agent k to its neighbors, where the coefficients are non-negative. This property is important since it shows that the control law requires only local information in order to be implemented (provided that $\lambda_2(\mathcal{L})$ is known to each agent). This structure is also used in subsequent proofs in this paper. The formal statement follows:

Lemma III.1: The instantaneous direction of motion of each robot $k \in V$ under any control of the form

$$\tau_k = -\beta \begin{bmatrix} \mathrm{tr}\left(\mathcal{M}_k^{-1}\frac{\partial \mathcal{M}_k}{\partial x_k}\right) \\ \mathrm{tr}\left(\mathcal{M}_k^{-1}\frac{\partial \mathcal{M}_k}{\partial y_k}\right) \end{bmatrix} \tag{10}$$

is a positive combination of the vectors $(\mathbf{x}_j - \mathbf{x}_k)$, where $j \in \mathcal{N}_k$ and $\beta \leq 0$.

Proof: See [14] ■

The result above can be used to establish the convergence behavior of the control law (7) when no additional control task is present. The algebraic connectivity will increase until it is larger than $\bar{\alpha}$. The result is stated below:

Proposition III.2: Consider the control law for the kth robot given by (7). If initially $\lambda_2(\mathcal{L})(t_0) > 0$, then the configuration of the agents converges to the set $E = \{\mathbf{x} \in \mathbb{R}^{2N} : \lambda_2(\mathcal{L}) \geq \bar{\alpha}\}$.

Proof: By Lemma III.1, each agent moves toward the interior of the convex hull $\mathrm{CH}(V_k)$ determined by its neighbor set. This is because the velocity of agent k is along the line joining a point in $\mathrm{CH}(V_k)$ to x_k. To see this, we remember that the convex hull of V_k is defined as the set of all convex combinations of $\mathbf{x}_j$ where $j \in \mathcal{N}_k$, the neighbor set of k. Any point in $\mathrm{CH}(V_k)$ has the form

$$q = \sum_{j \in \overline{\mathcal{N}}_k} \sigma_j x_j$$

for some $\sigma_j \geq 0$ and $\sum_{j\in\overline{\mathcal{N}}_k}\sigma_j = 1$. Thus, a vector that points from x_k to q has the form

$$\begin{aligned} q - x_k &= (\sigma_k - 1)x_k + \sum_{j\in\mathcal{N}_k}\sigma_j x_j \\ &= \left(-1 + \sum_{j\in\overline{\mathcal{N}}_k}\sigma_j\right)x_k + \sum_{j\in\mathcal{N}_k}\sigma_j(x_j - x_k) \\ &= \sum_{j\in\mathcal{N}_k}\sigma_j(x_j - x_k) \end{aligned}$$

which is exactly of the form derived in Proposition III.1.

Moreover, the agents defining the convex hull $\mathrm{CH}(V)$ of the positions of all agents in the network will move into that convex hull so that the perimeter will decrease by a simple application of the triangle inequality on Euclidean space. Stacking the lengths $\{d_i\}_1^m$ of the edges defining the convex hull in a vector v, this means that $\|v\|_1$ is monotonously decreasing with a lower bound 0. By the Bolzano-Weierstrass Theorem, each entry of the vector v approaches zero. Since the perimeter is shrinking, any agent inside the convex hull must have smaller distances to their neighbors than the perimeter. As a result, there exists a $T > 0$ such that when $t > T$, $d_{ij}(t) < \epsilon$ for any $\epsilon > 0$. $\lambda_2(\mathcal{L})$ is a nondecreasing function of the edge weights w_{ij} [12]. Due to our choice of relationship between w_{ij} and inter-robot distances d_{ij}, $\lambda_2(\mathcal{L})$ becomes a nonincreasing function of each distance d_{ij}. The algebraic connectivity is a continuous function of the distances d_{ij} and is bounded above by N. This bound is achieved when $d_{ij} = 0\ \forall i, j \in V$. Thus, as the distances d_{ij} decrease, $\lambda_2(\mathcal{L})$ will eventually increase until $\beta_k \equiv 0$. This can occur only when $\lambda_2(\mathcal{L}) \geq \bar{\alpha}$. To see this, note that if $\beta_k \equiv 0\ \forall k \in V$, then A is constant and $\hat{\lambda}_2^k \to \lambda_2(\mathcal{L})$. If $\lambda_2(\mathcal{L}) < \bar{\alpha}$, then after some time $\beta_k \neq 0$. Thus, if $\beta_k \equiv 0$, then $\lambda_2(\mathcal{L}) \geq \bar{\alpha}$, implying that the agents will converge to E. ■

C. Decentralized Connectivity Preserving Formation Controller

In this section, we develop on the connectivity controller presented in Section III-B by adding a formation controller on top of it.

We define a quadratic potential function for each robot k, $V_{fk}(\mathbf{x}_k)$, with a minimum located at the desired position $\mathbf{x}_{kd}$. Just as $\mathbf{x}_i$ are combined to form $\mathbf{x}$, we can obtain the desired configuration $\mathbf{x}_d$. The sum of the contributions of each robot gives rise to the formation potential function $V_f(\mathbf{x})$

$$\begin{aligned} V_{fk} &= \frac{1}{2}\langle \mathbf{x}_k - \mathbf{x}_{kd}, \mathbf{x}_k - \mathbf{x}_{kd}\rangle \\ V_f &= \sum_i^N V_{fk} \end{aligned}$$

where the brackets $\langle\cdot,\cdot\rangle$ represent the usual Euclidean inner product of vectors.

In the remainder of the subsection, we shall assume that the control law for each robot is given by

$$\tau_k = \tau_{k,c} + \tau_{k,f} \tag{11}$$

where $\tau_{k,c}$ is the decentralized connectivity controller (7), and $\tau_{k,f}$ is given by

$$\tau_{k,f} = -k_f \nabla_{\mathbf{x}_i} V_f \tag{12}$$

and $k_c, k_f > 0$ are control gains.

For any $\mathbf{x}$ such that $\|\mathbf{x}\| < \|\mathbf{x}\|_{\max} < \infty$, we can see that $\|\tau_{k,f}\| \leq \tau_{k,f,\max}$, for some $\tau_{k,f,\max} > 0$.

Proposition III.3: Consider the control law for the kth robot given by (11). If the robots are started such that $\lambda_2(\mathcal{L})(t_0) > 0$ and $|\hat{\lambda}_2^k(t) - \lambda_2(\mathcal{L})(t)| < \bar{\alpha}\ \forall\ k \in V, t \geq t_0$, then $\lambda_2(\mathcal{L})(t) > 0\ \forall t \geq t_0$.

Proof: See the Appendix. ■

Thus, the bounded formation control signal cannot lead to disconnection of the network G, since the connectivity control will eventually dominate any such signal and prevent the cut-bridge from being broken.

Proposition III.3 requires that the estimation error be less than $\bar{\alpha}$ for all future times. Suppose the estimation error satisfies this condition at t_0. Then, there is some time interval $[t_0, t_0 + T]$, $T > 0$ during which $|\lambda_2(\mathcal{L})(t) - \hat{\lambda}_2^k(t)| < \bar{\alpha}$. Then, by Proposition III.3, the connectivity control preserves connectivity and is bounded over this interval. We can tune the estimator gains high enough so that given this bounded velocity, the errors in estimation do not exceed $\bar{\alpha}$. This argument can be extended indefinitely to show that the estimation errors and control are bounded appropriately for all time.

We now show how bounded agent velocities lead to bounded estimation errors. The PI consensus estimator can be rewritten as

$$\begin{aligned} \dot{z} &= \gamma(\alpha - z) - K_P \mathcal{L}^* z + K_I \mathcal{L}^* w \\ \dot{w} &= -K_I \mathcal{L}^* z \end{aligned} \tag{13}$$

where $\mathcal{L}^*$ is the Laplacian corresponding to the unweighted adjacency matrix. We differentiate the above equation to obtain

$$\ddot{z} + (\gamma I + K_P \mathcal{L}^*)\dot{z} + K_I^2 \mathcal{L}^* z = \gamma \dot{\alpha}. \tag{14}$$

If $\dot{\alpha} = 0$ and the graph is connected, then $z \to \mathrm{Ave}(\alpha)\mathbf{1}$, as shown in [12]. The convergence rate is exponential, and can be controlled by setting the gains K_I, K_P. If $\dot{\alpha} \neq 0$, we can expect z to converge to a ball centered at $\mathrm{Ave}(\alpha)\mathbf{1}$ whose radius depends on $\|\dot{\alpha}\|$. The rate of convergence of z can be made much faster than the rate of convergence of (8), so that $\dot{\eta}$ is small. This is achieved by making the gains K_p, K_I much larger than the gains k_1, k_2, and k_3. This means that $\|\dot{\alpha}\|$ is small, so that the error in z is small.

When this happens, the decentralized estimator (8) for v_2 behaves like the centralized estimator, which was shown to converge [12] from almost any initial condition for appropriate values of the gains k_1, k_2, and k_3. If the terms A_{kj} change with time, v_2 also changes with time, and η will converge to a ball around v_2 whose radius depends on $\dot{A}$ (and the estimator gains), provided the time-varying graph is connected at all times. If $\dot{A}$ is bounded, then our error in estimation of v_2 and, in turn, of λ_2 will be bounded.

We can see that $\dot{A}$ depends on the magnitude of τ

$$\dot{A} = \frac{\partial A}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial A}{\partial \mathbf{x}} \tau$$

where $\partial A / \partial \mathbf{x}$ is bounded because $\partial w_{ij} / \partial d_{ij}$ is bounded. Thus, if τ is bounded, then so is $\dot{A}$ and we can increase the estimator gains such that the errors in the estimation of $\lambda_2(\mathcal{L})$ are bounded.

Using the aforementioned framework, we can ensure that (11) achieves the task of preserving connectivity of the mobile network. The next issue to address is its performance on the formation control task.

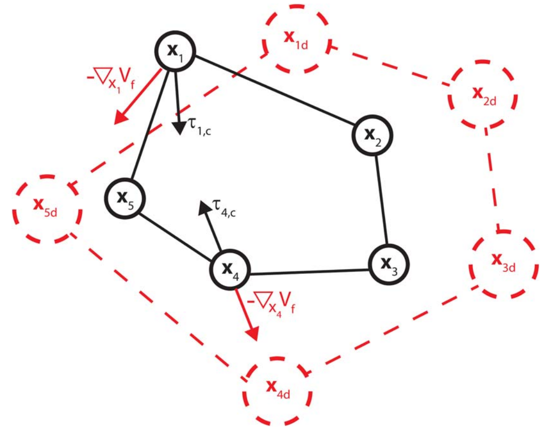

Fig. 1. If an agent x_1 is outside the convex hull of the desired formation (depicted by dotted red lines), the formation controller and the connectivity controller cannot cancel each other.

If the kth agent implements $\tau_{k,f}$, it is easy to show that $\mathbf{x}(t) \to \mathbf{x}_d$ as $t \to \infty$. When the agents implement control law (11), the following behavior can be guaranteed for any initial condition $\mathbf{x}(t_0)$.

Theorem III.2: Let $X_d = \{\mathbf{x}_{1d}, \mathbf{x}_{2d}, \ldots, \mathbf{x}_{Nd}\}$ be the set of desired positions of the team of mobile robots. Suppose the control effort for each robot k is given by (11). Then, each robot k converges to a configuration $\mathbf{x}_k$ contained in the convex hull $\mathrm{CH}(X_d)$ of the desired formation, that is, $\mathbf{x}_k \in \mathrm{CH}(X_d)\ \forall k \in \{1, 2, \ldots, N\}$.

Proof: Under the action of the control law (11), the agents reach the set of configurations $\mathbf{x} \in \mathbb{R}^{2N}$ where for all k, both of the following hold:

$$k_c \beta_k \mathrm{tr}\left(\mathcal{M}_k^{-1} \frac{\partial \mathcal{M}_k}{\partial x_k}\right) + k_f(x_k - x_{kd}) = 0$$

$$k_c \beta_k \mathrm{tr}\left(\mathcal{M}_k^{-1} \frac{\partial \mathcal{M}_k}{\partial y_k}\right) + k_f(y_k - y_{kd}) = 0$$

Since $\beta_k < 0$, this is equivalent to the statement that the angle between the vectors $[\mathrm{tr}(\mathcal{M}_k^{-1}(\partial \mathcal{M}_k/\partial x_k))\ \ \mathrm{tr}(\mathcal{M}_k^{-1}(\partial \mathcal{M}_k/\partial x_k))]^T$ and $[x_k - x_{kd}\ \ y_k - y_{kd}]^T$ is π rad. Due to Proposition III.1, the former always points into $\mathrm{CH}(V)$, so this is only possible if for each k, $\mathbf{x}_k \in \mathrm{CH}(X_d)$ (see Fig. 1). ■

Remark 1: Theorem III.2 provides a way to move the robots into the convex hull defined by the desired formation while maintaining connectivity. Even though the claims of the theorem are weaker, in any simulation, the robots converge to the desired formation $\mathbf{x}_d$, provided it is selected such that the $\lambda_2(\mathcal{L}(\mathbf{x}_d)) \geq \bar{\alpha}$.

Remark 2: The control due to connectivity becomes unbounded as $\lambda_2(\mathcal{L}_k) \to 0$, when $\hat{\lambda}_2^k < \bar{\alpha}$. Finite errors in formation yield finite control effort; hence, even if the desired formation is disconnected, the network will never become disconnected.

D. Decentralized Connectivity Preserving Formation Controller With Collision Avoidance

We can add another potential function V_a designed to introduce collision-avoidance behavior, to work in collaboration with the existing ones. In this way, we can guarantee that the robots do not collide while they move toward the desired formation. We use the avoidance (potential) functions as defined in [15] by

$$V_{aij} = \left(\min\left\{0, \frac{d_{ij}^2 - R^2}{d_{ij}^2 - r^2}\right\}\right)^2 \tag{15}$$

where d_{ij} is the Euclidean distance between robots i and j, r and R defines the avoidance region and reaction region, respectively. The potential functions are designed such that if the robots are started away from the avoidance region $\Omega_{ij} = \{x : \|x_i - x_j\| \leq r\}$, they never enter this region. The reaction region, on the other hand, given by $\mathcal{D}_{ij} = \{x : \|x_i - x_j\| \leq R\}$, is the region where robot i reacts to the presence of robot j. Ideally, we would set $R \leq \rho_1$, where ρ_1 was defined in Section II.

The sum of the pairwise potentials (15) between robots i and j constitutes the total avoidance potential function

$$V_a = \sum_{i=1}^{N} \sum_{j \neq i, j=1}^{N-1} \frac{1}{2} V_{aij}$$

and the collision-avoidance control for the kth agent is $\tau_{k,a} = -k_a \nabla_{\mathbf{x}_k} V_a$, where $k_a > 0$. Thus, the form of the control law for robot k with collision avoidance would be

$$\tau_k = \tau_{k,c} + \tau_{k,f} + \tau_{k,a} \tag{16}$$

where $\tau_{k,c}$ is the decentralized connectivity controller presented in Section III-B and $\tau_{k,f}$ is the formation control presented in Section III-C.

E. Extension to Wheeled Mobile Robots

In the case of nonholonomic-wheeled mobile robots, the kinematics are modeled by the nonlinear ordinary differential equations

$$\begin{aligned} \dot{x}_k &= v_k \cos(\theta_k) \\ \dot{y}_k &= v_k \sin(\theta_k) \\ \dot{\theta}_k &= \omega_k \end{aligned} \tag{17}$$

where $x_k \in \mathbb{R}$ and $y_k \in \mathbb{R}$ are the Cartesian coordinates, $\theta_k \in [0, 2\pi)$ is the orientation of the kth robot with respect to a fixed reference frame, and v_k, ω_k are the linear and angular velocity inputs, respectively. We would like the controllers developed so far to work with this system dynamics, rather than the first-order integrators (5).

The idea will be to turn the robot to the desired orientation, dictated by the direction of the controller derived for robots with dynamics (5). Let $X : \mathbb{R}^{2N} \to T\mathbb{R}^{2N} \cong \mathbb{R}^{2N} \times \mathbb{R}^{2N}$ be the vector field that we want our wheeled mobile robots to follow in the x and y directions. This vector field extends to the case when the underlying configuration space for each robot is $\mathbb{R}^2 \times S^1$ by defining $\tilde{X} : \mathbb{R}^{2N} \times S^N \to \mathbb{R}^{2N} \times S^N \times \mathbb{R}^{2N} \times \mathbb{R}^N$ such that $\tilde{X} = (q, \theta, X^f, Y^f)$, where (q, θ) denotes the configuration in $\mathbb{R}^{2N} \times S^N$, X^f denotes the fiber component of the vector field X, and Y^f denotes the fiber component of any vector field $Y : S^N \to S^N \times \mathbb{R}^N$

$$\theta_{kd} = \arctan_2\left(\left\langle X, \frac{\partial}{\partial y_k}\right\rangle, \left\langle X, \frac{\partial}{\partial x_k}\right\rangle\right). \tag{18}$$

Define the orientation error $e_{\theta_k} = \theta_k - \theta_{kd}$. Let us also define the desired velocity vector to be

$$\tau_{kd} := \left(\left\langle X, \frac{\partial}{\partial x_k}\right\rangle, \left\langle X, \frac{\partial}{\partial y_k}\right\rangle\right). \tag{19}$$

Note that the desired orientation θ_{kd} is the angle that this vector makes with the world x-axis. Assuming that $|e_{\theta_k}| \neq (\pi/2)$, we have the following result.

Proposition III.4: The convergence results presented in Propositions III.2 and III.3 and Theorem III.2 hold for the nonholonomic dynamics as given in (17) if the following controller is applied:

$$\begin{aligned} v_k &= -k_p \cos(e_{\theta_k}) \|\tau_{kd}\| \\ \omega_k &= -K_\theta e_{\theta_k} \end{aligned} \tag{20}$$

with gains $k_p, K_\theta > 0$.

Proof: See [14]. ■

TABLE I
PARAMETERS USED IN EXPERIMENTS

Parameter	Exp 1	Exp 2	Exp 3	Exp 4
k_c	1.0	1.0	1.0	1.0
k_f	0.0	1.0	0.0	1.0
k_a	0.1	1.0	0.1	0.1
K_θ	5.0	5.0	5.0	5.0
$\bar{\alpha}$	20	1	1.0	1.0
$\underline{\alpha}$	0	0	0	0
ρ_1 [m]	0.7	0.7	0.7	0.7
ρ_2 [m]	2.3	2.3	2.3	2.3
R [m]	0.7	1.0	0.7	0.7
r [m]	0.4	0.45	0.4	0.4

IV. EXPERIMENTAL IMPLEMENTATION

The connectivity control is demonstrated using an experimental setup consisting of six iRobot Creates. The kinematics of the Creates are given by (17), where the inputs are the desired linear and angular velocities v_k, ω_k. Each robot has a Linux-based Gumstix Verdex microcontroller board for implementing low-level control, which we program in C++. Each control board possesses a wireless transceiver. The position feedback is obtained using a VICON motion tracking system. The VICON system has submillimeter accuracy with a data rate of 100 Hz. The control commands for each robot are computed using the VICON data on a desktop computer and communicated to the robots using wireless TCP/IP. The parameters for the control are given in Table I.

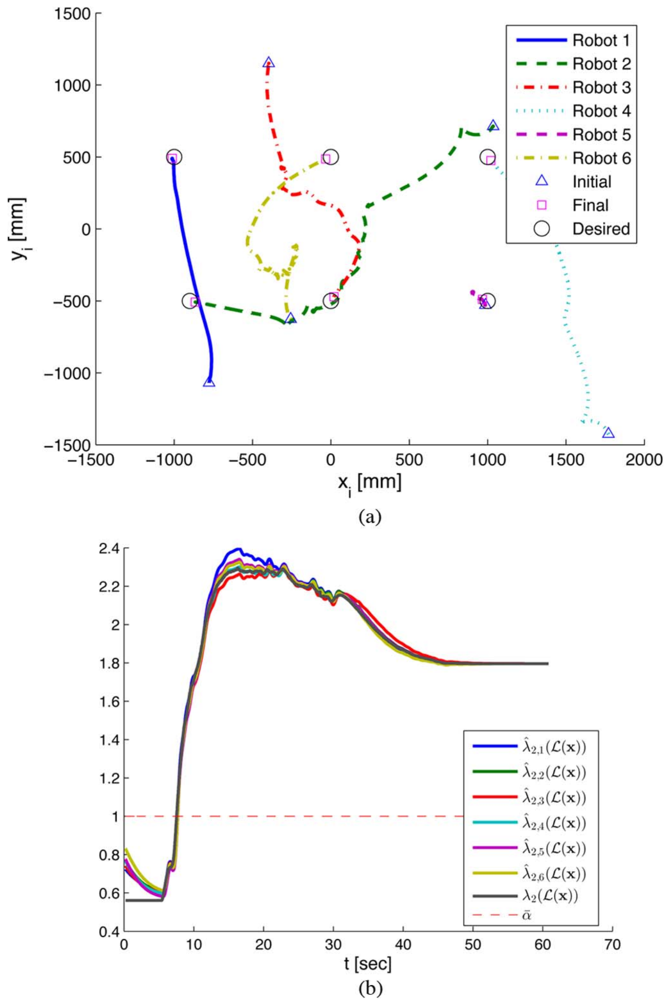

Fig. 2. Experiment with six Creates running the decentralized connectivity control, formation control, and collision avoidance. (a) Implicit plot of the six robots. (b) Estimates of $\lambda_2(\mathcal{L}(\mathbf{x}))$ and its true value.

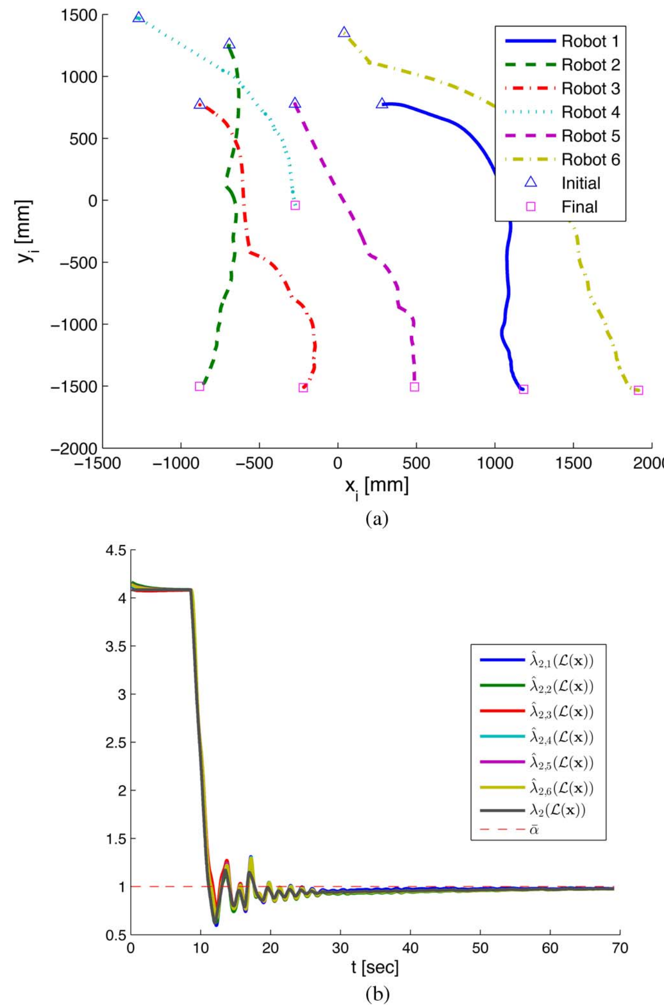

Fig. 3. Experiment with five Creates running the decentralized connectivity control, formation control, and collision avoidance. Robot 4 merely maintains connectivity and avoids collision. Maintaining connectivity results in it getting "dragged" by the other five robots as they move to their desired locations. (a) Implicit plot of the six robots. (b) Estimates of $\lambda_2(\mathcal{L}(\mathbf{x}))$ and its true value.

The controllers presented in Section III are implemented in experiments corresponding to different scenarios. When we refer to the controllers developed in Section III-B–D, we mean that they have been implemented using the procedure in Section III-E.

In the first experiment, each robot must achieve a desired position while avoiding other robots and maintaining connectivity. The control is of the form (16). We see in Fig. 2(a) that the steady-state position errors of the robots are small, and are a result of the dead zone in actuation. Thus, the agents have converged to their desired positions. Robots 2, 3, and 6 follow a circular path due to collision avoidance, since they are in each other's way. In Fig. 2(b), the initial estimates $\hat{\lambda}_2^k$ for each robot are close to the true value when the robots start moving. The estimates track the true value quite well. We see that the connectivity is allowed to decrease, and the final value of $\lambda_2(\mathcal{L})$ is less than the maximum possible value of 6.

In the second experiment, five robots use the same control as used in the first experiment. However, the fourth robot implements the decentralized connectivity control and the collision-avoidance control, but not the formation control, that is, $\tau_4 = \tau_{4,c} + \tau_{4,a}$. The remaining five robots are given desired positions with $y_i = -1500$ mm. In Fig. 3(a), we see that the steady-state position errors of these five robots are small. The initial algebraic connectivity of the robots is high, that is, $\lambda_2(\mathcal{L}) > 4$. At $t \approx 9$ s, the five robots move toward their desired locations, and away from robot 4. This causes a drop in connectivity; however, Robot 4 does not react until $\lambda_2(\mathcal{L}) < \bar{\alpha} = 1$, as seen in Fig. 4. The connectivity controller causes the formation to "drag" Robot 4 in order to maintain a large enough value of connectivity. The minimum value of $\lambda_2(\mathcal{L})$ is greater than 0.5 and the robots remain connected throughout the experiment.

V. Conclusion

In this paper, we have presented a decentralized connectivity control method for a mobile network based on maximization of the second smallest eigenvalue $\lambda_2(\mathcal{L})$ of the graph Laplacian $\mathcal{L}$. In practice, this is achieved by maximizing a local measure of connectivity given by the determinant of a matrix $\mathcal{M}_k = \mathcal{P}^T \mathcal{L}_k \mathcal{P}$, which eventually results in increasing $\lambda_2(\mathcal{L})$. We prove that the connectivity control maintains connectivity by

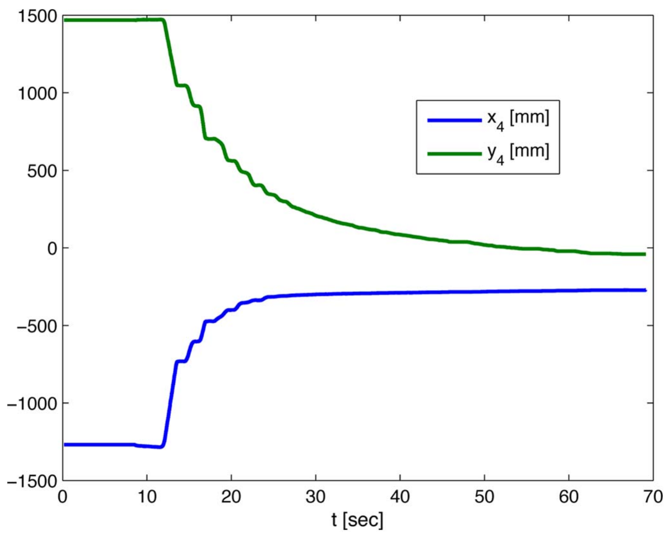

Fig. 4. Position of Robot 4. The robot moves only when $\lambda_2(\mathcal{L}(\mathbf{x})) < 1$.

increasing the connectivity away from zero whenever it is below a certain threshold. In addition, the connectivity control (6) can be integrated into a previous collision-avoiding formation controller [15] without losing the latter's convergence properties, provided the desired formation has a value of $\lambda_2(\mathcal{L})$ above the threshold used in our control.

The decentralized version of the connectivity control law relies on the estimator given in [12]. This controller is shown to behave similar to the centralized version in [11], except for the requirement of preventing the robots from moving until the error in the estimate of $\lambda_2(\mathcal{L})$ at the initial time is small.

Using the extension in [11], we can implement the decentralized controller on a team of nonholonomic-wheeled mobile robots. Experiments which demonstrate the properties of the controllers were provided. The experiments show the convergence of the mobile robots to desired positions in the formation while maintaining connectivity and avoiding collisions. This behavior is stronger than what Theorem III.2 promises and, thus, presents a future avenue of research. The effectiveness of the method in [12] is also validated during the experiment.

Appendix

Proof of Proposition III.3: The proof rests on the fact that when $\hat{\lambda}_2^k < \bar{\alpha}$, agent k maintains $\mathcal{N}_k$ by moving in a direction that reduces the distance to the farthest neighbor.

Let $k, j \in V$ be such that $w_{kj} > 0$ is small, and as $w_{kj} \to 0$, the graphs G_k and G_j become disconnected. The local graphs G_k and G_j, which contain the edge (k, j), must have small values of algebraic connectivity, which approach 0 as $w_{kj} \to 0$. In order to prevent disconnection of the graphs G_k and G_j, we must have $\tau_k \to c(\mathbf{x}_j - \mathbf{x}_k)$ as $w_{kj} \to 0$ for some $c > 0$. We assume that agent j must also behave in a similar manner. When this occurs, d_{kj} must decrease when w_{kj} is sufficiently small, increasing w_{kj} away from zero.

Without loss of generality, we assume that $\pi_k(k) = 1$ and $\pi_k(j) = 2$ (see Section II), and refer to the edge weight w_{kj} as w_{12}. The proof of Lemma III.1 shows that the control effort $\tau_{k,c}$ has the form

$$\tau_{k,c} = -\beta_k \sum_{j=2}^{N_k+1} \gamma_k^{1j} \begin{bmatrix} 2\frac{\partial w_{1j}}{\partial x_k} \\ 2\frac{\partial w_{1j}}{\partial y_k} \end{bmatrix} \tag{21}$$

where the weights $\gamma_k^{i'j'}$ are given by

$$\gamma_k^{i'j'} = \sum_{p=2}^{N_k+1} \frac{1}{\lambda_p(\mathcal{L}_k)} \left(v_p^T v^{i'j'}\right)^2 \tag{22}$$

where $i', j' \in V_k$, $\mathcal{L}_k v_p = \lambda_p v_p$, and $v^{i'j'} \in \mathbb{R}^{(N_k+1)}$. By assumption, $\lambda_2(\mathcal{L}_k) \to 0$. Since $\lambda_2(\mathcal{L}_k) < \lambda_3(\mathcal{L}_k) < \ldots < \lambda_{N_k+1}(\mathcal{L}_k)$, the term due to $p = 2$ dominates, and we can rewrite (22) as

$$\gamma_k^{i'j'} \approx \frac{1}{\lambda_2(\mathcal{L}_k)} \left(v_2^T v^{i'j'}\right)^2. \tag{23}$$

As $w_{12} \to 0$, $\lambda_2(\mathcal{L}_k) \to 0$ by assumption, so that the multiplicity of the 0 eigenvalue is 2. The nullspace of $\mathcal{L}_k$ is 2-D and is spanned by $\mathbf{1}$ and e_2, where $e_i \in \mathbb{R}^{(N_k+1)}$ is the standard basis vector with the ith element equal to unity. To see that e_2 lies in the nullspace, note that as $w_{12} \to 0$, the second column of $\mathcal{L}_k$ approaches the zero vector. Thus, as $w_{12} \to 0$, $v_2 \to v$, where $v = c_1\mathbf{1} + c_2 e_2$, $c_2 \neq 0$. Thus, the expression $v_2^T v^{1j} \to v^T v^{1j}$, where the latter can easily be computed as

$$v^T v^{1j} = \begin{cases} \frac{c_2}{\sqrt{2}}, & \text{if } j = 2 \\ 0, & \text{otherwise.} \end{cases} \tag{24}$$

Thus, as $w_{12} \to 0$, we can substitute (23) in (21) to obtain

$$\tau_{k,c} \to -\beta_k \sum_{j=2}^{N_k+1} \frac{1}{\lambda_2} \left(v_2^T v^{1j}\right)^2 \begin{bmatrix} 2\frac{\partial w_{1j}}{\partial x_k} \\ 2\frac{\partial w_{1j}}{\partial y_k} \end{bmatrix}. \tag{25}$$

Using (24) in the expression above, we obtain

$$\begin{aligned} \tau_{k,c} \to &- \beta_k \frac{1}{\lambda_2} \left(v^T v^{12}\right)^2 \begin{bmatrix} 2\frac{\partial w_{12}}{\partial x_k} \\ 2\frac{\partial w_{12}}{\partial y_k} \end{bmatrix} \\ = &- \beta_k \frac{1}{\lambda_2} c_2^2 \delta_{12} \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix} \end{aligned} \tag{26}$$

which is clearly of the form $c(\mathbf{x}_2 - \mathbf{x}_1)$, which is $c(\mathbf{x}_j - \mathbf{x}_k)$ when referring to the agents as members of V. By a similar argument, $\tau_j \to c'(\mathbf{x}_k - \mathbf{x}_j)$ for some $c' > 0$. Note that as $\lambda_2(\mathcal{L}_k)$ and $\lambda_2(\mathcal{L}_j)$ approach zero, the variables c and c' increase without bound, so that $\|\tau_{k,c}\| \gg \|\tau_{k,f}\|$ and $\|\tau_{j,c}\| \gg \|\tau_{j,f}\|$. Since $|\hat{\lambda}_2^k - \lambda_2(\mathcal{L})| < \bar{\alpha}$, when $\lambda_2(\mathcal{L})$ is sufficiently small, even in the presence of estimation errors, $\beta_k \neq 0$. Thus, even though w_{kj} can become very small, it cannot decrease until zero, since the two agents at opposite ends of such an edge will eventually move toward each other. This shows that any graph G_k remains connected when $\hat{\lambda}_2^k < \bar{\alpha}$, so that all edges in the connected graph G are preserved. This implies that the graph G cannot become disconnected when the agents move according to control law (7a). Note that this argument implies that there is some $\tilde{\epsilon} > 0$ such that $\lambda_2(\mathcal{L}_k)(t) \geq \tilde{\epsilon}\ \forall\ t \geq t_0, k \in V$. Thus, $(1/\lambda_2(\mathcal{L}_k)) \leq (1/\tilde{\epsilon})$. Due to the form of (26), we can conclude that $\tau_{k,c}$ has an upper bound. ■

REFERENCES

[1] M. Zavlanos, M. Egerstedt, and G. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proc. IEEE*, vol. 99, no. 9, pp. 1525–1540, Sep. 2011.

[2] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 693–703, Aug. 2007.

[3] M. Zavlanos and G. Pappas, "Controlling connectivity of dynamic graphs," in *Proc. 44th IEEE Conf. Decis. Control, Eur. Control Conf.*, Dec. 2005, pp. 6388–6393.

[4] H. Yu and P. Antsaklis, "Formation control of multi-agent systems with connectivity preservation by using both event-driven and time-driven communication," in *Proc. IEEE 51st Annu. Conf. Decis. Control*, 2012, pp. 7218–7223.

[5] Z. Kan, A. Dani, J. Shea, and W. Dixon, "Network connectivity preserving formation stabilization and obstacle avoidance via a decentralized controller," *IEEE Trans. Autom. Control*, vol. 57, no. 7, pp. 1827–1832, Jul. 2012.

[6] D. Dimarogonas and K. Johansson, "Bounded control of network connectivity in multi-agent systems," *IET Control Theory Appl.*, vol. 4, no. 8, pp. 1330–1338, 2010.

[7] M. Zavlanos and G. Pappas, "Distributed connectivity control of mobile networks," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1416–1428, Dec. 2008.

[8] A. Ajorlou, A. Momeni, and A. Aghdam, "A class of bounded distributed control strategies for connectivity preservation in multi-agent systems," *IEEE Trans. Autom. Control*, vol. 55, no. 12, pp. 2828–2833, Dec. 2010.

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

[10] 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.

[11] A. C. Satici, H. A. Poonawala, H. Eckert, and M. W. Spong, "Connectivity preserving formation control with collision avoidance for nonholonomic wheeled mobile robots," in *Proc. Int. Conf. Intell. Robots Syst.*, 2013, pp. 5080–5086.

[12] P. Yang, R. Freeman, G. Gordon, K. Lynch, S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity in mobile sensor networks," in *Proc. Amer. Control Conf.*, Jun. 2008, pp. 2678–2683.

[13] M. Zavlanos and G. Pappas, "Potential fields for maintaining connectivity of mobile networks," *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 812–816, Aug. 2007.

[14] H. Poonawla, A. Satici, H. Eckert, and M. Spong, "Collision-free formation control with decentralized connectivity preservation for nonholonomic wheeled mobile robots." [Online]. Available: http://www.utdallas.edu/~mspong/reports/

[15] S. Mastellone, D. Stipanovic, and M. Spong, "Remote formation control and collision avoidance for multi-agent nonholonomic systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 1062–1067.

Hasan A. Poonawala (M'14) received the B.Tech. degree in mechanical engineering from the National Institute of Technology, Surathkal, India, in 2007, the M.S. degree in mechanical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2009, and the Ph.D. degree in electrical engineering from the University of Texas, Dallas, TX, USA, in 2014.

Currently, he is a Postdoctoral Researcher at the University of Texas at Dallas. His current research interests include robotics, multiagent systems, and nonlinear control theory.

Aykut C. Satici received the B.Sc. and M.Sc. degrees in mechatronics engineering from Sabanci University, Istanbul, Turkey, in 2008 and 2010, respectively. He received the M.Sc. degree in mathematics, in 2013, and the Ph.D. degree in electrical engineering, in 2014, from University of Texas at Dallas, Dallas, TX, USA.

His research interests include dynamics, robotics, geometric mechanics and control, and cooperative control.

Hazen Eckert is currently an undergraduate senior studying computer science, computer engineering, and economics at the University of Texas at Dallas, Dallas, TX, USA.

His research interests include robotics, big data analytics, and machine learning.

Mark W. Spong (F'96) received the B.A. degree (Hons.) in mathematics and physics from Hiram College, Hiram, OH, USA, in 1975, the M.S. degree in mathematics from New Mexico State University, Las Cruces, NM, USA, in 1977, and the M.S. and D.Sc. degrees in systems science and mathematics from Washington University, St. Louis, MO, USA, in 1979 and 1981, respectively.

He was with the University of Illinois at Urbana-Champaign, Urbana, IL, USA, from 1984 to 2008. He is currently the Dean of the Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, Dallas, TX, USA, where he is a Professor of Electrical Engineering and holds both the Lars Magnus Ericsson Chair and the Excellence in Education Chair. He has authored or co-authored more than 250 technical papers in control and robotics and four books, and holds one patent. His current research interests include robotics, mechatronics, and nonlinear control theory.

Dr. Spong was the President of the IEEE Control Systems Society and was both the Editor-in-Chief and an Associate Editor of the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY and an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, the *IEEE Control Systems Magazine*, and the IEEE TRANSACTIONS ON AUTOMATIC CONTROL. He was the Vice President for Publication Activities from 2000 to 2002. He was the recipient of several awards, including the first Intelligent Robots and Systems Fumio Harashima Award for Innovative Technologies, the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY Outstanding Paper Award, the Senior Scientist Research Award from the Alexander von Humboldt Foundation, the Distinguished Member Award from the IEEE Control Systems Society, the John R. Ragazzini and O. Hugo Schuck Awards from the American Automatic Control Council, and the IEEE Third Millennium Medal.