Decentralized Connectivity Preserving Formation Control with Collision Avoidance for Nonholonomic Wheeled Mobile Robots

Hasan Poonawala, Aykut C Satici, Hazen Eckert and Mark W Spong

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 achieve this involves each agent preserving its edges 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 result in a monotonic increase in connectivity until the network is completely connected. A continuous feedback control method was proposed which allows the connectivity to decrease, that is, edges in the network may be broken. This method requires global knowledge of the 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 non-holonomic wheeled mobile robots, and the performance is demonstrated in experiment on a team of wheeled mobile robots.

I. Introduction

The study of mobile-robot networks has been an active area of research for over a decade. Such systems 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 large numbers of robots. An obvious method of maintaining connectivity is to preserve the edges present in the network for all time [2], [3]. Most decentralized methods to preserve connectivity utilize a variation of this idea. A notable exception

Hasan Poonawala, Aykut C Satici and Hazen Eckert are with the Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, Richardson, Texas, USA {acsatici,hasanp,hxe110130}@utdallas.edu

Mark W Spong is the Dean of the Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, Richardson, Texas, USA mspong@utdallas.edu

This research was partially supported by the National Science Foundation Grants ECCS 07-25433 and CMMI-0856368, by the University of Texas STARs program, and by DGIST R&D Program of the Ministry of Education, Science and Technology of Korea(12-BD-0101).

is found in [4], where the authors propose algorithms to decide if edges may be deleted while still ensuring 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 using a hysterisis protocol. A method that uses bounded control to tackle this phenomenon is given in [5], 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 $\mathcal{L}$. Thus, another method for maintaining connectivity amongst a group of mobile robots is to maximize the second smallest eigenvalue of the graph Laplacian [6]. In this method, the edge strengths are non-increasing functions of the distance between robots. The resulting graph is always completely connected, as seen in the simulation results in [6]. This method is effective for solving rendezvous problems, which was a primary goal in [6]. It can also can be extended to some other applications [1], [6] 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 [7] we modify the approach in [6] so that the resulting behavior is such that global connectivity is maintained, but not increased till 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 [4], however our controller requires each agent to have access to the positions of all other nodes.

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 [8]. A further contribution in this work is the experimental validation of this decentralized controller on a network of non-holonomic wheeled mobile robots (WMRs). This is also the first reported implementation of the algorithm in [8] using experimental data.

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.

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 $V = \{1, ..., N\}$ 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

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

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:

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

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, as seen in Figure III.1. 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 $\mathcal{L}(G)$. Thus, for the graph to be connected, only one eigenvalue of $\mathcal{L}(G)$ will be zero. The second smallest eigenvalue $\lambda_2(\mathcal{L}(G))$ thus becomes an indicator of connectivity in the graph.

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 $\mathcal{L}(G)$. The matrix $\mathcal{M}(G)$ is given by

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

where $\mathcal{P} \in \mathbb{R}^{N \times N-1}$ satisfies $\mathcal{P}^T \mathbf{1} = 0$ and $\mathcal{P}^T \mathcal{P} = I_{N-1}$. Thus, the determinant of $\mathcal{M}(G)$ vanishes if and only if $\lambda_2(L)(G)$ vanishes.

For each $k \in V$ we can define the neighbor set $\mathcal{N}_k$ as

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

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

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

Note that $\bar{\mathcal{N}_k} \subseteq V$ has order $N_k + 1$, where N_k is the number of neighbors of k in G. Each member of $\bar{\mathcal{N}_k}$ can be assigned a position in $V_k = \{1, 2, ..., N_k + 1\}$. 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 $\mathcal{M}_k = \mathcal{M}(G_k)$.

For the rest of the 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 applying to $\mathcal{M}$.

III. MOTIVATION

Consider the scenario where we would like a team of robots to stay connected with 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.

One way to attack this problem is to come up with a connectivty controller based on maximization of $\lambda_2(\mathcal{L})$ and add another controller that achieves the tracking aspect of the task. The downfall of this approach is that the connectivity may conflict with the ability of the tracking controller to achieve the desired goal, or restrict the set of robot positions that can be tracked.

Example. Consider $N = 5$ first-order robots, whose dynamics are represented by the simple integrator, with a maximum detection range of $\rho_2 = 0.70$m. Assume that two robots can detect each other perfectly if they are a distance of $\rho_1 = 0.20$m. or smaller away. We model this connectivity pattern with the weight function defined in (II.1).

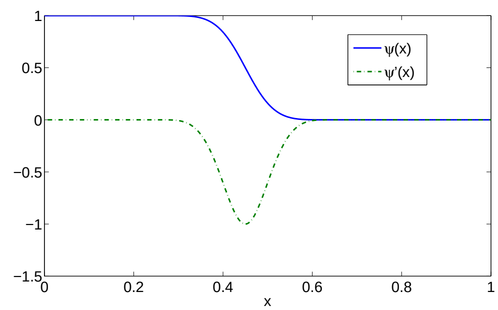


Fig. III.1: Bump function

Suppose a second smallest eigenvalue-maximizing control law such as the one given in [1] is applied on each of the robots; that is, the control law for the k^{th} robot will be the gradient of the potential function

$$\phi(\mathbf{x}) = \log\det(\mathcal{P}^T\mathcal{L}(\mathbf{x})\mathcal{P})^{-1} := \log\det(\mathcal{M}(\mathbf{x}))^{-1}$$

where $\mathbf{x}_k$ denotes the position vector (x_k, y_k) of robot k. The controller for the k^{th} robot reads

$$\tau_k = -\frac{\partial\phi}{\partial x_k}(\mathbf{x}) = \mathrm{tr}\left(\mathcal{M}^{-1}\frac{\partial\mathcal{M}}{\partial x_k}(\mathbf{x})\right)$$

We simulate this control law with the first robot commanded to remain at the position $x_1 = (0.5, 0.5)$. This is equivalent to the robot tracking any constant set point. The behaviors of the remaining robots are illustrated in Figure IV.1. The bold green curve is the circle of radius ρ_1 around the location of the first robot. We immediately notice that all of the robots are forced into this circle because of the second smallest eigenvalue maximizing control law. This is because the connection strengths are maximized when each pair of robots is separated by no more than ρ_1.

This means that formations to be tracked should lie completely inside this circle. This could be a serious limitation, for example in tasks related to coverage. Thus, we see that connectivity control based on second smallest eigenvalue maximization alone limits the success of achieving additional behaviors.

IV. CONTROL DESIGN

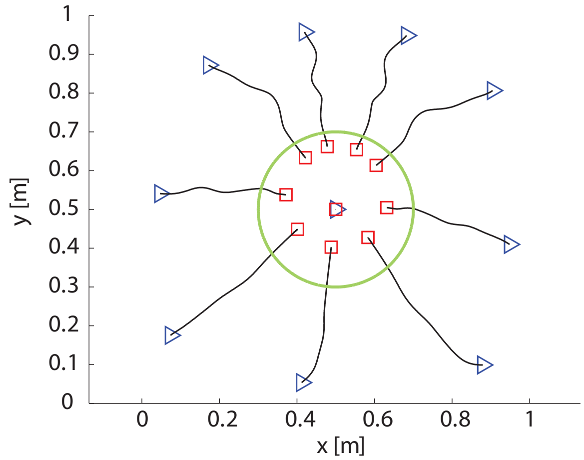


Fig. IV.1: $\lambda_2(\mathcal{L})$ maximizing controller

A connectivity preserving controller with the desired properties mentioned in Section III was presented in [7]. 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 IV-B. We also analyse the performance of the decentralized connectivity controller when used in conjunction with formation or collision-avoidance control laws in Sections IV-C and IV-D respectively.

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

$$\dot{\mathbf{x}}_i = \tau_i \tag{IV.1}$$

where $\mathbf{x}_i \in \mathbb{R}^2$ is the position of the i^{th} mobile robot given by $\mathbf{x}_i = (x_i, y_i)$. In Section IV-E In the next subsection, we recount the centralized control law from [7] and its properties.

A. Centralized connectivity control law

The connectivity control law in [] 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 [7], the following potential function was proposed:

$$D(\mathbf{x}) := \det(\mathcal{M}(\mathbf{x}))$$

$$V_c(D) = \left(\min\left\{0, \frac{D^2 - \bar{\alpha}^2}{D^2 - \underline{\alpha}^2}\right\}\right)^2$$

This function and its gradient blow up whenever the determinant approaches the lower bound $\underline{\alpha}$ and are zero whenever the determinant is greater than the upper bound $\bar{\alpha}$. Thus, using the above potential function 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 non-zero lower bound which we can select. The performance of consensus based algorithms improves with 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 lowers 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.

Upon taking partial derivatives of V_c with respect to the coordinates x_k and y_k of the k^{th} robot, we find

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

where

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

The following result was shown in [7].

Proposition IV.1. *Under the control law*

$$\tau_k = -\nabla_{\mathbf{x}_k} V_c(\mathbf{x}) = -\beta(\mathbf{x}) \begin{bmatrix} \text{tr}\left(\mathcal{M}^{-1}\frac{\partial \mathcal{M}}{\partial x_k}\right) \\ \text{tr}\left(\mathcal{M}^{-1}\frac{\partial \mathcal{M}}{\partial y_k}\right) \end{bmatrix} \tag{IV.2}$$

the first-order robots with dynamics (IV.1) *converges to the set* $E = \{\mathbf{x} \in \mathbb{R}^{2N} : \det(\mathcal{M}(\mathbf{x})) \geq \bar{\alpha}\}$ *and the graph* G *whose nodes the robots represent stays connected for all time.*

B. Decentralized Connectivity Control law

The connectivity controller (IV.2) can be viewed as the control given in [1] multiplied by a gain which is dependent on the connectivity of the network. The controller is centralized due to the need to compute the matrix $\mathcal{M}^{-1}(\mathbf{x})$ and the determinant of $\mathcal{M}$. The terms $\frac{\partial \mathcal{M}}{\partial \mathbf{x}_k}$ depend on the terms $\frac{\partial w_{ij}}{\partial \mathbf{x}_k}$, which vanish for robots that are not neighbors. Thus, $\frac{\partial \mathcal{M}}{\partial \mathbf{x}_k}$ is a local computation for each robot.

It was shown that the velocity vector τ_k in (IV.2) was a positive combination of the vectors from that robot to each of its neighbors. The direction of τ_k was determined by the gradient of $\det \mathcal{M}$, the calculation of which requires each agent to possess global information. Each robot can be commanded to move in the direction determined by the gradient of $\det \mathcal{M}_k$ instead, which requires each neighbor to have information about its neighbors only. Thus, the decentralized connectivity controller becomes

$$\tau_{k,c} = -k_c\beta_2(\mathbf{x}) \begin{bmatrix} \text{tr}\left(\mathcal{M}_k^{-1}\frac{\partial \mathcal{M}_k}{\partial x_k}\right) \\ \text{tr}\left(\mathcal{M}_k^{-1}\frac{\partial \mathcal{M}_k}{\partial y_k}\right) \end{bmatrix} \tag{IV.3a}$$

$$\beta_2(\mathbf{x}) = \min\left\{\frac{\left(\lambda_2^2(\mathcal{L}) - \bar{\alpha}^2\right)}{\left(\lambda_2^2(\mathcal{L})\right)}, 0\right\} \tag{IV.3b}$$

where $k_c > 0$ and $\beta_2(\mathbf{x}) \leq 0 \ \forall \mathbf{x} \in \mathbb{R}^{2N}$. Note that if $\lambda_2(\mathcal{L})$ is available at each node, then the control law requires information about its neighbors. This means that the control law is decentralized. In order to understand the behaviour of the agents under the action of (IV.3a), we show the following property,

Proposition IV.2. *The instantaneous direction of motion of each robot $k \in V$ under any control of the form*

$$\tau_k = -\beta \begin{bmatrix} \mathrm{tr}\left(\mathcal{M}_k^{-1}\frac{\partial \mathcal{M}_k}{\partial x_k}\right) \\ \mathrm{tr}\left(\mathcal{M}_k^{-1}\frac{\partial \mathcal{M}_k}{\partial y_k}\right) \end{bmatrix} \tag{IV.4}$$

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

Proof: Each agent has a subgraph $G_k = (V_k, W)$ as defined in Section II. We define the the symmetric matrix $A_k^{ij} \in \mathbb{R}^{(N_k+1)\times(N_k+1)}$ as

$$A_k^{ij}(n,m) \;= A_k^{ij}(m,n) = \begin{cases} w_{ij} & \text{if } m = i \text{ and } n = j \\ 0 & \text{otherwise} \end{cases}$$

where w_{ij} is defined as in (II.1). This corresponds to the adjacency matrix of a subgraph of G_k consisting of the same $(N_k + 1)$ robots, but only robots i and j are connected by weight w_{ij}. We can construct a matrix L_k^{ij} from A_k^{ij} using the standard process of obtaining a Laplacian matrix from an adjacency matrix. This matrix has the property that

$$L_k^{ij} = 2w_{ij}v^{ij}(v^{ij})^T \tag{IV.5}$$

where $v^{ij} \in \mathbb{R}^{(N_k+1)}$ with its l^{th} component given by

$$v_l^{ij} = \begin{cases} -1/\sqrt{2} & \text{if } l = i \\ 1/\sqrt{2} & \text{if } l = j \\ 0 & \text{otherwise} \end{cases} \tag{IV.6}$$

It should be noted that we recover the original graph Laplacian, $\mathcal{L}_k$, by the expression

$$\mathcal{L}_k = \sum_{i=1}^{N}\sum_{j>i}^{N} L_k^{ij} \tag{IV.7}$$

The partial derivatives of L_k^{ij} can be expressed as

$$\frac{\partial L_k^{ij}}{\partial x_l} = 2\frac{\partial w_{ij}}{\partial x_l}v^{ij}(v^{ij})^T$$
$$\frac{\partial L_k^{ij}}{\partial y_l} = 2\frac{\partial w_{ij}}{\partial y_l}v^{ij}(v^{ij})^T$$

We take the inverse $\mathcal{M}_k^{-1} > 0$ of the matrix $\mathcal{M}_k > 0$ obtained from $\mathcal{L}_k$ by using (II.2) and express it in terms of the eigenvalue decomposition of $\mathcal{L}_k$:

$$\mathcal{M}_k^{-1} = \sum_{p=2}^{N_k+1}\frac{1}{\lambda_p}u_p u_p^T$$

where $\mathcal{L}_k v_p = \lambda_p(\mathcal{L}_k)v_p$, $u_p = P^T v_p$ for each $p \in \{1, 2, 3, ..., N_k + 1\}$. We calculate, for some l,

$$\begin{aligned}
\mathrm{tr}\left(\mathcal{M}_k^{-1}P^T\frac{\partial L^{ij}}{\partial x_l}P\right) &= \mathrm{tr}\left(\sum_{p=2}^{N_k+1}\frac{1}{\lambda_p}u_pu_p^T 2\frac{\partial w_{ij}}{\partial x_l}P^Tv^{ij}(v^{ij})^TP\right)\\
&= 2\frac{\partial w_{ij}}{\partial x_l}\mathrm{tr}\left(\sum_{p=2}^{N_k+1}\frac{1}{\lambda_p}u_pu_p^TP^Tv^{ij}(P^Tv^{ij})^T\right)\\
&= 2\frac{\partial w_{ij}}{\partial x_l}\sum_{p=2}^{N_k+1}\frac{1}{\lambda_p}(u_p^TP^Tv^{ij})^2 = 2\gamma_k^{ij}\frac{\partial w_{ij}}{\partial x_l}
\end{aligned} \tag{IV.8}$$

and similarly,

$$\mathrm{tr}\left(\mathcal{M}_k^{-1}P^T\frac{\partial L^{ij}}{\partial x_l}P\right) = 2\gamma_k^{ij}\frac{\partial w_{ij}}{\partial y_l}$$

where $\gamma_{ij} > 0$. Now, take the vector

$$\tau_{k,l}^{ij} = \begin{bmatrix}\mathrm{tr}\left(\mathcal{M}_k^{-1}P^T\frac{\partial L^{ij}}{\partial x_l}P\right)\\ \mathrm{tr}\left(\mathcal{M}_k^{-1}P^T\frac{\partial L^{ij}}{\partial y_l}P\right)\end{bmatrix} = 2\gamma_k^{ij}\begin{bmatrix}\frac{\partial w_{ij}}{\partial x_l}\\ \frac{\partial w_{ij}}{\partial y_l}\end{bmatrix}$$

We have the relations

$$\begin{aligned}
\frac{\partial w_{ij}}{\partial x_i} &= \frac{\partial w_{ij}}{\partial d_{ij}}\frac{\partial d_{ij}}{\partial x_i} = \frac{\partial w_{ij}}{\partial d_{ij}}\frac{x_i - x_j}{d_{ij}}\\
\frac{\partial w_{ij}}{\partial y_i} &= \frac{\partial w_{ij}}{\partial d_{ij}}\frac{\partial d_{ij}}{\partial y_i} = \frac{\partial w_{ij}}{\partial d_{ij}}\frac{y_i - y_j}{d_{ij}}
\end{aligned}$$

thus

$$\begin{bmatrix}\frac{\partial w_{ij}}{\partial x_i}\\ \frac{\partial w_{ij}}{\partial y_i}\end{bmatrix} = \delta_{ij}\begin{bmatrix}x_j - x_i\\ y_j - y_i\end{bmatrix}$$

where $\delta_{ij} \geq 0$, since $\frac{\partial w_{ij}}{\partial d_{ij}} \leq 0$. To compute the control for l^{th} robot, we use the fact that

$$\frac{\partial \mathcal{L}_k}{\partial x_l} = \sum_{i=1}^{N}\sum_{j>i}^{N}\frac{\partial L_k^{ij}}{\partial x_l} = \sum_{j\neq l}^{N}\frac{\partial L_k^{lj}}{\partial x_l}$$

since $L_k^{ij} = L_k^{ji}$ and $\frac{\partial L_k^{ij}}{\partial x_l} = 0$ whenever neither $l \neq i$, nor $l \neq j$. Therefore, the control vector (IV.4) is computed as

$$\tau_k = -\beta\begin{bmatrix}\mathrm{tr}\left(\mathcal{M}_k^{-1}P^T\frac{\partial \mathcal{L}_k}{\partial x_k}P\right)\\ \mathrm{tr}\left(\mathcal{M}_k^{-1}P^T\frac{\partial \mathcal{L}_k}{\partial y_k}P\right)\end{bmatrix} = -\beta(\mathbf{x})\sum_{j\neq k}^{N_k+1}\begin{bmatrix}\mathrm{tr}\left(\mathcal{M}_k^{-1}P^T\frac{\partial L^{kj}}{\partial x_k}P\right)\\ \mathrm{tr}\left(\mathcal{M}_k^{-1}P^T\frac{\partial L^{kj}}{\partial y_k}P\right)\end{bmatrix} = -\beta(\mathbf{x})\sum_{j\neq k}^{N_k+1}2\gamma_{kj}\delta_{kj}\begin{bmatrix}x_j - x_k\\ y_j - y_k\end{bmatrix}$$

which is clearly a positive combination of the displacement vectors from robot k to the robots $j \in \mathcal{N}_k$ (note, $\beta(\mathbf{x}) \leq 0$).

■

Proposition IV.3. *Consider the control law for the k^{th} robot given by* (IV.3)*. If the robots are started such that $\lambda_2(\mathcal{L})(t_0) > 0$, then $\lambda_2(\mathcal{L})(t) > 0$ $\forall t \geq t_0$, and the configuration of the agents converges to the set $E = \left\{\mathbf{x} \in \mathbb{R}^{2N} : \lambda_2(\mathcal{L}) \geq \bar{\alpha}\right\}$*

Proof: Agent $k \in V$ has a local graph $G_k = (V_k, W)$. Let the graph G_k be such that $w_{kj} = \epsilon > 0$ and as $w_{kj} \to 0$ the graph becomes disconnected. In order to prevent disconnection, we must have that $\tau_k \to c(\mathbf{x}_j - \mathbf{x}_k)$ as $w_{1j} \to 0$ for some $c > 0$. We assume that the opposite node of this edge also behaves in a similar manner. When this occurs, d_{kj} must decrease when w_{kj} is sufficiently small, increasing w_{jk} away from zero.

Note that agents k and j have indices $k' = \pi_k(k)$ and $j' = \pi_k(j)$ respectively in V_k (See section II). The control law for agent k was shown to be a positive combination of the relative position vectors to its neighbors. From (IV.8) the weights are given by

$$\gamma_k^{k'j'} = \sum_{p=2}^{N_k+1} \frac{1}{\lambda_p} (u_p^T P^T v^{k'j'})^2 \tag{IV.9}$$

If $w_{kj} \to 0$ then $\lambda_2(\mathcal{L}_k) \to 0$. Since $\lambda_2 < \lambda_3 < \dots \lambda_{N_k+1}$, the term due to $p = 2$ dominates, and we can rewrite above as

$$\gamma_k^{k'j'} \approx \frac{1}{\lambda_2} (u_2^T P^T v^{k'j'})^2 \tag{IV.10}$$

We have that

$$\mathcal{L}_k = \sum_{i'=1}^{N_k+1} \sum_{j'>i'}^{N_k+1} 2w_{i'j'} v^{i'j'} (v^{i'j'})^T$$

so that

$$\mathcal{L}_k v_2 = \lambda_2(\mathcal{L}_k) v_2 = \sum_{i'=1}^{N} \sum_{j'>i'}^{N} 2w_{i'j'} v^{i'j'} (v^{i'j'})^T v_2$$

As $\lambda_2(\mathcal{L}_k) \to 0$, we get

$$\sum_{i'=1}^{N} \sum_{j'>i'}^{N} 2w_{i'j'} v^{i'j'} (v^{i'j'})^T u_2 = \sum_{i'=1}^{N} \sum_{j'>i'}^{N} 2w_{i'j'} ((v^{i'j'})^T u_2) v^{i'j'} \to 0 \tag{IV.11}$$

The weights $w_{i'j'} = w_{ij}$ are non-negative. Due to the form of (IV.10), we are concerned with the behavior of $v^{k'j'}, j' \in \{1, 2, 3, ..., N_k + 1\}$ $\{k'\}$. By definition, the k' $^{\text{th}}$ component of each $v^{k'j'}$ is $-1/\sqrt{2}$. Thus, the only way (IV.11) can hold for the first component is that either $w_{k'j'} \to 0$ or $(v^{k'j'})^T v_2 \to 0 \Rightarrow u_2^T P^T v^{k'j'} \to 0$. Thus, we can conclude that

$$\tau_{k,c} \to -\beta(\mathbf{x}) \sum_{j | w_{k'j'} \to 0}^{N_k+1} \frac{1}{\lambda_2} (u_2^T P^T v^{k'j'})^2 \begin{bmatrix} 2\frac{\partial w_{k'j'}}{\partial x_k} \\ 2\frac{\partial w_{k'j'}}{\partial y_k} \end{bmatrix} \tag{IV.12}$$

If only one edge is close to vanishing, then

$$\tau_{k,c} \to -\beta(\mathbf{x}) \frac{1}{\lambda_2} (u_2^T P^T v^{k'j'})^2 \begin{bmatrix} 2\frac{\partial w_{k'j'}}{\partial x_k} \\ 2\frac{\partial w_{k'j'}}{\partial y_k} \end{bmatrix} \tag{IV.13}$$

which is clearly of the form $c(\mathbf{x}_j - \mathbf{x}_k)$ (note that we revert to addressing nodes by their position in V). Thus, even though w_{kj} can become very small, it cannot decrease till zero, since the two agents at opposite ends of such an edge will eventually move towards each other. This shows that the graph G remains connected for all time when each agent moves under the action of (IV.3a).

By Proposition IV.2, each agent moves towards the interior of the convex hull CH(V_k) determined by its neighbor set.

Moreover, the agents defining the convex hull CH(V) of the whole graph will move into the 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 Bolzano-Weierstrass Theorem, each entry of the vector v is approaching 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 $\epsilon > 0$. Since $\lambda_2(\mathcal{L})$ is a monotonically decreasing function of each distance d_{ij}, λ_2 will increase until $\beta_2(\mathbf{x}) \equiv 0$. ■

One of the nice features of the controller (IV.3a) is that its computation requires only local knowledge if $\lambda_2(\mathcal{L})$ is known. The authors in [8] have introduced an estimator for this critical piece of information with tunable gains that govern its rate of convergence. Once this estimator is combined with the control law (IV.3a), we achieve a decentralized connectivity controller.

The only restriction that the decentralization imposes is that the motions of the robots be slower than the rate of convergence of the estimator. Conversely, the estimator gains should be selected judiciously so that a time-scale separation between the convergence of the estimator states and the robot states is established.

C. *Decentralized Connectivity Preserving Formation Controller*

In this section, we develop on the connectivity controller presented in Section IV-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 $\mathbf{x}_{kd}$. The sum of the contributions of each robot gives rise to the formation potential function, $V_f(\mathbf{x})$.

$$V_{fk} = \frac{1}{2}\langle \mathbf{x}_k - \mathbf{x}_{kd}, \mathbf{x}_k - \mathbf{x}_{kd} \rangle$$

$$V_f = \sum_i^N V_{fk}$$

where the brackets $\langle \cdot, \cdot \rangle$ represents the usual Euclidean inner product of vectors. For convenience, we define

$$\mathcal{N}_k \ = \{j \in \{1, ...N\} : \ j \neq k \text{ and } d_{jk} < \rho_2\}$$

which is the index set of neighbors of robot k. We now show an important property of the connectivity control law.

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

$$\tau_k = \tau_{k,c}(\mathbf{x}) - k_f \nabla_{\mathbf{x}_i} V_f(\mathbf{x}) \tag{IV.14}$$

where $\tau_{k,c}(\mathbf{x})$ is the decentralized connectivity controller presented in section IV-B, and $k_c, k_f > 0$ are control gains.

Theorem IV.1. *Suppose the control effort for each robot k is given by* (IV.14). *Let V_d denote the vertex set for the desired formation. Then the robots converge to a set E contained in the convex hull* CH(V_d) *of the desired formation.*

Proof: Under the action of the control law (IV.14), the agents reach the set of configurations $\mathbf{x} \in \mathbb{R}^{2N}$ where for all k, both of the following holds

$$\begin{aligned} k_c \beta_2(\mathbf{x}) \mathrm{tr}\left(\mathcal{M}_k^{-1} \frac{\partial \mathcal{M}_k}{\partial x_k}\right) + k_f\left(x_k - x_{kd}\right) &= 0 \\ k_c \beta_2(\mathbf{x}) \mathrm{tr}\left(\mathcal{M}_k^{-1} \frac{\partial \mathcal{M}_k}{\partial y_k}\right) + k_f\left(y_k - y_{kd}\right) &= 0 \end{aligned}$$

Since $\beta(\mathbf{x}) < 0$, this is equivalent to the statement that the angle between the vectors $\left[\mathrm{tr}\left(\mathcal{M}_k^{-1} \frac{\partial \mathcal{M}_k}{\partial x_k}\right) \ \ \mathrm{tr}\left(\mathcal{M}_k^{-1} \frac{\partial \mathcal{M}_k}{\partial x_k}\right)\right]^T$ and $[x_k - x_{kd} \ \ y_k - y_{kd}]^T$ is π rad. Due to Proposition IV.2, the former always points into $\mathrm{CH}(V)$, so this is only possible if for each k, $\mathbf{x}_k \in \mathrm{CH}(V_d)$ (see Figure IV.2).

■

Remark 1. Theorem IV.1 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 $\det\left(\mathcal{M}(\mathbf{x}_d)\right) \geq \bar{\alpha}$.

Remark 2. The control due to connectivity becomes unbounded as $\det \mathcal{M} \to 0$. 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 yet another potential function, $V_a(x)$, designed to introduce collision avoidance behavior, to work in collaboration with the existing ones. By this way, we can guarantee that the robots do not collide while they move towards the desired formation. We use the avoidance (potential) functions as defined in [9] by

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

where d_{ij} is the Euclidean distance between robots i and j, r and R define the avoidance region and sensing region, respectively. The potential functions are designed such that if the robots are started away from the avoidance region $\Omega_{ij} = \left\{x : \|x_i - x_j\|_{\leq} r\right\}$, they never enter this region. The sensing region, on the other hand, given by $\mathcal{D}_{ij} = \left\{x : \|x_i - x_j\|_{\leq} R\right\}$, is the region where robot i can sense the presence of robot j.

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

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

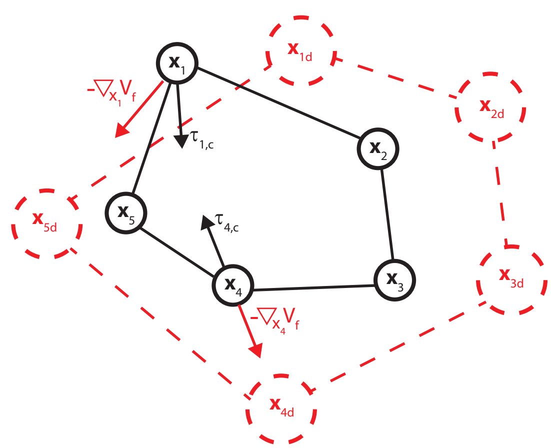


Fig. IV.2: If an agent x_1 is outside the convex hull of the desired formation (depicted by dotted red lines), the formation controller and the connectivity controller cannot conflict so much to cancel the effect of each other.

Thus the form of the control law for robot k with the collision avoidance would be

$$\tau_k = \tau_{k,c}(\mathbf{x}) - k_f \nabla_{\mathbf{x}_i} V_f(\mathbf{x}) - k_a \nabla_{\mathbf{x}_i} V_a(\mathbf{x}) \tag{IV.16}$$

where $\tau_{k,c}(\mathbf{x})$ is the decentralized connectivity controller presented in section IV-B and k_f, k_a are positive gains.

E. Extension to Wheeled Mobile Robots

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

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

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 k^{th} robot with respect to the world 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 (IV.1).

The idea will be to turn the robot to the desired orientation, dictated by the direction of the connectivity controller derived for robots with dynamics (IV.1). 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 $\tilde{X} = \left(q, \theta, X^f, Y^f\right)$, 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 filed $Y : S^N \to S^N \times \mathbb{R}^N$

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

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

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

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

Proposition IV.4. *All of the convergence results presented so far hold for the non-holonomic dynamics as given in* (IV.17) *if the following controller is applied*

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

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

TABLE V.1: 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

Proof: Let us take the time derivative of the potential function $V_{nh} = \int_0^t \langle X(q(\tau)), \dot{q}(\tau)\rangle \, \mathrm{d}\tau + \frac{1}{2}\sum_{k=1}^{N} e_{\theta_k}^2$. Then,

$$\begin{aligned}
\frac{dV_{nh}}{dt} &= \sum_{k=1}^{N} \left\langle \left(\langle X, \frac{\partial}{\partial x_k}\rangle, \langle X, \frac{\partial}{\partial y_k}\rangle \right), (\dot{x}_k, \dot{y}_k) \right\rangle + e_{\theta_k}\dot{e}_{\theta_k} \\
&= \sum_{k=1}^{N} \left\langle \left(\langle X, \frac{\partial}{\partial x_k}\rangle, \langle X, \frac{\partial}{\partial y_k}\rangle \right), (v_k \cos(\theta_k), v_k \sin(\theta_k)) \right\rangle + e_{\theta_k}\dot{e}_{\theta_k} \\
&= \sum_{k=1}^{N} -k_p \cos(e_{\theta_k}) \|\tau_{kd}\| \underbrace{\left\langle \left(\langle X, \frac{\partial}{\partial x_k}\rangle, \langle X, \frac{\partial}{\partial y_k}\rangle \right), (\cos(\theta_k), \sin(\theta_k)) \right\rangle}_{\|\tau_{kd}\| \, \cos(e_{\theta_k})} - K_\theta e_{\theta_k}^2 \\
&= \sum_{k=1}^{N} -k_p \cos^2(e_{\theta_k}) \|\tau_{kd}\|^2 - K_\theta e_{\theta_k}^2 \leq 0
\end{aligned} \quad \text{(IV.21)}$$

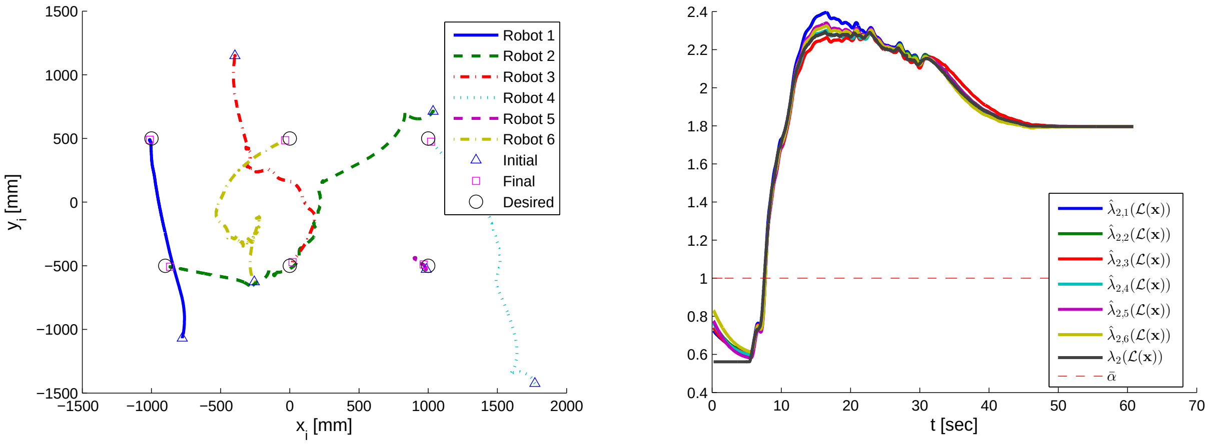
with equality only if $\|\tau_{kd}\| \cos(e_{\theta_k}) = 0$, for all k. But this is only the case if the states are in the desired set. ■

V. Experimental Implementation

The connectivity control is demonstrated using an experimental setup consisting of six iRobot Creates. The kinematics of the Creates are given by (IV.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, which we program in C++. The position feedback is obtained using a VICON motion tracking system. The VICON system has sub-millimeter accuracy with a data rate of 100Hz.

The controllers presented in Section IV are implemented in experiments corresponding to different scenarios. When we refer to controllers developed in Sections IV-B through IV-D, we mean that they have been implemented using the procedure in Section IV-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 (IV.16). We see in figure V.1a 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



(a) Implicit plot of the six robots

(b) Estimates of $\lambda_2(\mathcal{L}(\mathbf{x}))$ and its true value

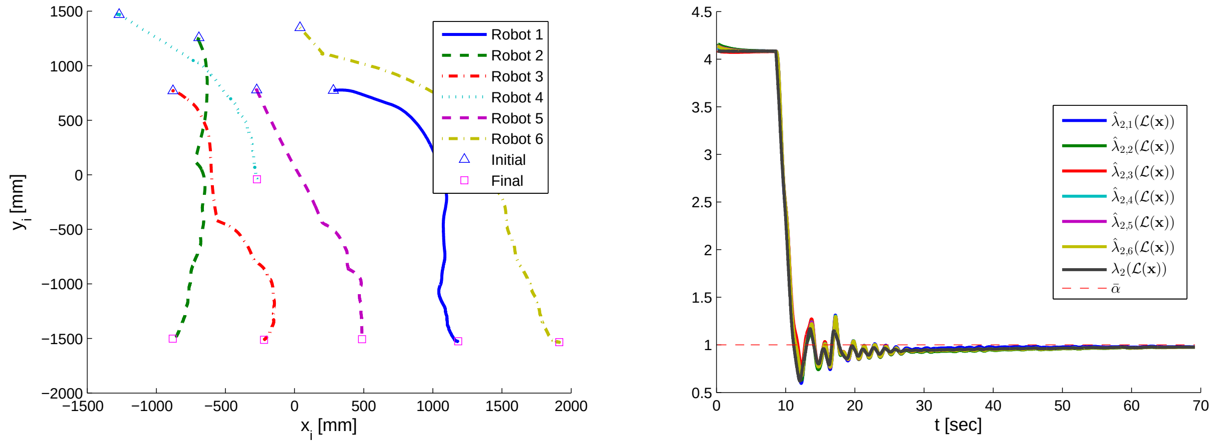
Fig. V.1: Experiment with six Creates running the decentralized connectivity control, formation control and collision avoidance.

positions. Robots 2, 3 and 6 follow a circular path due to collision avoidance, sine they are in each others way. In figure V.1b the initial estimates $\hat{\lambda}_{2,i}$ 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 $\lambda_2(\mathcal{L}(\mathbf{x}))$ 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. The remaining five robots are given desired positions with $y_i = -1500$mm. In figure V.2a, we see that the steady state position errors of these five robots are small. The initial connectivity of the robots is high, that is, $\lambda_2(\mathcal{L}) > 4$. At $t \approx 9$s the five robots move towards 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 figure V.3. The connectivity controller causes the formation to 'drag' Robot 4 in order to maintain a high enough connectivity. The minimum value of $\lambda_2(\mathcal{L})$ is above 0.5 and the robots remain connected throughout the experiment.

VI. 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 increasing the connectivity away from zero whenever it is below a certain threshold. In addition, the connectivity control (IV.2) can be integrated into a previous collision-avoiding formation controller [9] 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.



(a) Implicit plot of the six robots

(b) Estimates of $\lambda_2(\mathcal{L}(\mathbf{x}))$ and its true value

Fig. V.2: 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.

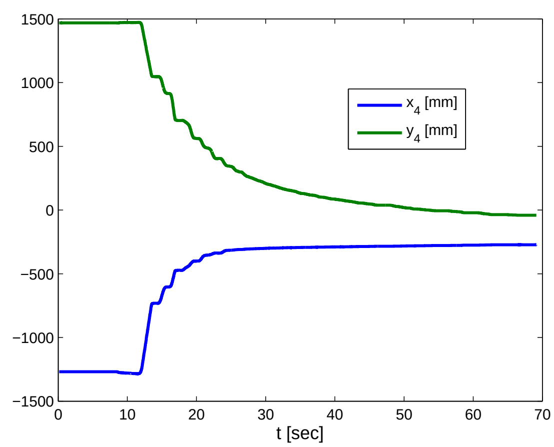


Fig. V.3: The position of Robot 4. The robot moves only when $\lambda_2(\mathcal{L}(\mathbf{x})) < 1$

The decentralized version of the connectivity control law relies on the estimator given in [8]. This controller is shown to behave similar to the centralized version in [7] , 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 [7], we can implement the decentralized controller on a team of non-holonomic wheeled mobile robots. Experiments which demonstrate the properties of the controllers were provided. The expermients 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 IV.1 promises and thus presents a future avenue of research. The effectiveness of the method in [8] is also validated during the experiment.

REFERENCES

[1] M. Zavlanos, M. Egerstedt, and G. Pappas, “Graph-theoretic connectivity control of mobile robot networks,” *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1525 –1540, sept. 2011.

[2] M. Ji and M. Egerstedt, “Distributed coordination control of multiagent systems while preserving connectedness,” *Robotics, IEEE Transactions on*, vol. 23, no. 4, pp. 693 –703, aug. 2007.

[3] M. Zavlanos and G. Pappas, “Controlling connectivity of dynamic graphs,” in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC ’05. 44th IEEE Conference on*, dec. 2005, pp. 6388 – 6393.

[4] ——, “Distributed connectivity control of mobile networks,” *Robotics, IEEE Transactions on*, vol. 24, no. 6, pp. 1416 –1428, dec. 2008.

[5] A. Ajorlou, A. Momeni, and A. Aghdam, “A class of bounded distributed control strategies for connectivity preservation in multi-agent systems,” *Automatic Control, IEEE Transactions on*, vol. 55, no. 12, pp. 2828–2833, 2010.

[6] Y. Kim and M. Mesbahi, “On maximizing the second smallest eigenvalue of a state-dependent graph laplacian,” *Automatic Control, IEEE Transactions on*, vol. 51, no. 1, pp. 116 – 120, jan. 2006.

[7] 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 *submitted to International Conference on Intelligent Robots and Systems*, 2013.

[8] 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 *American Control Conference, 2008*, june 2008, pp. 2678 –2683.

[9] S. Mastellone, D. Stipanovic, and M. Spong, “Remote formation control and collision avoidance for multi-agent nonholonomic systems,” in *Robotics and Automation, 2007 IEEE International Conference on*, april 2007, pp. 1062 –1067.