2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

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

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

Abstract—**The preservation of connectivity in mobile robot networks is critical to the success of existing algorithms designed to achieve various goals. The available connectivity control algorithms mainly work through preservation of existing edges in the network. A link may be deleted if distributed decision making determines that the edge is not a cut-bridge. A controller is presented which allows edges to be broken in a continuous manner without higher-level decision making. The controller is based on maximization of the second smallest eigenvalue of the graph Laplacian. The controllers are designed for holonomic robots, and are extended for implementation on non-holonomic wheeled mobile robots. Finally, the performance of the extended controllers are demonstrated experimentally.**

I. Introduction

Multi-robot networks has been an active area of research for several years. Such systems afford an inexpensive and robust method to achieve certain coverage tasks or cooperative missions. Several of the algorithms that solve tasks using multi-robot networks assume that the network is connected at all times, so that 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.

A good review of different methods to control and maintain connectivity can be found in [1]. The connectivity can be maintained in a centralized or decentralized manner. An obvious method of maintaining connectivity is to preserve the edges present in the network for all time [2]–[4]. Many decentralized connectivity preserving methods utilize a variation of this.

It is clear that in many cases edges may be deleted without losing global connectivity, and several attempts have been made to allow for this behaviour. To the best of our knowledge, all rely on discrete decision making in addition to motion control. The authors in [5], [6] 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.

Another centralized method for maintaining connectivity amongst a group of mobile robots is to maximize the second smallest eigenvalue of the graph Laplacian [7], when 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 [7]. This method is effective for solving rendezvous problems, and can be extended to other applications [1].

The main contribution of this paper is the development of a centralized connectivity controller which maintains, but does not increase the global connectivity until the network is completely connected. Edges may be broken under the influence of additional control objectives (such as exploration or coverage) without losing global connectivity and without needing any higher-level decision making. We consider formation control and collision avoidance as the additional tasks to be achieved. We then extend these controllers in order to facilitate implementation on non-holonomic wheeled mobile robots (WMRs) and then present the results of experiments. 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 to be achieved, where this set of formations can be modified using a parameter in the control law.

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

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

Aykut C Satici, Hasan Poonawala 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).

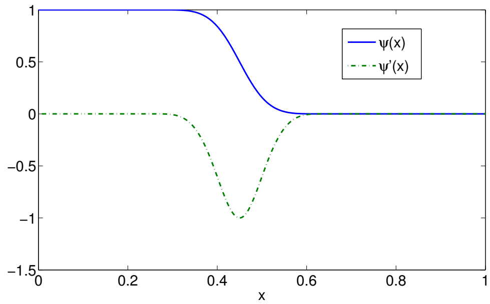


Fig. 1: Bump function

If $\rho_2 - \rho_1 > 1$, then $\psi'(x)$ may be small even when $\rho_1 < x < \rho_2$, unlike as seen in Figure 1. One solution is to normalize the inputs x and parameters ρ_1, ρ_2 through division by ρ_2. 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 input to ψ, we obtain a smooth weighting $w_{ij} = \psi(d_{ij})$ from full connectivity to no connectivity for any two robots, as seen in Figure 1. The edge weights give rise to the graph Laplacian $\mathcal{L} \in \mathbb{R}^{N\times N}$ 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 $\mathcal{L}$. Thus, for the graph to be connected, at most one eigenvalue of $\mathcal{L}$ can be zero. The second smallest eigenvalue $\lambda_2(\mathcal{L})$ thus becomes an indicator of connectivity in the graph.

The Laplacian $\mathcal{L}$ can be converted to a matrix $\mathcal{M} \in \mathbb{R}^{N-1\times N-1}$, whose eigenvalues are the largest $N-1$ eigenvalues of $\mathcal{L}$. The matrix $\mathcal{M}$ is given by

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

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 $\mathcal{M}$ vanishes if and only if $\lambda_2(L)$ vanishes.

III. Motivation

Consider a scenario where we would like a team of robots to stay connected with each other while performing some other task. This task might be to arrange themselves in a formation or to explore an area. 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 connectivity controller based on maximization of $\lambda_2(\mathcal{L})$ and to add another controller that achieves the tracking aspect of the task. The downfall of this approach is that the connectivity control may conflict with the ability of the tracking controller to achieve the desired goal, or may restrict the set of robot positions that can be tracked.

Example. Consider five 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. 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

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

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

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

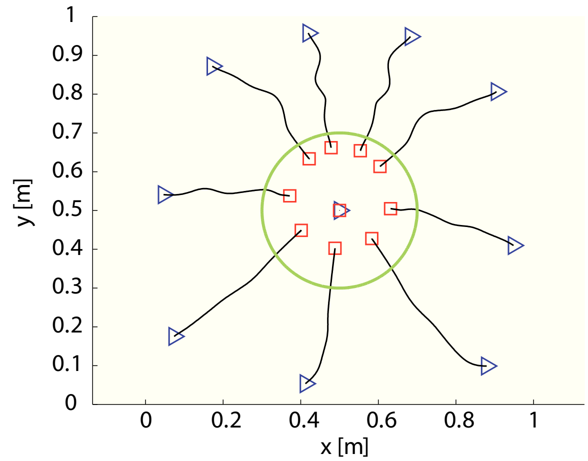


Fig. 2: $\lambda_2(\mathcal{L})$ maximizing controller

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 2. 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 under this 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 desired formations 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

The control goals to achieve in this paper are twofold. The first requirement is for the network of agents to not lose connectivity at any instant in time, provided they are initialized connected. This amounts to keeping the second smallest eigenvalue of the Laplacian matrix $\lambda_2(\mathcal{L})$ away from zero. In addition to this, it is desired that the controller lures the agents to a specified formation.

The maximum possible value of $\lambda_2(\mathcal{L})$ is achieved when the graph is completely connected, however the connectivity when agents achieve the desired formation may not correspond to this value. The connectivity controller is active and pulls the agents away from their desired positions. The formation is not achieved. In order to make the objectives compatible with each other we reformulate the problem as follows: Given a desired formation how does one design a connectivity preserving controller such that the agents are lured towards the desired formation without losing connectivity? We attack the connectivity control problem first in section IV-A, by coming up with a navigation function with parameters that can be modified to achieve any formation control goal a posteriori.

In Sections IV-A through IV-C we assume that the mobile robots possess the following kinematics

$$\dot{\mathbf{x}}_i = \tau_i \tag{3}$$

where $\mathbf{x}_i \in \mathbb{R}^2$ is the position of the i^{th} mobile robot given by $\mathbf{x}_i = (x_i, y_i)$. We then show how each of the controllers can be extended for implementation on non-holonomic wheeled mobile robots in Section IV-D.

A. Connectivity Controller

The connectivity controller must maintain connectivity while possessing some flexibility so that attraction to a desired formation can be simultaneously achieved by another controller to be designed later. The measure of connectivity that we used is the second smallest eigenvalue $\lambda_2(\mathcal{L})$ of the graph Laplacian $\mathcal{L}$, or equivalently the smallest eigenvalue $\underline{\lambda}(\mathcal{M})$ of the matrix $\mathcal{M}$, defined as in equation (2).

We start the design of the connectivity controller by parametrizing the space over which the connectivity controller is active. This is achieved by sandwiching the function $\det(\mathcal{M})$ by two scalars $\underline{\alpha}$ and $\bar{\alpha}$.

$$\underline{\alpha} \leq \det(\mathcal{M}) = \det(\mathcal{P}^T \mathcal{L}(\mathbf{x}) \mathcal{P}) \leq \bar{\alpha}$$

This defines a subset S of the total configuration space $\Pi_{i=1}^{N} \mathbb{R}^{2N}$ where the connectivity controller is active. The connectivity controller will ensure the lower bound $\underline{\alpha}$ is never reached, once the robots are started in $\{\mathbf{x} \in \mathbb{R}^{2N} : \det(\mathcal{M}(\mathbf{x})) \geq \underline{\alpha}\}$. On the other hand, in the complement of this subset S, the connectivity controller will not be active.

These bounds immediately correspond to bounds on $\underline{\lambda}(\mathcal{M})$ once the determinant is seen as the product of the eigenvalues. In other words, a lower bound $\underline{\alpha} > 0$ on the determinant also bounds the second smallest eigenvalue of the graph Laplacian away from zero. Similarly the number $\bar{\alpha} \leq N^{N-1}$ corresponds to an upper threshold for the second smallest eigenvalue. The absolute upper bound N^{N-1} on $\bar{\alpha}$ exists because of the very nature of the graph Laplacian.

The aim of the controller will be to never let the robots reach a configuration where $\det \mathcal{M} < \underline{\alpha}$ and be unresponsive whenever $\det \mathcal{M} > \bar{\alpha}$. We address this problem through the following potential function $V_c(x)$ [8]:

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

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

This function blows up whenever the determinant approaches the lower bound and is zero whenever the determinant is greater than the upper bound. Upon taking partial derivatives of V_c with respect to the coordinates x_k and y_k of the k^{th} robot, we find that

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

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

where

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

Proposition IV.1. *Under the control law*

$$\tau_k = -\nabla_{\mathbf{x}_k} V_c(\mathbf{x}) \tag{4}$$

the team of first-order robots with individual dynamics (3) *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.*

Proof: The first statement is straightforward once we take the derivative of the potential function V_c along the trajectory of the system (3), yielding

$$\begin{aligned} \dot{V}_c &= \nabla_{\mathbf{x}} V_c(\mathbf{x}) \dot{x} \\ &= -\left(\nabla_{\mathbf{x}} V_c(\mathbf{x})\right)^T \nabla_{\mathbf{x}} V_c(\mathbf{x}) \end{aligned}$$

which is smaller than zero whenever $\det(\mathcal{M}(\mathbf{x})) < \bar{\alpha}$.

Consequently, the second statement follows the fact that the level sets $V_c^{-1}([0, \gamma]) = \left\{\mathbf{x} \in \mathbb{R}^{2N} : V_c(\mathbf{x}) \leq \gamma\right\}$ of $V_c(\mathbf{x})$ are positively invariant. ■

B. Connectivity Preserving Formation Controller

In this section, we develop on the connectivity controller presented in Section IV-A by adding a formation controller on top of it. We define a quadratic potential function $V_{fk}(\mathbf{x}_k)$ for each robot 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.

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

$$V_f = \sum_{i}^{N} V_{fk}$$

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.

Proposition IV.2. *The instantaneous direction of motion of each robot i under the control*

$$\tau_i = -\nabla_{\mathbf{x}_i} V_c \tag{5}$$

is a positive combination of the vectors $\mathbf{x}_j - \mathbf{x}_i$, where $j \in \mathcal{N}_i$.

In particular, for each k, the velocity vector of agent k points into the convex hull, $CH(V)$, of the set of vertices V.

Proof:

We define the symmetric matrix $A^{ij} \in \mathbb{R}^{N\times N}$ as

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

where w_{ij} is defined as in (1). This corresponds to the adjacency matrix of a graph consisting of the same N robots, but the only possible non-zero edge weight is w_{ij}. We can construct a graph Laplacian L^{ij} from A^{ij} in the standard way, which has the property that

$$L^{ij} = 2w_{ij}vv^T$$

where $v \in \mathrm{R}^N$ with its k^{th} component given by

$$v_k = \begin{cases} -1/\sqrt{2} & \text{if } k=i \\ 1/\sqrt{2} & \text{if } k=j \\ 0 & \text{otherwise} \end{cases}$$

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

$$\mathcal{L} = \sum_{i=1}^{N}\sum_{j>i}^{N} L^{ij}$$

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

$$\frac{\partial L^{ij}}{\partial x_k} = 2\frac{\partial w_{ij}}{\partial x_k}vv^T$$
$$\frac{\partial L^{ij}}{\partial y_k} = 2\frac{\partial w_{ij}}{\partial y_k}vv^T$$

We take the inverse $\mathcal{M}^{-1} > 0$ of the matrix $\mathcal{M} > 0$ defined in (2) and express it via its eigenvalue decomposition

$$\mathcal{M}^{-1} = \sum_{l=2}^{N}\frac{1}{\lambda_l}u_l u_l^T$$

where $u_l \in \mathbb{R}^N$, for each l. We calculate,

$$\begin{aligned} \operatorname{tr}\left(M^{-1}P^T\frac{\partial L^{ij}}{\partial x_k}P\right) &= \operatorname{tr}\left(\sum_{l=1}^{N-1}\frac{1}{\lambda_l}u_l u_l^T 2\frac{\partial w_{ij}}{\partial x_k}P^T vv^T P\right) \\ &= 2\frac{\partial w_{ij}}{\partial x_k}\operatorname{tr}\left(\sum_{l=1}^{N-1}\frac{1}{\lambda_l}u_l u_l^T P^T v(P^T v)^T\right) \\ &= 2\frac{\partial w_{ij}}{\partial x_k}\sum_{l=1}^{N-1}\frac{1}{\lambda_l}(u_l^T P^T v)^2 = 2\gamma_{ij}\frac{\partial w_{ij}}{\partial x_k} \end{aligned}$$

and similarly,

$$\operatorname{tr}\left(M^{-1}P^T\frac{\partial L^{ij}}{\partial x_k}P\right) = 2\gamma_{ij}\frac{\partial w_{ij}}{\partial y_k}$$

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

$$\begin{aligned} \tau_k^{ij} &= \begin{bmatrix} \operatorname{tr}\left(\mathcal{M}^{-1}P^T\frac{\partial L^{ij}}{\partial x_k}P\right) \\ \operatorname{tr}\left(\mathcal{M}^{-1}P^T\frac{\partial L^{ij}}{\partial y_k}P\right) \end{bmatrix} \\ &= 2\gamma_{ij}\begin{bmatrix} \frac{\partial w_{ij}}{\partial x_k} \\ \frac{\partial w_{ij}}{\partial y_k} \end{bmatrix} \end{aligned}$$

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 k^{th} robot, we use the fact that

$$\frac{\partial \mathcal{L}}{\partial x_k} = \sum_{i=1}^{N}\sum_{j>i}^{N}\frac{\partial L^{ij}}{\partial x_k} = \sum_{j\neq k}^{N}\frac{\partial L^{kj}}{\partial x_k}$$

since $L^{ij} = L^{ji}$ and $\frac{\partial L^{ij}}{\partial x_k} = 0$ whenever $k \neq i,\ k \neq j$. Therefore, the control vector (5) is computed as

$$\begin{aligned} \tau_i &= -\beta(\mathbf{x})\begin{bmatrix} \operatorname{tr}\left(\mathcal{M}^{-1}P^T\frac{\partial L}{\partial x_i}P\right) \\ \operatorname{tr}\left(\mathcal{M}^{-1}P^T\frac{\partial L}{\partial y_i}P\right) \end{bmatrix} \\ &= -\beta(\mathbf{x})\sum_{j}^{N}\begin{bmatrix} \operatorname{tr}\left(\mathcal{M}^{-1}P^T\frac{\partial L^{ij}}{\partial x_i}P\right) \\ \operatorname{tr}\left(\mathcal{M}^{-1}P^T\frac{\partial L^{ij}}{\partial y_i}P\right) \end{bmatrix} \\ &= -\beta(\mathbf{x})\sum_{j}^{N}2\gamma_{ij}\delta_{ij}\begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} \end{aligned}$$

which is clearly a positive combination of the displacement vectors from robot i to the robots j. Moreover, $j \in \mathcal{N}_i$ since otherwise $\frac{\partial w_{ij}}{\partial d_{ij}} = 0$ and hence $\delta_{ij} = 0$.

■

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

$$\tau_k = -\left(k_c\nabla_{\mathbf{x}_i}V_c(\mathbf{x}) + k_f\nabla_{\mathbf{x}_i}V_f(\mathbf{x})\right) \tag{6}$$

for positive gains $k_c, k_f > 0$. The next proposition states a consensus-like behavior of the robots under this control law.

Proposition IV.3. *Suppose the control effort for each robot k is given by* (6)*. Then the average position of the robots converge to the desired average position. In other words,*

$$\lim_{t\to\infty}\frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k(t) = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_{kd}$$

Proof: Take the Lyapunov function candidate $V = V_c(\mathbf{x}) + V_f(\mathbf{x}) > 0$ and take its derivative along the solutions of the system (3).

$$\begin{aligned} \dot{V} = &-\sum_{k=1}^{N}\left[k_c\beta(\mathbf{x})\operatorname{tr}\left(\mathcal{M}^{-1}\frac{\partial \mathcal{M}}{\partial x_k}\right) + k_f\left(x_k - x_{kd}\right)\right]^2 \\ &-\sum_{k=1}^{N}\left[k_c\beta(\mathbf{x})\operatorname{tr}\left(\mathcal{M}^{-1}\frac{\partial \mathcal{M}}{\partial y_k}\right) + k_f\left(y_k - y_{kd}\right)\right]^2 \end{aligned}$$

which is equal to zero only when the expressions in the square terms are zero for all k. Since the same arguments holds for the second square term in this expression (y-component), we concentrate on the first square term.

Using the identity $\mathrm{tr}(AB) + \mathrm{tr}(AC) = \mathrm{tr}\left(A(B+C)\right)$,

$$\begin{aligned}
&\sum_{k=1}^{N}\left[k_c\beta(\mathbf{x})\mathrm{tr}\left(\mathcal{M}^{-1}\frac{\partial\mathcal{M}}{\partial x_k}\right) + k_f\left(x_k - x_{kd}\right)\right] \\
&\quad= \sum_{k=1}^{N} k_c\beta(\mathbf{x})\mathrm{tr}\left(\mathcal{M}^{-1}\frac{\partial\mathcal{M}}{\partial x_k}\right) + \sum_{k=1}^{N} k_f\left(x_k - x_{kd}\right) \\
&\quad= k_c\beta(\mathbf{x})\mathrm{tr}\left(\mathcal{M}^{-1}\sum_{k=1}^{N}\frac{\partial\mathcal{M}}{\partial x_k}\right) + \sum_{k=1}^{N} k_f\left(x_k - x_{kd}\right) \\
&\quad= \sum_{k=1}^{N} k_f\left(x_k - x_{kd}\right)
\end{aligned}$$

where we have used the anti-symmetry of $\frac{\partial\mathcal{M}}{\partial x_k}$ with respect to the differentiated variable to write down the last equality. This completes the proof. ■

Theorem IV.1. *Suppose the control effort for each agent i is given by (6). Let V_d denote the vertex set for the desired formation. Then the agents converge to a set E contained in the convex hull $\mathrm{CH}(V_d)$ of the desired formation.*

Proof: We take the same Lyapunov function candidate as in the proof of Proposition IV.3 and compute its derivative along the solutions of (3), concluding that the positively invariant set the agents reach is the set of configurations $\mathbf{x} \in \mathbb{R}^{2N}$ where for all i, both the following conditions hold

$$\begin{aligned}
k_c\beta(\mathbf{x})\mathrm{tr}\left(\mathcal{M}^{-1}\frac{\partial\mathcal{M}}{\partial x_i}\right) + k_f\left(x_i - x_{id}\right) &= 0 \\
k_c\beta(\mathbf{x})\mathrm{tr}\left(\mathcal{M}^{-1}\frac{\partial\mathcal{M}}{\partial y_i}\right) + k_f\left(y_i - y_{id}\right) &= 0
\end{aligned}$$

Since $\beta(\mathbf{x}) < 0$, this is equivalent to the statement that the angle between the vectors $\left[\mathrm{tr}\left(\mathcal{M}^{-1}\frac{\partial\mathcal{M}}{\partial x_i}\right)\ \ \mathrm{tr}\left(\mathcal{M}^{-1}\frac{\partial\mathcal{M}}{\partial x_i}\right)\right]^T$ and $[x_i - x_{id}\ \ y_i - y_{id}]^T$ is π rad. Due to Proposition IV.2, the former always points into $\mathrm{CH}(V)$, so this is only possible if for each i, $\mathbf{x}_i \in \mathrm{CH}(V_d)$ (see Figure 3). ■

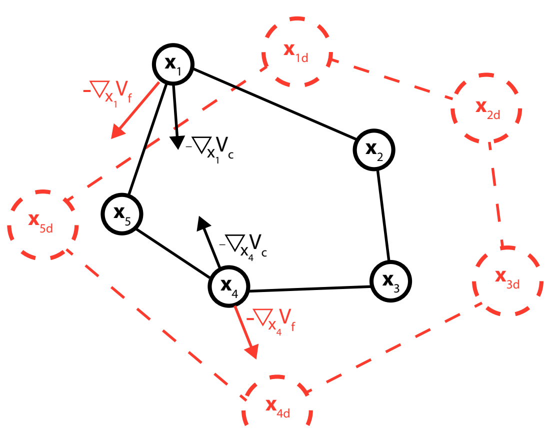


Fig. 3: If an agent x_1 is outside the convex hull of the desired formation (dashed red lines), the formation controller and the connectivity controller cannot cancel each other.

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 $\det\mathcal{M} \rightarrow 0$. Finite errors in formation yield finite control effort, hence even if the desired formation is disconnected, the network will never become disconnected.

C. 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 prevent that the robots from colliding with each other while they move towards the desired formation. We use the avoidance (potential) functions as defined in [8] by

$$V_{aij} = \left(\min\left\{0, \frac{d_{ij}^2 - R^2}{d_{ij}^2 - r^2}\right\}\right)^2 \tag{7}$$

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 $\Omega_{ij} = \{x : \|x_i - x_j\| \leq r\}$, they never enter this region. The sensing region, on the other hand, given by $\mathcal{D}_{ij} = \{x : \|x_i - x_j\| \leq R\}$, is the region where robot i can sense the presence of robot j.

The sum of the pairwise potentials (7) between robots i and j constitute the total avoidance potential function

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

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

$$\tau_k = -\left(k_c\nabla_{\mathbf{x}_i}V_c(\mathbf{x}) + k_f\nabla_{\mathbf{x}_i}V_f(\mathbf{x}) + k_a\nabla_{\mathbf{x}_i}V_a(\mathbf{x})\right) \tag{8}$$

where k_c, k_f, and k_a are positive gains.

D. Extension to Wheeled Mobile Robots

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

$$\begin{aligned}
\dot{x}_k &= v_k\cos\left(\theta_k\right) \\
\dot{y}_k &= v_k\sin\left(\theta_k\right) \\
\dot{\theta}_k &= \omega_k
\end{aligned} \tag{9}$$

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 (3).

The idea will be to turn the robot to the desired orientation dictated by the direction of the controller derived for robots with dynamics (3) based on any potential function V

$$\theta_{kd} = \arctan_2\left(\frac{\partial V}{\partial y_k}, \frac{\partial V}{\partial x_k}\right) \tag{10}$$

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

$$\tau_{kd} := \left(\frac{\partial V}{\partial x_k}, \frac{\partial V}{\partial y_k}\right) \tag{11}$$

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* (9) *if the following controller is applied*

$$\begin{aligned} v_k &= -k_p \cos\left(e_{\theta_k}\right) ||\tau_{kd}|| \\ \omega_k &= -k_\theta e_{\theta_k} \end{aligned} \tag{12}$$

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

Proof: Let us take the time derivative of the potential function $V_{nh} = V + \frac{1}{2}\sum_{k=1}^{N} e_{\theta_k}^2$, where V is the potential function used to derive any controller for robots with first-order integrator dynamics. Then,

$$\begin{aligned} \frac{dV_{nh}}{dt} &= \sum_{k=1}^{N}\left[\frac{\partial V}{\partial x_k}\dot{x}_k + \frac{\partial V}{\partial y_k}\dot{y}_k + e_{\theta_k}\dot{e}_{\theta_k}\right] \\ &= \sum_{k=1}^{N}\Bigg[-k_p\cos\left(e_{\theta_k}\right)||\tau_{kd}||\left(\frac{\partial V}{\partial x_k}\cos\left(\theta_k\right)\right. \\ &\qquad\qquad \left. + \frac{\partial V}{\partial y_k}\sin\left(\theta_k\right)\right) - k_\theta e_{\theta_k}^2\Bigg] \\ &= \sum_{k=1}^{N}\left[-k_p\cos^2\left(e_{\theta_k}\right)||\tau_{kd}||^2 - k_\theta e_{\theta_k}^2\right] \leq 0 \end{aligned} \tag{13}$$

with equality only if $\|\tau_{kd}\| = e_{\theta_k} = 0$, for all k. But this is only the case if the states are in the desired set. The last equality in (13) is obtained by the observing that

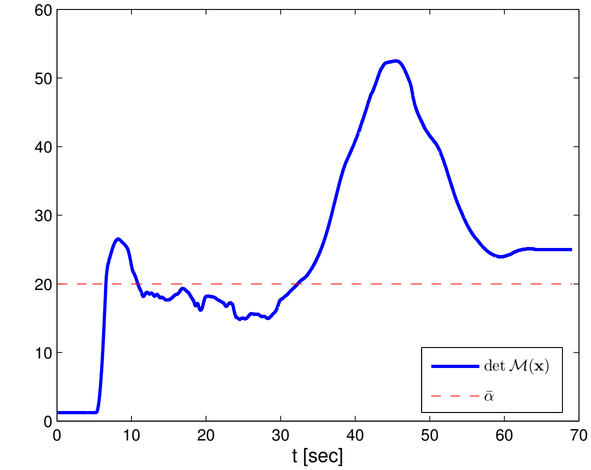
$$\begin{aligned} &\left(\frac{\partial V}{\partial x_k}\cos\left(\theta_k\right) + \frac{\partial V}{\partial y_k}\sin\left(\theta_k\right)\right) \\ &= \left\langle\left(\frac{\partial V}{\partial x_k}, \frac{\partial V}{\partial y_k}\right), \left(\cos\left(\theta_k\right), \sin\left(\theta_k\right)\right)\right\rangle \\ &= \|\tau_{kd}\|\cos e_{\theta_k} \end{aligned}$$

■

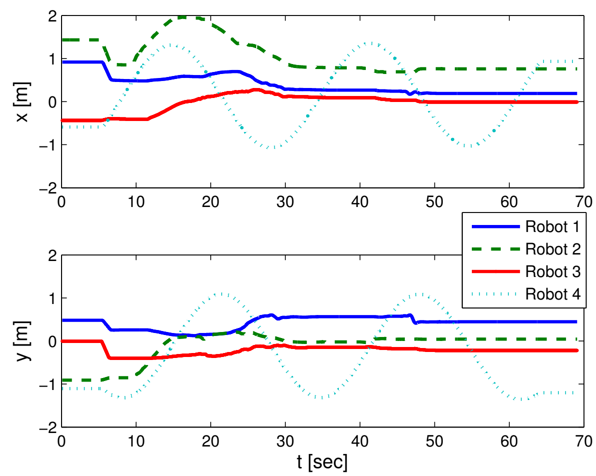
V. Experimental Implementation

The connectivity control is demonstrated using an experimental setup consisting of four iRobot Creates. The kinematics of the Creates are given by (9), 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-A through IV-C, we mean that they have been implemented using the procedure in Section IV-D.



(a) det $(\mathcal{M}(\mathbf{x}))$



(b) $x - y$ coordinates for each robot

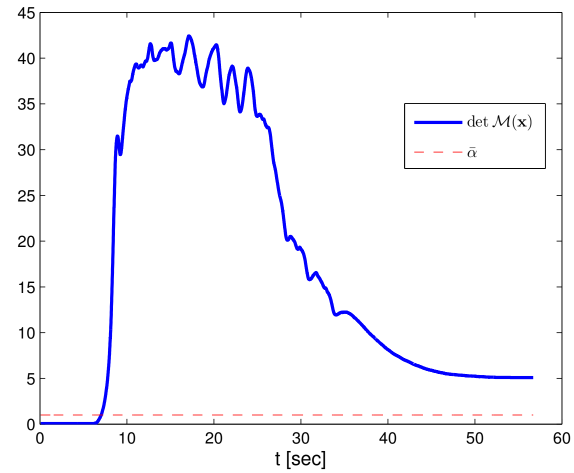
Fig. 4: Experiment where Robot 4 moves in a circle, while Robots $1-3$ avoid collisions and maintain connectivity.

In the first experiment, one robot is commanded to move with constant linear velocity 0.3m/s and angular velocity 0.25rad/s so that it moves in a circle. The remaining robots run controller (8), with parameters given in Table I. Note that the formation control gain k_f is set to zero. The other robots move in reaction to the motion of Robot 4, until they finally settle into positions such that $\det\mathcal{M} \geq \bar{\alpha}$ (Figures 4a and 4b). We omit the plot for y position due to space constraints. The brief motion at $t = 46$s is due to collision avoidance. This demonstrates how connectivity can be maintained above a desired value without preference to any particular network topology.

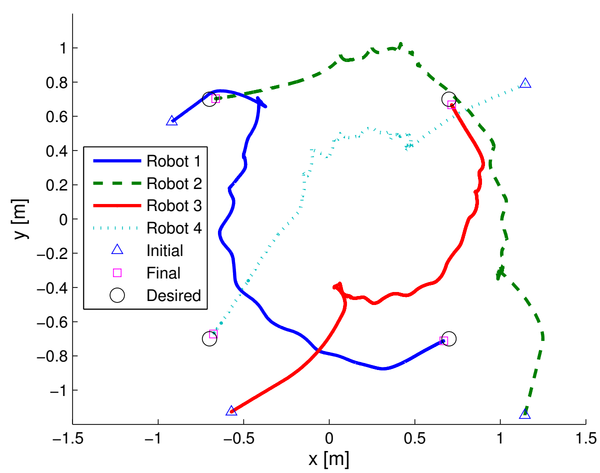
In the second experiment, all the robots are tasked with converging to a formation, while maintaining connectivity and avoiding collisions (Using (8)). The initial configuration is such that each robot must pass near the centroid of the desired formation if only the formation control were active, ensuring that collisions may occur if without the collision

TABLE I: Parameters used in the 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	2.0	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



(a) $\det(\mathcal{M}(\mathbf{x}))$



(b) Implicit plot

Fig. 5: Experiments with four Creates performing connectivity control, formation control and collision avoidance.

avoidance terms. Notice in figure 5b that Robots 3 and 4 initially move in a straight line and then move in a circular path due to collision avoidance. We see that the steady state position errors of the robots are small, and are a result of the dead-zone in actuation. The connectivity increases from the initial small value, as seen in figure 5a. Note that the final value of $\det \mathcal{M}(\mathbf{x})$ is less than the maximum possible value of 64. Also note that the connectivity decreases as the robots pull away from each other to reach their desired positions in the formation.

VI. Conclusion

In this paper we have presented a connectivity control method for a mobile network based on maximization of the second smallest eigenvalue $\lambda_2(\mathcal{L})$ of the graph Laplacian $\mathcal{L}$. This can be achieved by maximizing a measure of connectivity given by the determinant of a matrix $\mathcal{M} = \mathcal{P}^T\mathcal{L}\mathcal{P}$. The connectivity is increased away from zero whenever it is below a certain threshold. In addition, the connectivity control (4) can be integrated into a previous collision-avoiding formation controller [8], with similar convergence properties, provided the desired formation has a value of $\det(\mathcal{M})$ above the threshold used in our control. The set of formations that can be tracked using our connectivity control is larger than those which disallow edge deletions or maximize $\lambda_2(\mathcal{L})$ till the theoretical limit of N.

We have shown how to extend the controllers in order to be able to implement them on non-holonomic 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 IV.1 promises and thus presents a future avenue of research.

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] J. Reich, V. Misra, D. Rubenstein, and G. Zussman, "Connectivity maintenance in mobile wireless networks via constrained mobility," *Selected Areas in Communications, IEEE Journal on*, vol. 30, no. 5, pp. 935–950, 2012.

[5] M. Zavlanos and G. Pappas, "Distributed connectivity control of mobile networks," *Robotics, IEEE Transactions on*, vol. 24, no. 6, pp. 1416 – 1428, dec. 2008.

[6] J. Wagenpfeil, A. Trachte, T. Hatanaka, M. Fujita, and O. Sawodny, "A distributed minimum restrictive connectivity maintenance algorithm," in *9th IFAC Symposium on Robot Control SYROCO2009*, no. 2003. Robot Control, Volume 9 — Part 1, 2009, pp. 499–504. [Online]. Available: http://www.ifac-papersonline.net/Detailed/45637.html

[7] 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.

[8] 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.