

Potential Fields for Maintaining Connectivity of Mobile Networks

Michael M. Zavlanos, *Student Member, IEEE*, and George J. Pappas, *Senior Member, IEEE*

Abstract—The control of mobile networks of multiple agents raises fundamental and novel problems in controlling the structure of the resulting dynamic graphs. In this paper, we consider the problem of controlling a network of agents so that the resulting motion always preserves the *connectivity* property of the network. In particular, the connectivity condition is translated to differentiable constraints on individual agent motion by considering the dynamics of the Laplacian matrix and its spectral properties. Artificial potential fields are then used to drive the agents to configurations away from the undesired space of disconnected networks while avoiding collisions with each other. We conclude by illustrating a class of interesting problems that can be achieved while preserving connectivity constraints.

Index Terms—potential fields, dynamic graphs, Laplacian matrix, graph connectivity.

I. INTRODUCTION

Controlling dynamic graphs has recently emerged as a fundamental problem in the area of multi-agent and multi-robot systems. Motivations come from the area of controlling formations of ground or aerial vehicles with applications in air traffic control, satellite clustering, automatic highways, mobile robotics and mobile sensor networks. One of the main goals in this area is to achieve a coordinated objective while using only relative information concerning positions and velocities. The objective investigated in this paper is that of maintaining connectivity of the underlying graph.

Due to their frequent appearance in the area of multi-agent systems, dynamic graphs have already received considerable attention. In [2], a measure of local connectedness of a network, that under certain conditions is sufficient for global connectedness, is introduced. This approach is distributed in the sense that this measure depends on neighbor-to-neighbor communication only. Motivated by a class of problems associated with control of distributed dynamic systems is also [3], where the authors consider a controllability framework for state-dependent dynamic graphs. In [4], the problem of finding the graph that corresponds to the maximum second smallest eigenvalue of its Laplacian is investigated. The authors propose a method that searches the graph space towards the direction that maximizes the second smallest eigenvalue of the graph Laplacian, and prove local convergence of their method. The second smallest eigenvalue of the graph Laplacian has also emerged as an important parameter in many system and control problems defined over networks [5], [6], [7]. In fact, recently, it

has been observed that this eigenvalue is a measure of stability and robustness of the networked dynamic system [6], [7].

Other research areas closely related to the problems discussed in this paper are formation stabilization [5] – [17], and in particular, consensus seeking by autonomous agents [5], [6], [7], as well as coverage tasks [18]. One of the main goals in formation stabilization and consensus problems is convergence of the agents to a common velocity. Various approaches have been studied, such as, control laws that involve graph Laplacians for the fixed (or switched) associated neighborhood graphs and artificial potential functions [7] or Lyapunov function methods [9], where the notion of “formation feedback” as a means to regulate agent motion in order to satisfy the global formation constraints, was also introduced. Navigation functions for distributed formation stabilization with collision avoidance constraints are used in [10], [11], whereas necessary and sufficient conditions for achieving consensus are investigated in [6].

Motivated by the importance of connectivity in mobile sensor networks as well as the connectivity assumption often made in formation stabilization or consensus problems, in this paper, we consider graph connectivity as our primary objective. As in our previous work [1], we consider the problem of designing controllers for multi-agent systems that *maintain* connectivity of their underlying graphs. In contrast to the open loop controllers based on the solution of optimization problems [1], [4], in this paper we propose a centralized feedback control framework based on artificial potential fields. In the absence of agent failures and under the assumption that the initial graph is connected, graph connectivity is guaranteed for all time. The idea is to model connectivity as an imaginary obstacle in our free space and use artificial potential fields to avoid collisions with it. Instead of the adjacency matrix and its dynamics [1], we use as a model for connectivity the Laplacian matrix of a graph and its spectral properties. Due to its closed loop nature, our potential fields approach to maintaining graph connectivity is more robust and easily extendible to accounting for secondary objectives, such as collision avoidance between the agents or the presence of formation leaders in the group.

The rest of this paper is organized as follows. In Section II we define the network connectivity problem and develop the necessary graph theoretic background. In Section III, we deal with the technical issues of our approach and propose the potential fields based solution to the problem of maintaining connectivity of mobile networks. Finally, in Section IV, we discuss secondary objectives, such as collision avoidance and tracking leaders, and verify through computer simulations, various connectivity tasks that illustrate the setting we have developed.

This work is partially supported by ARO MURI SWARMS Grant W911NF-05-1-0219 and the NSF ITR Grant 0324977.

Michael M. Zavlanos and George J. Pappas are with GRASP Laboratory, Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA {zavlanos, pappasg}@grasp.upenn.edu

II. PROBLEM FORMULATION

A. Graph Theoretic Formulation

Consider n nodes in an m -dimensional space \mathbb{R}^m . We denote by $x_i(t) \in \mathbb{R}^m$ the coordinates of the i -th node at time t , where by convention, x_i is considered a $m \times 1$ column vector, and by $\mathbf{x}(t) = [x_1^T(t) \dots x_n^T(t)]^T$ the $mn \times 1$ vector resulting from stacking the coordinates of the nodes into a single vector. Suppose that the dynamics of the i -th node, for all $i \in \{1, 2, \dots, n\}$, are given by $\dot{x}_i(t) = u_i(t)$, where $u_i(t)$ is the control vector taking values in some set $U \subseteq \mathbb{R}^m$. In vector notation, the system dynamics become,

$$\dot{\mathbf{x}}(t) = \mathbf{u}(t) \quad (1)$$

where $\dot{\mathbf{x}}(t) = [\dot{x}_1^T(t) \dots \dot{x}_n^T(t)]^T$ and $\mathbf{u}(t) = [u_1^T(t) \dots u_n^T(t)]^T$ are $mn \times 1$ vectors, respectively. The network of agents described by system (1), gives rise to a dynamic graph $\mathcal{G}(\mathbf{x}(t))$.¹

Definition 2.1: We call $\mathcal{G}(\mathbf{x}(t)) = (\mathcal{V}, \mathcal{E}(\mathbf{x}(t)))$ a dynamic graph consisting of,

- a set of vertices $\mathcal{V} = \{1, \dots, n\}$ indexed by the set of mobile agents.
- a set of edges $\mathcal{E}(\mathbf{x}(t)) = \{(i, j) \mid d_{ij}(\mathbf{x}(t)) < \delta\}$, with $d_{ij}(\mathbf{x}(t)) = \|x_i(t) - x_j(t)\|_2$ the Euclidean distance between agents i and j and $\delta > 0$ a constant.

Since we have control over the node (or vertex) dynamics, the question that naturally arises is whether we can control the motion of the agents, so that $\mathcal{G}(\mathbf{x}(t))$ satisfies a graph theoretic property of interest for all time $t \geq 0$. In particular, we are interested in the connectivity property of a graph.

Definition 2.2 (Graph Connectivity): We say that a graph \mathcal{G} is connected if there exists a path, i.e., a sequence of distinct vertices such that consecutive vertices are adjacent, between any two vertices in \mathcal{G} .

We denote by \mathcal{C}_n the set of all connected graphs on n vertices, and we address the problem of whether we can control the motion of all agents so that $\mathcal{G}(\mathbf{x}(t))$ always lies in the desired set \mathcal{C}_n of connected graphs. More formally,

Problem 1 (Graph Theoretic Formulation): Let \mathcal{C}_n be the set of connected graphs and assume any input to the graph $\mathcal{G}(\mathbf{x}(t))$ in the form of node motion. Given \mathcal{C}_n , determine a control law $\mathbf{u}(t)$ so that if $\mathcal{G}(\mathbf{x}(0)) \in \mathcal{C}_n$ then $\mathcal{G}(\mathbf{x}(t)) \in \mathcal{C}_n$ for all $t \geq 0$.

In other words, we would like the set \mathcal{C}_n to be an invariant of motion for system (1). We will achieve this goal by choosing an equivalent formulation using the algebraic representation of the dynamic graph $\mathcal{G}(\mathbf{x}(t))$.

B. Algebraic Formulation

The edge set of any dynamic graph can be associated with a set of weights such that for all $(i, j) \in \mathcal{E}(\mathbf{x}(t))$ the corresponding weight equals 1 and for all $(i, j) \notin \mathcal{E}(\mathbf{x}(t))$ the corresponding weight equals 0. In matrix form, these weights consist the *weighted adjacency matrix* of a graph. The 0-1 nature of these weights can be relaxed to allow for any positive

¹In the literature also referred to as a *proximity graph* due to the definition of the edge set.

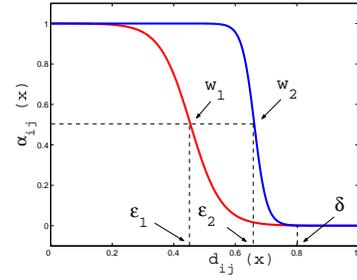


Fig. 1. Weight function $a_{ij}(\mathbf{x}) = \sigma_w(\epsilon - d_{ij}(\mathbf{x}))$ for $\delta = 0.8$ and parameter values $w_1 = 20$, $\epsilon_1 = 0.4547$ and $w_2 = 50$ and $\epsilon_2 = 0.6619$.

weight for all $(i, j) \in \mathcal{E}(\mathbf{x}(t))$. This relaxation enables us to define smooth weight functions in our potential fields model. We, hence, have the following definition.

Definition 2.3 (Weighted Adjacency Matrix): Given a graph \mathcal{G} with vertices $\mathcal{V} = \{v_1, \dots, v_n\}$ and edges in the set \mathcal{E} , we define the weighted adjacency matrix of \mathcal{G} to be the matrix $A = (a_{ij})$ such that $a_{ij} > 0$ if $(v_i, v_j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. Since we do not allow self-loops, for every $i \in \{1, 2, \dots, n\}$ we define $a_{ii} = 0$.

Note that if A is an adjacency matrix of an undirected graph, as will be the case we will be dealing with, then $A = A^T$. In order to capture the dependence of the edge set $\mathcal{E}(\mathbf{x}(t))$ on the state $\mathbf{x}(t)$, we need to introduce a dynamic adjacency matrix,

$$A(\mathbf{x}(t)) = (a_{ij}(\mathbf{x}(t))) \quad (2)$$

where $a_{ij}(\mathbf{x}(t)) = \sigma_w(\epsilon - d_{ij}(\mathbf{x}(t)))$ and $\sigma_w(y) = \frac{1}{1 + e^{-wy}}$ is the sigmoid function with $w > 0$ a constant (Figure 1).² Clearly, for any $\delta > 0$ in Definition 2.1 there exist constants $w, \epsilon > 0$ such that $a_{ij}(\mathbf{x}(t)) > 0$ for $d_{ij}(\mathbf{x}(t)) < \delta$ and $a_{ij}(\mathbf{x}(t)) \rightarrow 0$ for $d_{ij}(\mathbf{x}(t)) \geq \delta$. Hence, equation (2) is, pointwise in $\mathbf{x}(t)$, consistent with Definition 2.3 of a weighted Adjacency matrix. Moreover, since $d_{ij}(\mathbf{x}(t)) = d_{ji}(\mathbf{x}(t))$ we have that $a_{ij}(\mathbf{x}(t)) = a_{ji}(\mathbf{x}(t))$ and so $A(\mathbf{x}(t))$ is symmetric.

Another way to represent the structure of a graph $\mathcal{G}(\mathbf{x}(t))$ is by a weighted Laplacian matrix,

$$L(\mathbf{x}(t)) = \Delta(\mathbf{x}(t)) - A(\mathbf{x}(t)) \quad (3)$$

where $\Delta(\mathbf{x}(t)) = \text{diag}\left(\sum_{j=1}^n a_{ij}(\mathbf{x}(t))\right)$ is the *Valency* matrix. The spectral properties of the Laplacian matrix are closely related to graph connectivity. In particular,

Lemma 2.4 ([19]): Let $\lambda_1(\mathbf{x}(t)) \leq \dots \leq \lambda_n(\mathbf{x}(t))$ be the ordered eigenvalues of the Laplacian matrix $L(\mathbf{x}(t))$. Then,

- $\lambda_1(\mathbf{x}(t)) = 0$ for all $\mathbf{x}(t) \in \mathbb{R}^{nm}$, with corresponding eigenvector $\mathbf{1}$, i.e., the vector of all entries equal to 1.
- $\lambda_2(\mathbf{x}(t)) > 0$ if and only if $\mathcal{G}(\mathbf{x}(t))$ is connected.

Using Lemma 2.4(b) we can define the set $\mathcal{X}_{\mathcal{C}_n}$ of desired states corresponding to graphs in \mathcal{C}_n as,

$$\mathcal{X}_{\mathcal{C}_n} = \{ \mathbf{x}(t) \in \mathbb{R}^{nm} \mid \lambda_2(\mathbf{x}(t)) > 0 \} \quad (4)$$

and so Problem 1 can be reformulated as,

Problem 2 (Algebraic Formulation): Let $\mathcal{X}_{\mathcal{C}_n}$ be the desired set of states and assume any input to the graph $\mathcal{G}(\mathbf{x}(t))$ in the form of node motion. Given $\mathcal{X}_{\mathcal{C}_n}$, determine a control law $\mathbf{u}(t)$ so that if $\mathbf{x}(0) \in \mathcal{X}_{\mathcal{C}_n}$ then $\mathbf{x}(t) \in \mathcal{X}_{\mathcal{C}_n}$ for all $t \geq 0$.

²Note that any other smooth function with similar properties can be used.

III. POTENTIAL FIELD FOR GRAPH CONNECTIVITY

Let $\mathcal{G}(\mathbf{x}(t))$ be a dynamic graph and $L(\mathbf{x}(t))$ its weighted Laplacian matrix as defined in equation (3). Let P be a fixed $n \times (n-1)$ matrix with column structure $P = [p_1 \ p_2 \ \dots \ p_n]$ such that $p_i^T p_j = 0$ for all $i, j = 1, \dots, n$ and $p_i^T \mathbf{1} = 0$ for all $i = 1, \dots, n$, where $\mathbf{1}$ denotes an n -dimensional vector with all entries equal to 1.³ By Lemma 2.4(a), $L(\mathbf{x}(t))$ is positive semi-definite for all $\mathbf{x}(t) \in \mathbb{R}^{nm}$ and hence, we have the following result for the expression $P^T L(\mathbf{x})P$.⁴

Proposition 3.1: $P^T L(\mathbf{x})P \succeq 0$ for all $\mathbf{x} \in \mathbb{R}^{nm}$.

Proof: Let $\mathbf{z} \in \mathbb{R}^{n-1}$ and consider the quadratic form $\mathbf{z}^T P^T L(\mathbf{x})P \mathbf{z} = (P\mathbf{z})^T L(\mathbf{x})P\mathbf{z}$. Let $\mathbf{w} = P\mathbf{z}$. Since P is full rank, $\mathbf{w} = P\mathbf{z}$ defines an injective mapping between \mathbb{R}^{n-1} and \mathbb{R}^n and since $\mathbf{w}^T L(\mathbf{x})\mathbf{w} \geq 0$ for all $\mathbf{w} \in \mathbb{R}^n$ and all $\mathbf{x} \in \mathbb{R}^{nm}$ we conclude that $\mathbf{z}^T P^T L(\mathbf{x})P \mathbf{z} \geq 0$ for all $\mathbf{z} \in \mathbb{R}^{n-1}$ and all $\mathbf{x} \in \mathbb{R}^{nm}$ which completes the proof. ■

The following proposition characterizes the eigenvalues of $P^T L(\mathbf{x})P$.

Proposition 3.2: Let $0 = \lambda_1(\mathbf{x}) \leq \lambda_2(\mathbf{x}) \leq \dots \leq \lambda_n(\mathbf{x})$ be the ordered eigenvalues of the Laplacian matrix $L(\mathbf{x})$. Then, $\lambda_2(\mathbf{x}) \leq \dots \leq \lambda_n(\mathbf{x})$ are the eigenvalues $P^T L(\mathbf{x})P$.

Proof: The proof of this result is based on the relation between the characteristic polynomials of $L(\mathbf{x})$ and $P^T L(\mathbf{x})P$. Since, $\begin{bmatrix} P^T \\ \frac{1}{\|\mathbf{1}\|} \mathbf{1}^T \end{bmatrix} \begin{bmatrix} P & \frac{1}{\|\mathbf{1}\|} \mathbf{1} \end{bmatrix} = I_n$ we have that,

$$\begin{aligned} \det(L(\mathbf{x}) - sI) &= \det \begin{bmatrix} P^T \\ \frac{1}{\|\mathbf{1}\|} \mathbf{1}^T \end{bmatrix} (L(\mathbf{x}) - sI) \begin{bmatrix} P & \frac{1}{\|\mathbf{1}\|} \mathbf{1} \end{bmatrix} \\ &= \det \begin{bmatrix} P^T(L(\mathbf{x}) - sI)P & 0 \\ 0 & -s \end{bmatrix} \\ &= -s \det P^T(L(\mathbf{x}) - sI)P \end{aligned}$$

which completes the proof. ■

A *sufficient* condition that guarantees that $\mathbf{x}(t) \in \mathcal{X}_{C_n}$ for all $t \geq 0$ is to choose $\mathbf{u}(t)$ so that $\lambda_2(\mathbf{x}(t))$ is maximized over time. This approach is followed in [4] where the authors discretize the problem and, at every time step, solve the semi-definite program $\max\{\epsilon \mid P^T L(\mathbf{x}(t))P \succeq \epsilon I_n\}$. In this paper we choose a different approach and construct a potential field that “blows up” when the state $\mathbf{x}(t)$ approaches the boundary of the free space \mathcal{X}_{C_n} defined as $\partial\mathcal{X}_{C_n} = \{\mathbf{x}(t) \in \mathbb{R}^{nm} \mid \lambda_2(\mathbf{x}(t)) = 0\}$. Since $\lambda_2(\mathbf{x}(t))$ is not a differentiable function⁵, we use the fact that $\lambda_2(\mathbf{x}(t))$ is positive if and only if the determinant of the matrix $P^T L(\mathbf{x}(t))P$ is positive, which enables us to equivalently represent the set of desired states \mathcal{X}_{C_n} in (4) as,

$$\mathcal{X}_{C_n} = \{ \mathbf{x}(t) \in \mathbb{R}^{nm} \mid \det(P^T L(\mathbf{x}(t))P) > 0 \}$$

³A possible choice for P could be the matrix $I_n - \frac{1}{n} \mathbf{1}\mathbf{1}^T$ where I_n is the n -dimensional identity matrix. Observe that $\text{rank}(I_n - \frac{1}{n} \mathbf{1}\mathbf{1}^T) = n-1$ and $(I_n - \frac{1}{n} \mathbf{1}\mathbf{1}^T)\mathbf{1} = \mathbf{0}$. Hence, the columns of $I_n - \frac{1}{n} \mathbf{1}\mathbf{1}^T$ span the space $\mathbf{1}^\perp$ and we can use Gram-Schmidt orthonormalization to get P .

⁴To simplify notation, we drop the dependence of the state \mathbf{x} on time t .

⁵By Propositions 3.1 and 3.2, $\lambda_2(\mathbf{x})$ can be written as,

$$\lambda_2(\mathbf{x}) = \inf_{\mathbf{z} \in \mathbf{1}^\perp} \frac{\mathbf{z}^T L(\mathbf{x})\mathbf{z}}{\mathbf{z}^T \mathbf{z}} = \inf_{\mathbf{z} \neq \mathbf{0}} \frac{\mathbf{z}^T P^T L(\mathbf{x})P \mathbf{z}}{\mathbf{z}^T \mathbf{z}}$$

Hence, we can define a potential field on the set \mathcal{X}_{C_n} and use its negative gradient as the control input to navigate the multi-agent system away from the undesired set $\partial\mathcal{X}_{C_n}$. In particular, we have the following proposition.

Proposition 3.3: Define the potential field of the form,

$$\phi_c(\mathbf{x}(t)) = \frac{1}{(\det(P^T L(\mathbf{x}(t))P))^a} \quad (5)$$

where a is any positive constant. Then, the control $\mathbf{u}_c(t) = -\nabla_{\mathbf{x}} \phi_c(\mathbf{x}(t))$ guarantees that $\mathbf{x}(t) \in \mathcal{X}_{C_n}$ for all $t \geq 0$.

Proof: By Proposition 3.2, we have that $\det(P^T L(\mathbf{x})P) = \prod_{i=2}^n \lambda_i(\mathbf{x})$, and hence $\lambda_2(\mathbf{x}) > 0$ if and only if $\det(P^T L(\mathbf{x})P) > 0$ by positive semi-definiteness of $P^T L(\mathbf{x})P$. Clearly, $\phi_c(\partial\mathcal{X}_{C_n}) = \infty$ and $\phi_c(\mathcal{X}_{C_n}) < \infty$. Hence, $\phi_c(\mathcal{X}_{C_n}) < \phi_c(\partial\mathcal{X}_{C_n})$ and since for all $\mathbf{x} \in \mathcal{X}_{C_n}$ we have that $\dot{\phi}_c(\mathbf{x}) = -\|\nabla_{\mathbf{x}} \phi_c(\mathbf{x})\|^2 \leq 0$, we conclude that the set \mathcal{X}_{C_n} is an invariant of motion for the multi-agent system, i.e., the graph will always remain connected. ■

In other words, we treat graph connectivity as an *obstacle* in the free space and move in the direction of the negative gradient of the potential field. Note, that the controller in Proposition 3.3 guarantees that $\lambda_2(\mathbf{x}) > 0$ for all time but not that $\lambda_2(\mathbf{x})$ will also be maximized as is the case in [4], since even if $\det(P^T L(\mathbf{x})P) = \prod_{i=2}^n \lambda_i(\mathbf{x})$ increases, $\lambda_2(\mathbf{x})$ can decrease. On the other hand, observe that $\lambda_2(\mathbf{x})$ will attain its global maximum if and only if $\det(P^T L(\mathbf{x})P)$ will also attain its global maximum.⁶ However, since $L(\mathbf{x})$ is a function of the state vector \mathbf{x} through the sigmoid function (see equations 2 - 3), $\det(P^T L(\mathbf{x})P)$ is not a concave function of the state \mathbf{x} , and so only convergence to a local maximum is guaranteed. We conclude this section by providing a closed form expression for the controller in Proposition 3.3.

Proposition 3.4: The control law,

$$\mathbf{u}_c(t) = \frac{a}{(\det M(\mathbf{x}(t)))^a} \cdot \begin{bmatrix} \text{tr} \left[M^{-1}(\mathbf{x}(t)) \frac{d}{dx_1} M(\mathbf{x}(t)) \right] \\ \vdots \\ \text{tr} \left[M^{-1}(\mathbf{x}(t)) \frac{d}{dx_n} M(\mathbf{x}(t)) \right] \end{bmatrix} \quad (6)$$

where $M(\mathbf{x}) = P^T L(\mathbf{x})P$, guarantees that the multi-agent closed loop system $\dot{\mathbf{x}}(t) = \mathbf{u}_c(t)$ always remains in the set \mathcal{X}_{C_n} corresponding to connected graphs in C_n .

Proof: Let $M(\mathbf{x}) = P^T L(\mathbf{x})P$ and denote by $c_{ij}(\mathbf{x})$ the cofactor of the entry $m_{ij}(\mathbf{x})$ of the matrix $M(\mathbf{x})$. Let $C(\mathbf{x})$ denote the cofactor matrix and denote by $c_{ij}^T(\mathbf{x})$ the (i, j) -th entry of $C^T(\mathbf{x})$, i.e., $c_{ij}^T(\mathbf{x}) = c_{ji}(\mathbf{x})$. Since the determinant is a differentiable function of matrix entries, in particular it is a sum of products of entries, the chain rule gives,

$$\frac{d}{dx_k} \det M(\mathbf{x}) = \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \left(\frac{\partial}{\partial m_{ij}} \det M(\mathbf{x}) \right) \frac{d}{dx_k} m_{ij}(\mathbf{x})$$

For all $j = 1, \dots, n-1$, computation of the Laplace expansion of the determinant along the j -th column gives $\det M(\mathbf{x}) = \sum_{i=1}^{n-1} c_{ij}(\mathbf{x}) m_{ij}(\mathbf{x})$ and hence $\frac{\partial}{\partial m_{ij}} \det M(\mathbf{x}) =$

⁶Note that the eigenvalues of the Laplacian matrix are upper bounded and hence, so is the determinant.

$c_{ij}(\mathbf{x})$. Therefore,

$$\begin{aligned} \frac{d}{dx_k} \det M(\mathbf{x}) &= \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} c_{ij}(\mathbf{x}) \frac{d}{dx_k} m_{ij}(\mathbf{x}) \\ &= \sum_{j=1}^{n-1} \sum_{i=1}^{n-1} c_{ji}^T(\mathbf{x}) \frac{d}{dx_k} m_{ij}(\mathbf{x}) = \text{tr} \left[C^T(\mathbf{x}) \frac{d}{dx_k} M(\mathbf{x}) \right] \end{aligned}$$

A direct consequence of the Laplace expansion of the determinant is the identity $I \cdot \det M(\mathbf{x}) = M(\mathbf{x}) C^T(\mathbf{x})$. Proposition 3.3 guarantees that $\lambda_2(\mathbf{x}(t)) > 0$ for all time, and so $\det M(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbb{R}^{nm}$. Thus $M(\mathbf{x})$ is always positive definite, and hence invertible. Therefore, by left multiplication of the previous identity by $M^{-1}(\mathbf{x})$, we get $M^{-1}(\mathbf{x}) \cdot \det M(\mathbf{x}) = C^T(\mathbf{x})$, and substituting in the expression for $\frac{d}{dx_k} \det M(\mathbf{x})$ we get,

$$\begin{aligned} \frac{d}{dx_k} \det M(\mathbf{x}) &= \text{tr} \left[\det M(\mathbf{x}) \cdot M^{-1}(\mathbf{x}) \frac{d}{dx_k} M(\mathbf{x}) \right] \\ &= \det M(\mathbf{x}) \cdot \text{tr} \left[M^{-1}(\mathbf{x}) \frac{d}{dx_k} M(\mathbf{x}) \right] \end{aligned}$$

where $M^{-1}(\mathbf{x}) = (P^T L(\mathbf{x}) P)^{-1}$ and $\frac{d}{dx_k} M(\mathbf{x}) = P^T \frac{d}{dx_k} L(\mathbf{x}) P$, and the result follows directly from Proposition 3.3 and a simple application of the chain rule. ■

IV. CONNECTIVITY TASKS

In Section III we developed a controller based on artificial potential fields that by construction, guarantees that the network of agents remains connected for all time. Due to its feedback nature, our controller is more robust and more flexible to account for secondary objectives, in contrast to optimization based approaches [1], [4]. In particular, depending on whether the objective is ‘‘repulsive’’ (collision avoidance) or ‘‘attractive’’ (convergence to a point), it can be, in general, included in the denominator or the numerator of the modified potential function, respectively. However, any modified controller should be studied to see whether it has guaranteed properties. For particular cases, such as stationary obstacle avoidance, this could be a challenging process. In this section, we first show that our controller maintains (and in particular increases) connectivity of the multi-agent system and then consider a nontrivial connectivity task involving collision avoidance and single leader tracking objectives.

A. Maintaining Connectivity

In this connectivity task we consider a group of $n = 20$ agents (dots), that are, initially, randomly distributed in the plane, so that the initial underlying graph is connected. Whenever two agents are adjacent in the corresponding graph, an edge is drawn between them. Under the control law in equation (6), the connectivity of the multi-agent system is maintained, and in particular, is increased leading to a closed loop solution to the rendezvous problem [15], [16]. However, as discussed in Section III, the gradient descent controller (6) does not provide guarantees for *rendezvous*. These results are illustrated in Figure 2.

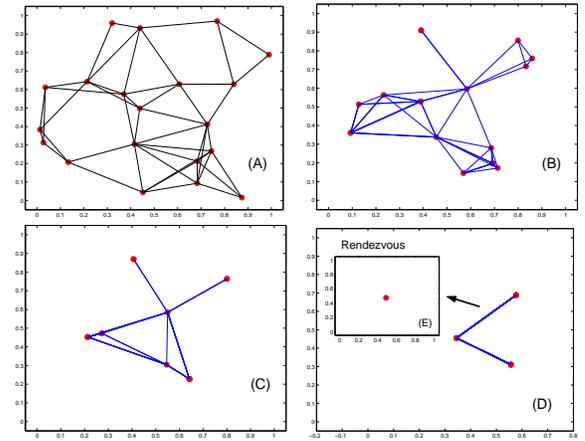


Fig. 2. 20 agents / Rendezvous.

B. Tracking a Leader with Collision Avoidance Constraints

1) *Collision Avoidance*: Many potential fields approaches have been proposed in the literature that guarantee collision avoidance for multi-agent systems [10], [11]. In this paper we choose to adopt the one introduced in [10] due to its ‘‘local’’ nature. The reason for this is because, unlike connectivity maintenance which imposes maximum distance constraints on the agents, collision avoidance imposes minimum distance constraints on the agents. Hence, we are dealing with two conflicting objectives, and in order to resolve this problem we need to make the collision avoidance objective ineffective once the agents are sufficiently far from one another. In particular, for every pair (i, j) of agents, the authors in [10], define the function (Figure 3),

$$\beta_{ij}(\mathbf{x}(t)) = \left(1 - \mu \frac{(\|x_i(t) - x_j(t)\| - d)^2}{1 + (\|x_i(t) - x_j(t)\| - d)^2} \right)^\rho$$

where $\mu = \frac{1+d^4}{d^4}$ and $\rho = \frac{1 - \text{sign}(\|x_i(t) - x_j(t)\| - d)}{2}$. This function varies in the interval $[0, 1]$, becoming 0 when the distance $\|x_i(t) - x_j(t)\|$ is 0 and attaining its maximum value 1 when $\|x_i(t) - x_j(t)\|$ is larger than d . In this way, the presence of other agents on the motion of an agent remains ‘‘local’’, within a region of radius d . Then, the net effect of all neighboring distances in the multi-agent system is captured in the function $\beta(\mathbf{x}(t)) = \prod_{i,j} \beta_{ij}(\mathbf{x}(t))$. Hence, we can define the potential field for collision avoidance $\phi_a(\mathbf{x}(t)) = \frac{1}{\beta(\mathbf{x}(t))^k}$, with $k > 0$ a constant, which composed with the connectivity potential field of Proposition 3.3 gives rise to the control law,

$$\mathbf{u}(t) = -\nabla_{\mathbf{x}} \phi(\mathbf{x}(t)) = -\nabla_{\mathbf{x}} (\phi_c(\mathbf{x}(t)) + \phi_a(\mathbf{x}(t))) \quad (7)$$

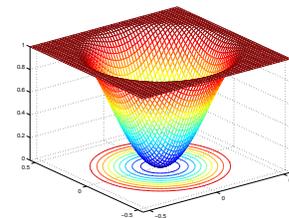


Fig. 3. Plot of the function $\beta_{ij}(\|x_i(t) - x_j(t)\|)$ for $d = .5$.

Clearly, $\phi(\mathbf{x}(t)) \rightarrow \infty$ whenever $\lambda_2(\mathbf{x}(t)) \rightarrow 0$ or $\beta(\mathbf{x}(t)) \rightarrow 0$ and thus, moving in the direction of the negative gradient of $\phi(\mathbf{x}(t))$, guarantees both that the graph will remain connected and that collisions between the agents will be avoided.

2) *Tracking a Leader*: Consider now the case where agent i is a leader for a group of n agents, with dynamics independent of the other $n - 1$ agents. We show, that under the control law (7), the agents track the leader and maintain a connected formation while avoiding collisions with each other. The following simulation demonstrates the aforementioned connectivity task. We consider a network of $n = 9$ agents with a leader labeled “L”, having dynamics of the form,

$$u_L = \begin{bmatrix} 1 + x_{L,2} \left(1 - \frac{3}{2}x_{L,2}\right) - x_{L,1}^3 \\ 1 - x_{L,1} \left(1 + x_{L,1}^2\right) - \frac{3}{2}x_{L,2}^2 \end{bmatrix}$$

The remaining 8 agents have controllers defined as in equation (7). Initially, all agents (dots) are randomly distributed in the plane so that the initial network is connected, and an edge is drawn between any two connected agents. Figures 4 and 5 illustrate our results for connectivity parameters $\epsilon = .25$ and $w = 10$ and collision avoidance radius $d = .1$. Observe that the imposed system specifications are always satisfied.

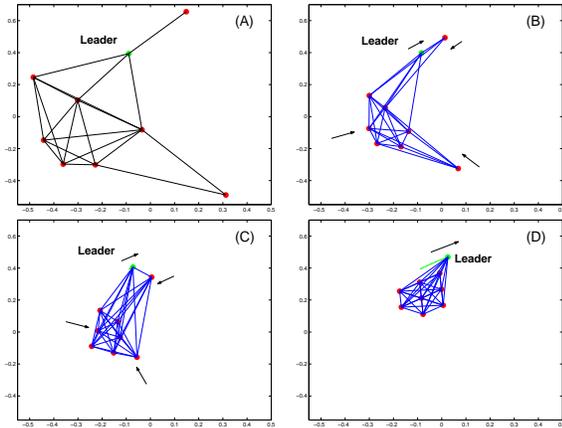


Fig. 4. One Leader and eight followers (initial phase).

V. CONCLUSIONS

In this paper we considered the problem of controlling the structure of dynamic graphs so that the resulting motion always preserves connectivity and collision avoidance properties. The idea was to define artificial potential fields where

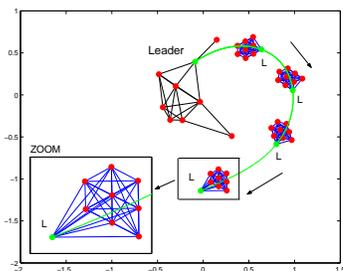


Fig. 5. One Leader (labeled “L”) and eight followers (tracking leader phase).

both objectives were modeled as obstacles in the free space and move in the direction of their negative gradient. Connectivity was captured through the smallest eigenvalue of the projected graph Laplacian matrix to the space perpendicular to the Laplacian eigenvector of ones. Compared to our previous work [1], this approach is more robust due to its feedback nature and more amenable to accounting for secondary objectives.

VI. ACKNOWLEDGMENTS

The authors would like to thank Ali Jadbabaie (University of Pennsylvania) and Maria Carmela De Gennaro (Universita del Sannio, Italy) for discussions on dynamic graphs’ connectivity.

REFERENCES

- [1] M. M. Zavlanos and G. J. Pappas. *Controlling Connectivity of Dynamic Graphs*, Proceedings of the 44th IEEE Conference on Decision and Control, pp. 6388-6393, Seville, Spain, Dec. 2005.
- [2] Demetri P. Spanos and Richard M. Murray. *Robust Connectivity of Networked Vehicles*, Proceedings of the 43rd IEEE Conference on Decision and Control, pp. 2893-2898, Bahamas, Dec. 2004.
- [3] Mehran Mesbahi. *On State-dependent Dynamic Graphs and their Controllability Properties*, IEEE Transactions on Automatic Control, vol. 50(3), pp. 387-392, March 2005.
- [4] Yoonsoo Kim and Mehran Mesbahi. *On Maximizing the Second Smallest Eigenvalue of a State-dependent Graph Laplacian*, IEEE Transactions on Automatic Control, vol. 51(1), pp. 116-120, Jan. 2006.
- [5] A. Jadbabaie, J. Lin and A. S. Morse. *Coordination of Groups of Mobile Autonomous Agents using Nearest Neighbor Rules*, IEEE Transactions on Automatic Control, vol. 48(6), pp. 988-1001, 2003.
- [6] R. Olfati-Saber and R. M. Murray. *Consensus Problems in Networks of Agents with Switching Topology and Time-Delays*, IEEE Transactions on Automatic Control, vol. 49, pp. 1520-1533, Sep. 2004.
- [7] H. Tanner, A. Jadbabaie and G. Pappas. *Flocking in Fixed and Switching Networks*, IEEE Transactions on Automatic Control, April 2005. Submitted.
- [8] J. P. Desai, J. P. Ostrowski and V. Kumar. *Modeling and Control of Formations of Nonholonomic Mobile Robots*, IEEE Transactions on Robotics and Automation, vol. 17(6), pp. 905-908, 2001.
- [9] P. Ögren, M. Egerstadt and X. Hu. *A Control Lyapunov Function Approach to Multi-agent Coordination*, IEEE Transactions on Robotics and Automation, vol. 18(5), pp. 847-851, Oct. 2002.
- [10] Amit Kumar and Herbert G. Tanner. *Decentralized Formation Stabilization and Collision Avoidance Using Navigation Functions*, IEEE Transactions on Robotics. Submitted.
- [11] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos and M. M. Zavlanos. *A Feedback Stabilization and Collision Avoidance Scheme for Multiple Independent Non-point Agents*, Automatica, vol. 42(2), pp. 229-243, Feb. 2006.
- [12] R. Sepulchre, D. Paley, N. E. Leonard. *Stabilization of Planar Collective Motion: All-to-All Communication*, IEEE Transactions on Automatic Control, 2006. To appear.
- [13] G. Lafferriere, A. Williams, J. Caughman and J.J.P. Veerman. *Decentralized Control of Vehicle Formations*, Systems and Control Letters, vol. 54(9), pp. 899-910, Sept. 2005.
- [14] T. Balch and R.C. Arkin. *Behavior-based Formation Control for Multi-robot Teams*, IEEE Transactions on Robotics and Automation, vol. 14(6), pp. 926-939, Dec. 1998.
- [15] J. Lin, A.S. Morse and B.D.O. Anderson. *The Multi-Agent Rendezvous Problem*, Proceedings of the 42nd IEEE Conference on Decision and Control, pp. 1508-1513, Maui, Hawaii, Dec. 2003.
- [16] J. Cortes, S. Martinez and F. Bullo. *Robust Rendezvous for Mobile Autonomous Agents via Proximity Graphs in Arbitrary Dimensions*, IEEE Transactions on Automatic Control, vol. 51(8), pp. 1289-1298, 2006.
- [17] W. Ren and R. Beard. *Consensus of Information under Dynamically changing Interaction Topologies*, Proceedings of the American Control Conference, pp. 4939-4944, June 2004.
- [18] Sameera Poduri and Gaurav S. Sukhatme. *Constrained Coverage for Mobile Sensor Networks*, In IEEE International Conference on Robotics and Automation, pp. 165-172, New Orleans, LA, May 2004.
- [19] Norman Biggs, *Algebraic Graph Theory*, Cambridge University Press, 1993.