2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Multi-Agent Coordination under Connectivity Constraints

Kunal Srivastava and Mark W. Spong

Abstract— **We study the problem of motion coordination of multiple agents, under a graph connectivity constraint. We develop a framework for the problem based on partial ordering of graphs and embedding of constraint sets and identify a certain convexity property of the constraint set induced by the graph connectivity constraint. This property is used to solve two instances of the coordination problem. In the first instance, the agents are required to converge from an arbitrary position to a formation characterized by its adjacency matrix. In the second instance, we study the specific problem of moving from one formation to another formation so that the formation graph remains connected at all times. We show that both these problems can be reduced to static convex optimization problems. Existence and uniqueness of solutions are also investigated.**

I. Introduction

The problem of motion coordination of multiple robotic agents has been studied in a number of settings. The term 'coordination' loosely refers to a global criterion that needs to be met by the robotic agents. In all of these cases the coordination task can be represented as a function of the state of the agents. In certain problems the coordination task is only dependent on the final states of the agents at the end of a finite or infinite time horizon. Let us call such problems as belonging to class I. The problem of finite time or asymptotic rendezvous falls into this class. In these problems the transient behavior of states is of little importance, unless an additional criterion such as collision avoidance is taken into account. In another class (class II), the coordination task is dependent on the state trajectory over a time horizon. An example for this case is the area coverage problem. For a third class of problems (class III) the coordination criterion needs to be met at each point in time. Problems in this class arise mainly because of the presence of constraints in the state space, which need to be satisfied at all times. We are concerned with a specific kind of constraint that arises due to the limited sensing and communication range of the wireless sensors and controllers. Note that these classes are not mutually exclusive and depend mainly on problem formulation. Typically, by choosing suitable states the problems can be reformulated to lie in a different class. An example is the Lagrange formulation of the optimal control problem, where by augmenting the state vector the problem can be reformulated only in terms of the final state. This formulation of the optimal control problem is said to be in Mayer form [16].

This research was partially supported by NSF Grant ECCS-0725433 and ONR Grant N-0001-4-05-10186.

Kunal Srivastava and Mark Spong are with Coordinated Science Laboratory, University of Illinois, 1308 W. Main St., Urbana, IL 61801 USA {kkunal2,mspong}@uiuc.edu

The main object of our study is the graph formed by the agents using nearest neighbor rules. By definition, such a graph is dependent on the state of the nodes which in this case is the position of the agents. Such graphs owing to their changing topology have been termed as dynamic graphs in the literature. In many multi vehicle coordination examples it is undesirable for the formation to break into subparts as this implies a loss of communication. This requirement can be modeled as a connectivity constraint on the graph formed with the nodes as the agents. This condition induces a constraint on the state space which needs to be satisfied at all times.

The remainder of this paper is organized as follows. In section II we place the current work in the context of previous research in related areas, and provide a brief literature survey. In section III we state and discuss some definition and preliminaries required for analysis. In section IV we state the main problems and derive the related results to solve these. Finally in section V we provide the conclusion and future directions for research.

II. Related Work

Dynamic graphs and its connectivity properties have been studied in a variety of contexts. Mesbahi in [6] studies state dependent graphs and defines a notion of controllability for such graphs. Further in [7] Kim and Mesbahi provide an algorithm for maximizing the second smallest eigenvalue of the graph Laplacian. This is akin to maximizing the connectivity of the graph. In [5] the authors provide a decentralized scheme for the same problem. Šiljak [8] studied dynamic graphs in relation to the structural properties of interconnected dynamical systems. He defined notions of input reachability, structural controllability and observability. He also studied the notion of connective stability, which is defined as the Lyapunov stability under structural perturbations. A closely related idea was studied by Erdös and Rényi [12], where they considered the probability of a graph to remain connected under random dropout of vertices and edges. Recently a number of researchers, while investigating problems in multi agent rendezvous and state agreement, have proved the requirement of graph connectivity as a necessary condition. In [1] and [3], this condition arises as a necessity for the proof of rendezvous. Some of the consensus protocols assume that the connectivity is maintained at all times [13], and in some protocols a relaxed condition of graph connectivity infinitely number of times is assumed [14], [15]. A number of authors have developed control laws for the connectivity maintenance problem for multiagent systems [3], [4]. In [3] the main focus of the paper

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

is to facilitate rendezvous while preserving connectedness. In [4] the authors develop a constraint on control actions which preserve connectivity. This constraint was enforced by solving an optimization problem at each time step.

Our approach differs from the earlier approaches to the problem in the sense that we provide an analysis of the constrained state space arising from the connectivity constraint. We also show how we can use this structure to come up with solutions which are optimal in some sense. The mathematical framework developed can be useful in addressing problems concerned with dynamically changing graph topologies.

III. Definitions and Preliminaries

We will first state some terms from graph theory which are of relevance here. For more details refer any of the standard texts like [9]. A graph G as an abstract object is modeled as the pair $V(G), E(G)$ where $V(G)$ denotes the vertex set and $E(G) \subseteq V(G) \times V(G)$ denotes the set of edges. Two vertices $x, y \in V(G)$ are said to be *adjacent* if an edge exists between them, i.e, $(x, y) \in E(G)$. A graph is called *complete* if every pair of vertices are adjacent. The complete graph on N vertices is denoted by K_N. A *subgraph* of a graph X is a graph Y such that $V(Y) \subseteq V(X)$ and $E(Y) \subseteq E(X)$. If $V(Y) = V(X)$, Y is called a *spanning subgraph*. A *cycle* is a connected graph where every vertex has exactly two neighbors. A connected acyclic graph is called a *tree* and a spanning subgraph with no cycles is called a *spanning tree*. The *adjacency matrix* of a graph is defined such that the i, j entry is either 1 or 0, if i, j are adjacent vertices or not respectively. We will be dealing with graphs with no self loops hence the diagonal entry is typically 0.

Let us consider a group of N agents. The position of the i^{th} agent in a global reference frame is denoted as the column vector $x_i \in \mathbb{R}^2$. We define the state vector as the concatenation $\mathbf{x} = (x_1^T, x_2^T, \ldots, x_N^T)^T$. We can identify each agent as a vertex of a graph G. Thus the vertex set consists of the indices $V(G) = \{1, \ldots, N\}$. Let us define the set $\mathcal{C}(R) \subset \mathbb{R}^4$ as $\mathcal{C}(R) = \{(x_i, x_j) | x_i, x_j \in \mathbb{R}^2 \quad \text{and} \quad \|x_i - x_j\| \leq R\}$. Since R is constant throughout the paper, we remove the dependence on R from the notation and refer to $\mathcal{C}(R)$ as simply $\mathcal{C}$. The characteristic function of the set is denoted as $\chi_{\mathcal{C}}$ and is defined as

$$\chi_{\mathcal{C}}(x_i, x_j) = \begin{cases} 1 & (x_i, x_j) \in \mathcal{C} \\ 0 & (x_i, x_j) \notin \mathcal{C} \end{cases}$$

We say that the agents i, j are adjacent or connected when $\chi_{\mathcal{C}}(x_i, x_j) = 1$. Thus the edge set of the graph G is $E(G) = \{(i, j) | \chi_{\mathcal{C}}(x_i, x_j) = 1\}$. Let us denote as $\mathcal{M}$ the set of $N \times N$ symmetric matrices so that each entry is either 0 or 1 and the diagonal terms are zero. This gives us a way to define the adjacency matrix as a function that maps the state space to the set $\mathcal{M}$.

$$\mathbf{A} : \mathbb{R}^{2N} \rightarrow \mathcal{M}$$

This is done as follows:

$$\mathbf{A}(\mathbf{x})_{ij} = \chi_{\mathcal{C}}(x_i, x_j) - \delta_{ij}$$

Where $\delta_{ij} = 1 \quad \text{if} \quad i = j$ and $\delta_{ij} = 0 \quad \text{if} \quad i \neq j$. Here we have subtracted 1 from the diagonal terms to force the diagonal terms to be zero. Once we fix the indices for the vertices, a graph can be completely described in terms of the adjacency matrix. Thus, we will use the term graph and adjacency matrix interchangeably.

Consider an equation of the form $\mathbf{A}(\mathbf{x}) = X$, where $X \in \mathcal{M}$. Checking if this equation admits a solution is equivalent to checking the feasibility of a system of inequalities. There are efficient algorithms available for this task [10], [11]. In case this equation has a solution then the graph X is said to be realizable. The feasibility problem can be cast as an LMI problem utilizing the quadratic structure of the norm and the $S-$procedure [2]. It is clear to see that the function $\mathbf{A}(\mathbf{x})$ is invariant under translation and rotation. This reflects in the fact that if $\mathbf{x}$ solves the above equation then so does $\mathbf{x} + \alpha\mathbf{1}$, where $\mathbf{1}$ denotes a vector of compatible dimension with each entry 1.

We define a relation " $\preccurlyeq_{\mathcal{M}}$ " on $\mathcal{M}$ as follows. Given $X, Y \in \mathcal{M}$ we will say $X \preccurlyeq_{\mathcal{M}} Y$ if $X_{ij} = 1 \Rightarrow Y_{ij} = 1$. In other words $X \preccurlyeq_{\mathcal{M}} Y$ if the graph represented by X is a spanning subgraph of the graph represented by Y. It is easily verifiable that $\preccurlyeq_{\mathcal{M}}$ indeed defines a relation. In fact this induces a partial order on the set $\mathcal{M}$. Given $X \in \mathcal{M}$ we define

$$\mathcal{C}_X = \{\mathbf{x} \in \mathbb{R}^{2N} | X \preccurlyeq_{\mathcal{M}} \mathbf{A}(\mathbf{x})\}$$

Thus $\mathcal{C}_X$ denotes the set of states $\mathbf{x}$ whose graphical representation preserves those edges present in X. It is worth pointing out that for $X \preccurlyeq_{\mathcal{M}} Y$ we have $\mathcal{C}_Y \subseteq \mathcal{C}_X$. This is a reiteration of the fact that as we impose more constraints on the state space, the feasible states satisfying the constraints shrink. This can be stated in more general terms as follows: Consider a function $f : X \rightarrow Y$ where X and Y are arbitrary sets. If there exists an ordering $\preccurlyeq_Y$ on Y then f induces an ordering on X as follows: $x \preccurlyeq_X y$ for $x, y \in X$ if $f(x) \preccurlyeq_Y f(y)$. An example is a Lyapunov function $V : \mathbb{R}^n \rightarrow \mathbb{R}$. Since $\mathbb{R}$ is totally ordered with the usual ordering we have the induced ordering on $\mathbb{R}^n$ given in terms of the nested level sets of V. Coming back to our original problem we have that under this ordering there is a maximum element in $\mathcal{M}$, denoted as $\mathbf{K}_N$. By definition $X \preccurlyeq_{\mathcal{M}} \mathbf{K}_N \quad \forall X \in \mathcal{M}$. $\mathbf{K}_N$ represents the complete graph on N vertices and clearly denotes the case when all agents can communicate with each other and hence $\mathbf{A}(\mathbf{x})_{ij} = 1 \quad \forall \mathbf{x} \in \mathcal{C}_{\mathbf{K}_N} \quad i \neq j$. Note that we could have ordered the elements of $\mathcal{M}$ in terms of set inclusion of the corresponding constraint set induced by the elements, i.e $X \preccurlyeq_{\mathcal{M}} Y$ if $\mathcal{C}_X \subseteq \mathcal{C}_Y$. In this ordering $\mathbf{K}_N$ would be the minimal element.

We will say that the graph $\mathcal{G}$ is connected if there is a sequence of vertices $i = k_0, k_1, k_2, \ldots, k_n = j$ between each agent i and j such that $\mathbf{A}(\mathbf{x})_{k_l k_{l+1}} = 1 \quad \forall l = 0, \ldots, n-1$. The set of adjacency matrices which represent connected graphs are a strict subset of $\mathcal{M}$. We will denote as $\mathcal{M}_c$ the set of all such adjacency matrices. Given a matrix $X \in \mathcal{M}$ let us define an auxiliary matrix $S_X = \sum_{k=0}^{N-1} X^k$. This matrix has a useful interpretation. The i, j entry of the matrix S_X gives the sum of the number of $k-$ hop paths, where $k = \{0, \dots, N-1\}$, between i and j. Thus we have that $X \in \mathcal{M}_c$ if and only if all entries of the matrix S_X are nonzero. The connectivity requirement imposes a constraint on the state space. The feasible set can be represented as

$$\Omega = \{\mathbf{x} \in \mathbb{R}^{2N} | \mathbf{A}(\mathbf{x}) \in \mathcal{M}_c\}$$

or equivalently,

$$\Omega = \bigcup_k \mathcal{C}_{A_k}, \quad \text{where} \quad A_k \in \mathcal{M}_c, \quad \forall k$$

We can restrict the partial order $\preccurlyeq_{\mathcal{M}}$ introduced earlier to $\mathcal{M}_c$. We will drop the subscript and denote the restriction on $\mathcal{M}_c$ by $\preccurlyeq$. In this set we have the minimal elements, which are a subset of the class of spanning trees involving N nodes. This is due to the fact that not all spanning trees are necessarily realizable in the sense described earlier. In fact, in [2] the authors have shown that the set of realizable graphs are a strict subset of the class of graphs on N nodes when $N \geq 5$. This is why the minimal elements which maintain connectivity are a strict subset of the spanning trees when $N \geq 5$. Note that the maximal element $\mathbf{K}_N$ introduced earlier also belongs to $\mathcal{M}_c$. The constraint set induced by $\mathbf{K}_N$ has a useful representation as follows.

$$\mathcal{C}_{\mathbf{K}_N} = \bigcap_k \mathcal{C}_{A_k}, \quad \text{where} \quad A_k \in \mathcal{M}_c, \quad \forall k$$

Given a set of matrices $X_0, X_1, \dots, X_n$ in $\mathcal{M}_c$. We say that these matrices form a chain, if for any X_i and X_j we have either $X_i \preccurlyeq X_j$ or $X_j \preccurlyeq X_i$. In other words, the partial ordering when restricted to a chain is a total ordering. We will label the minimal elements of the set $\mathcal{M}_c$ as Θ_k. As mentioned earlier these are the adjacency matrices representing spanning trees. Given any $X \in \mathcal{M}_c$ we always have at least one minimal element Θ_k for which $\Theta_k \preccurlyeq X$. It is of course possible that we have more than one minimal element for a given $X \in \mathcal{M}_c$. In fact for all X which are not trivially minimal elements themselves we have more than one minimal element satisfying the earlier condition.

IV. Problem Formulation

In this section we formulate and state our main results. The following lemma is crucial in what follows.

Lemma 4.1: The set $\mathcal{C}_X$ is closed and convex for each $X \in \mathcal{M}_c$.

Proof: It follows from the definition of $\mathcal{C}_X$ that if $\mathbf{x} = (x_1^T, x_2^T, \dots, x_N^T)^T \in \mathcal{C}_X$ and $\mathbf{y} = (y_1^T, y_2^T, \dots, y_N^T)^T \in \mathcal{C}_X$ then $\forall i, j$ such that $X_{ij} = 1$, we have $\|x_i - x_j\| \leq R$ and $\|y_i - y_j\| \leq R$. Hence for any $\mathbf{z} = \alpha\mathbf{x} + (1-\alpha)\mathbf{y}$ such that $\alpha \in [0, 1]$ we have

$$\begin{aligned} \|z_i - z_j\| &= \|\alpha(x_i - x_j) + (1-\alpha)(y_i - y_j)\| \\ &\leq \alpha\|x_i - x_j\| + (1-\alpha)\|y_i - y_j\| \\ &\leq \alpha R + (1-\alpha)R = R \end{aligned}$$

which implies $\mathbf{z} \in \mathcal{C}_X$. This proves convexity. The closedness follows from the fact that we do not have strict inequalities. ■

Remark 4.2: A point to note is that this result essentially follows from the convexity of the sublevel sets of the $\|\cdot\|$ function.

We now have the framework to undertake the first problem. Consider the following scenario. Let us assume that we have N agents which are distributed arbitrarily. We wish to compute terminal positions for each agents so that their final configuration corresponds to having a desired adjacency matrix as a subgraph. In doing so we also wish to minimize the cost function given by $\|\mathbf{x}_0 - \mathbf{x}_f\|$ where $\mathbf{x}_0$ is the initial position and $\mathbf{x}_f$ is the final state vector. The choice of this cost function is equivalent to minimizing the distance traveled by the agents to reach the final state and hence the energy expenditure. We will now show that this problem can be formulated as a convex optimization problem and admits a unique solution. The problem can be formulated as follows.

A. Problem I

Given an arbitrary initial condition $\mathbf{x}_0$ and an adjacency matrix $X \in \mathcal{M}_c$ we wish to solve the following optimization problem.

$$\min_{\mathbf{x}_f \in \mathcal{C}_X} \|\mathbf{x}_0 - \mathbf{x}_f\|$$

The convexity of the constraint set $\mathcal{C}_X$ and the cost function makes this a convex optimization problem. This is equivalent to the problem of determining the minimum distance to a closed convex set and hence admits a unique solution [17]. This problem differs from the more often used approach in formation stabilization where the desired formation is generally specified by the desired relative spacing between nodes. In our case the formation is specified in terms of the desired adjacency matrix. However because of the way we have defined the constraint set $\mathcal{C}_X$ the final configuration might have more connected vertices than specified by X. In other words, we will have $X \preccurlyeq A(\mathbf{x}_f)$ where $\mathbf{x}_f$ is the solution of the above optimization problem.

1) Example: The simulation result in figure1 illustrates the case when N=6. The initial conditions are randomly generated and the desired configuration is defined in terms of the adjacency matrix

$$X = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Problem I illustrates a straightforward application borne out of the convexity of the constraint sets. This problem can be seen as a constrained facility location problem, where the initial position of the nodes are the set points and the final points or the location of the facilities need to chosen so as to minimize a convex cost.

We will now consider the problem where unlike problem I the final position is completely specified, and the objective is to design manoeuvres which will maintain graph connectivity.

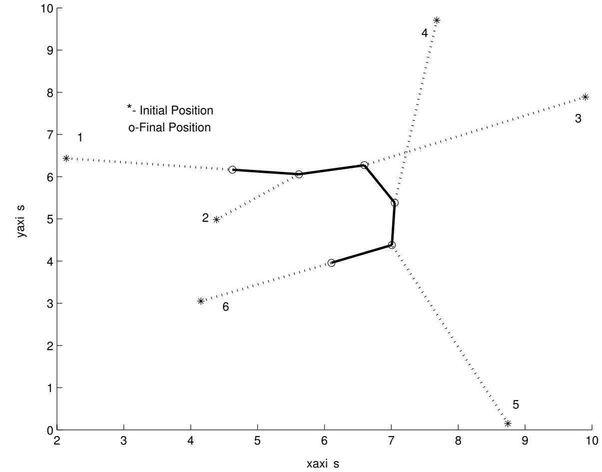

Fig. 1. Convergence to a desired adjacency matrix

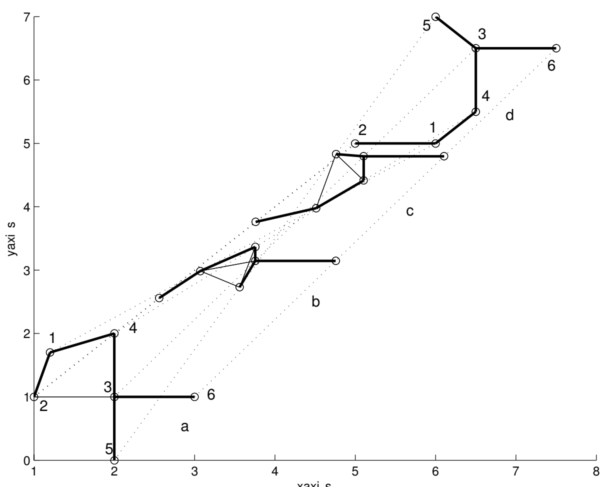

Fig. 2. Illustration of Proposition 4.3

B. Problem II

Given $\mathbf{x}_0 \in \Omega$ and $\mathbf{x}_f \in \Omega$ we wish to find the shortest path $\mathbf{x}(t)$, so that $\mathbf{x}(0) = \mathbf{x}_0$, $\mathbf{x}(T) = \mathbf{x}_f$ for some T and $\mathbf{x}(t) \in \Omega \quad \forall 0 \leq t \leq T$.

In this problem the shortest path is evaluated under the Euclidean norm in $\mathbb{R}^{2N}$. It is clear that under the Euclidean l_2 norm the shortest path between points x_i and x_f is the straight line joining them. Another important property of this shortest path is that if x_i and x_f belong to a convex set, then all the points on the straight line joining them belong to this set. The problem would be trivial if the constraint set Ω was convex; however, this is not the case. As can be seen from the definition of Ω, it is a union of convex sets and hence not necessarily convex itself. Thus, this is in general a nonconvex optimization problem. Another issue is that the constraint set is not defined as a set of inequalities. A complete characterization of the constraint set would require a complete enumeration of the graphs formed by N nodes and hence makes this a hard problem to solve. We will show that under some relaxation of optimality criterion we can formulate this as a convex optimization problem. The following proposition states the condition under which we can find a convex subset of the constraint set Ω containing the initial and final configuration. As mentioned earlier if this is the case then the straight line joining the initial and final configurations is optimal.

Proposition 4.3: If there exists a spanning subgraph Θ such that $\Theta \preccurlyeq \mathbf{A}(\mathbf{x}_0)$ and $\Theta \preccurlyeq \mathbf{A}(\mathbf{x}_f)$ then $\mathbf{x}(t) = (1-t)\mathbf{x}_0 + t\mathbf{x}_f$ where, $t \in [0,1]$; solves Problem II.

Proof: It follows from the definition of $\preccurlyeq$ that the condition $\Theta \preccurlyeq \mathbf{A}(\mathbf{x}_0)$ and $\Theta \preccurlyeq \mathbf{A}(\mathbf{x}_f)$ imply that $\mathbf{x}_0 \in \mathcal{C}_\Theta$ and $\mathbf{x}_f \in \mathcal{C}_\Theta$. From Lemma 4.1 we have that $\mathcal{C}_\Theta$ is convex and hence $\mathbf{x}(t) \in \mathcal{C}_\Theta \subset \Omega$. The optimality follows from the fact that given two points, the straight line joining them is shortest under the euclidian norm. ■

Corollary 4.4: If we have $\mathbf{A}(\mathbf{x}_0) \preccurlyeq \mathbf{A}(\mathbf{x}_f)$ or $\mathbf{A}(\mathbf{x}_f) \preccurlyeq \mathbf{A}(\mathbf{x}_0)$ then $\mathbf{x}(t) = (1-t)\mathbf{x}_0 + t\mathbf{x}_f$, for $t \in [0,1]$ solves Problem II.

Proof: This follows in a straightforward way from proposition 4.3. If $\mathbf{A}(\mathbf{x}_0) \preccurlyeq \mathbf{A}(\mathbf{x}_f)$ then for all $\Theta \preccurlyeq \mathbf{A}(\mathbf{x}_0)$ we also have $\Theta \preccurlyeq \mathbf{A}(\mathbf{x}_f)$, similarly for the reverse case. ■

Though Proposition 4.3 provides a more general result, it might be computationally challenging to verify the conditions under which it holds. The Corollary 4.4 provides a condition which is easier to verify but holds only for a subclass of cases where Proposition 4.3 holds. We now provide an algebraic criterion to verify if the conditions for the Proposition 4.3 are met or not. We define the matrix

$$[\mathbf{A}(\mathbf{x}_0) \wedge \mathbf{A}(\mathbf{x}_f)]_{ij} := \mathbf{A}(\mathbf{x}_0)_{ij}\mathbf{A}(\mathbf{x}_f)_{ij}$$

This matrix essentially represents the edges which are common in both $\mathbf{A}(\mathbf{x}_0)$ and $\mathbf{A}(\mathbf{x}_f)$. It has the important property that $\mathbf{A}(\mathbf{x}_0) \wedge \mathbf{A}(\mathbf{x}_f) \preccurlyeq_{\mathcal{M}} \mathbf{A}(\mathbf{x}_0)$ and $\mathbf{A}(\mathbf{x}_0) \wedge \mathbf{A}(\mathbf{x}_f) \preccurlyeq_{\mathcal{M}} \mathbf{A}(\mathbf{x}_f)$. Thus, if $\mathbf{A}(\mathbf{x}_0) \wedge \mathbf{A}(\mathbf{x}_f) \in \mathcal{M}_c$, then we have $\forall \Theta \preccurlyeq \mathbf{A}(\mathbf{x}_0) \wedge \mathbf{A}(\mathbf{x}_f)$, $\Theta \preccurlyeq \mathbf{A}(\mathbf{x}_0)$ and $\Theta \preccurlyeq \mathbf{A}(\mathbf{x}_f)$. Thus the problem reduces to verifying the condition $\mathbf{A}(\mathbf{x}_0) \wedge \mathbf{A}(\mathbf{x}_f) \in \mathcal{M}_c$. This can be done algebraically by checking if all the elements of the corresponding $S_{\mathbf{A}(\mathbf{x}_0) \wedge \mathbf{A}(\mathbf{x}_f)}$ defined earlier are nonzero, or studying the spectral properties of the associated Laplacian matrix [9]. This involves verifying if the second smallest eigenvalue of the Laplacian matrix corresponding to the graph represented by $\mathbf{A}(\mathbf{x}_0) \wedge \mathbf{A}(\mathbf{x}_f)$ is nonzero.

1) Example: The simulation result in Figure 2 illustrate the result in Proposition 4.3. The initial and final desired configuration are marked a) and d), respectively. The graph representing the common minimal element between initial and final configurations Θ is represented by bold lines. In the example considered we have $\Theta = \mathbf{A}(\mathbf{x}_0) \wedge \mathbf{A}(\mathbf{x}_f)$. It can be seen that this bold graph is preserved at intermediate configurations b) and c).

Proposition 4.3 provides us a condition under which the optimal solution to Problem II can easily be computed. It remains to be shown that there exists a feasible solution to Problem II even when the conditions of Proposition 4.3 are not met i.e $\mathbf{A}(\mathbf{x}_0) \wedge \mathbf{A}(\mathbf{x}_f) \notin \mathcal{M}_c$. This case can be illustrated via the following example. Let $\mathbf{A}(\mathbf{x}_0) = \Theta_k$ and $\mathbf{A}(\mathbf{x}_f) = \Theta_l$ and $\Theta_k \neq \Theta_l$ where the equality is evaluated term by term. In this case $\mathbf{A}(\mathbf{x}_0) \wedge \mathbf{A}(\mathbf{x}_f) = \mathbf{0} \notin \mathcal{M}_c$. The following Proposition proves the feasibility of problem II.

Proposition 4.5: Given any $\mathbf{x}_0$ and $\mathbf{x}_f$ in Ω there exists a $\mathbf{x}(t)$ connecting $\mathbf{x}_0$ and $\mathbf{x}_f$ such that $\mathbf{x}(t) \in \Omega$.

Proof: This can be seen as a consequence of the fact that $\mathcal{C}_{\mathbf{K}_N} \subseteq \mathcal{C}_X, \quad \forall X \in \mathcal{M}_c$. In other words there always exists a path $\mathbf{x}(t)$ from each state $\mathbf{x}_0 \in \mathcal{C}_{X_0}$ to each state $\mathbf{x}_s \in \mathcal{C}_{\mathbf{K}_N}$ so that $\mathbf{x}(t) \in \mathcal{C}_{X_0}$. By concatenating two segments from $\mathbf{x}_0$ to $\mathbf{x}_s$ and $\mathbf{x}_f$ to $\mathbf{x}_s$. We have a path joining $\mathbf{x}_0$ and $\mathbf{x}_f$, which satisfies the constraint. ■

The previous Proposition provides a suboptimal solution to Problem II. However, the solution it provides is highly inefficient as it requires the agents to converge to a state which corresponds to a fully connected graph. Such a manoeuvre is not always desirable and certainly not optimal. Consider a scenario where the agents are distributed on a large geographical area and we want to swap the positions of two agents. In this case it doesn't make sense to converge to a configuration where we have full connectivity. Hence, it makes sense to search for ways to exploit the structure of the constraint set more effectively. Our approach will be a heuristic approach in the sense that it does not guarantee a globally optimal solution.
The following formulation explicitly considers the case when the condition of Proposition 4.3 are not met.

C. Problem III

Given $\mathbf{x}_0, \mathbf{x}_f \in \Omega$ such that $\mathbf{A}(\mathbf{x}_0) \wedge \mathbf{A}(\mathbf{x}_f) \notin \mathcal{M}_c$ and $X \in \mathcal{M}_C$ such that $\mathbf{A}(\mathbf{x}_0) \preccurlyeq X$ and $\mathbf{A}(\mathbf{x}_f) \preccurlyeq X$ we wish to solve the following problem.

$$\min_{\mathbf{x} \in \mathcal{C}_X} \|\mathbf{x} - \mathbf{x}_0\| + \|\mathbf{x} - \mathbf{x}_f\|$$

Let us denote the cost function as $f(\mathbf{x})$. A solution to Problem III ,also provides a suboptimal solution to problem II. Again, as we did earlier we evaluate distance in terms of the l_2 norm. Since, we have $\mathbf{A}(\mathbf{x}_0) \preccurlyeq X$ and $\mathbf{A}(\mathbf{x}_f) \preccurlyeq X$, this implies that $\mathcal{C}_X \subseteq \mathcal{C}_{\mathbf{A}(\mathbf{x}_0)} \cap \mathcal{C}_{\mathbf{A}(\mathbf{x}_f)}$. Thus, the concatenation of segments joining $\mathbf{x}_0 \to \mathbf{x}$ and $\mathbf{x} \to \mathbf{x}_f$, gives us a suboptimal path between the initial and final position which satisfies the constraint. Note that in proving Proposition 4.5 the choice of X we used is $\mathbf{K}_N$. Let us define $[\mathbf{A}(\mathbf{x}_0) \vee \mathbf{A}(\mathbf{x}_f)]_{ij} := \mathbf{A}(\mathbf{x}_0)_{ij} \circ \mathbf{A}(\mathbf{x}_f)_{ij}$, where $\circ$ signifies the logical "or" operation. This matrix has the property that $\mathbf{A}(\mathbf{x}_0) \preccurlyeq \mathbf{A}(\mathbf{x}_0) \vee \mathbf{A}(\mathbf{x}_f)$ and $\mathbf{A}(\mathbf{x}_f) \preccurlyeq \mathbf{A}(\mathbf{x}_0) \vee \mathbf{A}(\mathbf{x}_f)$. Thus we can pick X as $X = \mathbf{A}(\mathbf{x}_0) \vee \mathbf{A}(\mathbf{x}_f)$. This choice is essentially a least upper bound on the set consisting of the elements $\mathbf{A}(\mathbf{x}_0)$ and $\mathbf{A}(\mathbf{x}_f)$. Typically the choice of X plays a big factor in the kind of solution achieved, and hence it can be chosen to satisfy any specified criterion. However, the choice of X is not completely independent as it needs to satisfy the condition $\mathbf{A}(\mathbf{x}_0) \preccurlyeq X$ and $\mathbf{A}(\mathbf{x}_f) \preccurlyeq X$. The following lemma provides condition on the uniqueness of solution to Problem III.

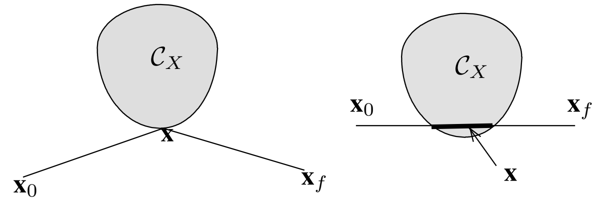

Fig. 3. Illustration for lemma 4.6

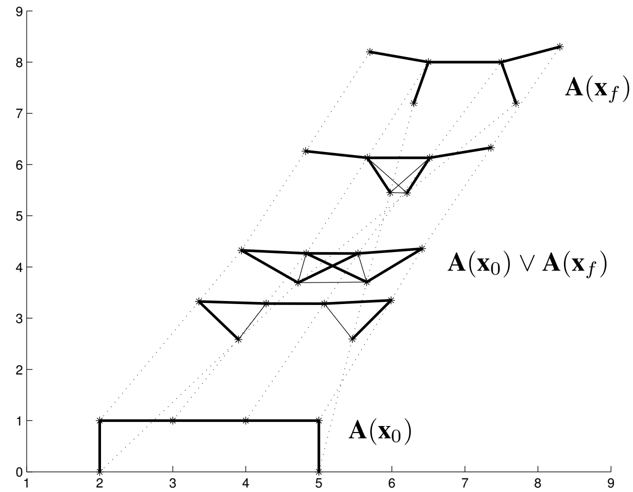

Fig. 4. Illustration for problem III

Lemma 4.6: The solution to problem III is unique if and only if we have $\forall \alpha \in [0, 1], \quad (1-\alpha)\mathbf{x}_0 + \alpha\mathbf{x}_f \notin \mathcal{C}_X$, otherwise all such $\mathbf{x} = (1-\alpha)\mathbf{x}_0 + \alpha\mathbf{x}_f \in \mathcal{C}_X$ solve Problem III.

Proof: Figure 3 illustrates the idea behind Lemma 4.6. It is straightforward to note that the cost function is constant for all $\mathbf{x}(\alpha) = (1-\alpha)\mathbf{x}_0 + \alpha\mathbf{x}_f$ where $\alpha \in [0, 1]$. Also all such $\mathbf{x}(\alpha)$ is the solution of the unconstrained minimization problem. Hence if for some α we have $\mathbf{x}(\alpha) \in \mathcal{C}_X$ then all such $\mathbf{x}(\alpha)$ are the minimizers. To prove the converse assume that $\mathbf{x}(\alpha) \notin \mathcal{C}_X$. Also assume that the solution is not unique, i.e $\exists \mathbf{x}, \mathbf{y} \in \mathcal{C}_X$ such that

$$f(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_0\| + \|\mathbf{x} - \mathbf{x}_f\| = \|\mathbf{y} - \mathbf{x}_0\| + \|\mathbf{y} - \mathbf{x}_f\| = c$$

Then it is a consequence of the triangle inequality that for all $\bar{\mathbf{x}}(\alpha) = (1-\alpha)\mathbf{x} + \alpha\mathbf{y}$, we have $f(\bar{\mathbf{x}}(\alpha)) \leq c$. Since c is the minimum, we have $f(\bar{\mathbf{x}}(\alpha)) = c$. However, since $\mathbf{x}(\alpha)$ is the only linear level set, we have a contradiction. ■
It is possible to further weaken the requirement on the choice of X in Problem III. In loose terms, it can be said that the larger the set X is, the better solution we achieve, i.e if for X_1 and X_2 we have $\mathcal{C}_{X_2} \subseteq \mathcal{C}_{X_1}$, then the corresponding solutions $\mathbf{x}_1^*$, $\mathbf{x}_2^*$ to Problem III satisfy $f(\mathbf{x}_1^*) \leq f(\mathbf{x}_2^*)$. Thus, our objective reduces to finding a 'large' constraint set $\mathcal{C}_X$. This is equivalent to finding a satisfactory X with least

number of non-zero elements. Since more non-zero elements imply the presence of more constraints which needs to be satisfied and hence reduce the search space. The following approach gives us a heuristic way to come up with such a set $\mathcal{C}_X$, and also provides a way as to weaken the requirement on X as required in the statement of Problem III.

Given $\mathbf{x}_0$ and $\mathbf{x}_f$ let $\{\Theta_0^1, \Theta_0^2, \ldots, \Theta_0^{k_1}\}$ and $\{\Theta_f^1, \Theta_f^2, \ldots, \Theta_f^{k_2}\}$ denote the sets of minimal elements, which satisfy $\Theta_0^j \preccurlyeq \mathbf{A}(\mathbf{x}_0)\ \forall j = 1, \ldots, k_1$ and $\Theta_f^j \preccurlyeq \mathbf{A}(\mathbf{x}_f)\ \forall j = 1, \ldots, k_2$ respectively. For our current discussion we will assume that the conditions for proposition 4.3 are not met. This implies that $\Theta_0^i \neq \Theta_f^j,\ \forall i = 1, \ldots, k_1$ and $\forall j = 1, \ldots, k_2$, or equivalently $\mathbf{A}(\mathbf{x}_0) \wedge \mathbf{A}(\mathbf{x}_f) \notin \mathcal{M}_c$.

Define $X^{ij} = \Theta_0^i \vee \Theta_f^j$. Then, X^{ij} can replace X in Problem III. Note that we no longer have $\mathbf{A}(\mathbf{x}_0) \preccurlyeq X^{ij}$ or $\mathbf{A}(\mathbf{x}_f) \preccurlyeq X^{ij}$. However, since $\Theta_0^i \preccurlyeq X^{ij}$ and $\Theta_f^j \preccurlyeq X^{ij}$ we have $\mathcal{C}_{X^{ij}} \subseteq \mathcal{C}_{\Theta_0^i} \cap \mathcal{C}_{\Theta_f^j}$. Also $\mathbf{x}_0 \in \mathcal{C}_{\Theta_0^i}$, $\mathbf{x}_f \in \mathcal{C}_{\Theta_f^j}$ and $\mathbf{x} \in \mathcal{C}_{\Theta_0^i} \cap \mathcal{C}_{\Theta_f^j}$, where $\mathbf{x}$ is the solution to problem III. Thus concatenation of segments joining $\mathbf{x}_0 \rightarrow \mathbf{x}$ and $\mathbf{x} \rightarrow \mathbf{x}_f$, provides a suboptimal solution.

This approach can be seen in light of the notion of *graph grammar* [18]. As the choice of the matrix X essentially gives a rule for graph transition while maintaining connectivity.

Figure 4 illustrates the main idea through an example. In the figure $\mathbf{A}(\mathbf{x}_0)$ represents the initial formation $\mathbf{A}(\mathbf{x}_f)$ represents the desired formation and $\mathbf{A}(\mathbf{x}_0) \vee \mathbf{A}(\mathbf{x}_f)$ represents the Adjacency matrix for the intermediate formation. As can be seen the initial and final configurations violate the condition required for Proposition 4.3. However, by passing through an intermediate configuration as provided by the choice of a suitable $X = \mathbf{A}(\mathbf{x}_0) \vee \mathbf{A}(\mathbf{x}_f)$, the graph connectivity is maintained at all times.

V. Conclusion

We have developed a mathematical framework to address dynamically evolving formation graphs due to the motion of the nodes in a plane. In this framework we considered two instances of the multi-agent coordination problem and showed that in both these instances certain convexity properties of the constraint set can be leveraged to reduce the problem to a static optimization problem. However the solution we provided in this paper is centralized and the control law it generates is an open loop control law. Thus, it suffers from the shortcomings which come from using an open loop law, like inability to handle disturbance and uncertainty. However, our approach can be used to design higher level supervisory control laws, which generates setpoints in the state space. The lower layer control laws can be designed as a feedback law to achieve these setpoints. This lower layer can control law can be made decentralized by using potential functions. Thus the control task can be split into two phases, in the first phase the agents communicate among themselves to compute the centralized policy and then the local controllers implement the decentralized algorithm. This forms the basis of future research.

References

[1] N. Moshtagh, A. Jadbabaie "Distributed Geodesic Control Laws for Flocking of Nonholonomic Agents" *IEEE Transactions on Automatic Control* (52) 4:681-686, 2007

[2] A. Muhammad and M. Egerstedt. "Positivstellensatz Certificates for Non Feasibility of Connectivity Graphs in Multi-agent Coordination". *16th IFAC World Congress*, Prague, 2005.

[3] Meng Ji, M. Egerstedt. "Distributed Formation Control While Preserving Connectedness" *IEEE Conference on Decision and Control* December 2006.

[4] M.M. Zavlanos, G.J. Pappas "Controlling Connectivity of Dynamic Graphs" *IEEE Conference on Decision and Control* December 2005.

[5] Maria Carmela De Gennaro, Ali Jadbabaie, "Decentralized Control of Connectivity of Multiagent Systems" *IEEE Conference on Decision and Control* December 2006

[6] M. Mesbahi. "On State-Dependent Dynamic Graphs and Their Controllability Properties", *IEEE Transactions on Automatic Control* (50) 3: 387- 392, 2005.

[7] Y. Kim and M. Mesbahi. "On Maximizing the Second Smallest Eigenvalue of a State-Dependent Graph Laplacian" *IEEE Transactions on Automatic Control* (51) 1:116-120, 2006

[8] D. Šiljak "Decentralized Control of Complex systems" Academic Press Inc. 1991

[9] Godsil, Royle "Algebraic Graph Theory" Springer, 2001

[10] D. Bertsekas "Convex Analysis and Optimization" *Lecture Notes*

[11] S. Boyd, L. Vandenberghe: "Convex Optimization" *Cambridge University Press* 2004.

[12] P. Erdös, A. Rényi, "On Random Graphs", *Publicationes Mathematicae* 6(1959)

[13] J.A. Fax and R.M. Murray, "Graph Laplacian and Stabilization of Vehicle formations," in *Proc. 15th IFAC, 2002*, pp. 283-288

[14] 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, no. 6, pp. 988-1001, Jun. 2003.

[15] K. Sugihara and I. Suzuki, "Distributed Motion Coordination of Multiple Robots", in *Proc. IEEE Int. Symp. Intell. Control*, 1990, pp. 138-143

[16] Athans and Falb, "Optimal Control", McGraw-Hill, 1966

[17] D. Luenberger, "Optimization by Vector Space Methods", Wiley Professional Paperback Series.

[18] E. Klavins, R. Ghrist and D. Lipsky, "A Grammatical Approach to Self-Organizing Robotic Systems", *IEEE Transactions on Automatic Control*, Vol 51, No.6, June 2006