Proceedings of the
46th IEEE Conference on Decision and Control
New Orleans, LA, USA, Dec. 12-14, 2007

Distributed Connectivity Control of Mobile Networks

Michael M. Zavlanos and George J. Pappas

Abstract— **Control of mobile networks raises fundamental and novel problems in controlling the structure of the resulting dynamic graphs. In particular, in applications involving mobile sensor networks and multi-agent systems, a great new challenge is the development of distributed motion algorithms that guarantee connectivity of the overall network. In this paper, we address this challenge using a novel control decomposition. First, motion control is performed in the continuous state space, where nearest neighbor potential fields are used to maintain existing links in the network. Second, distributed coordination protocols in the discrete graph space ensure connectivity of the switching network topology. Coordination is based on locally updated estimates of the abstract network topology by every agent as well as distributed auctions that enable tie breaking whenever simultaneous link deletions may violate connectivity. Integration of the overall system results in a distributed, multi-agent, hybrid system for which we show that, under certain secondary objectives on the agents and the assumption that the initial network is connected, the resulting motion always satisfies connectivity of the network. Our approach can also account for communication time delays in the network as well as collision avoidance, while its efficiency and scalability properties are illustrated in nontrivial computer simulations.**

I. Introduction

Controlling mobile networks has recently emerged as a fundamental problem in the control of multi-agent 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 is to achieve a coordinated objective while using only local information. The objective investigated in this paper is that of maintaining connectivity of the underlying network.

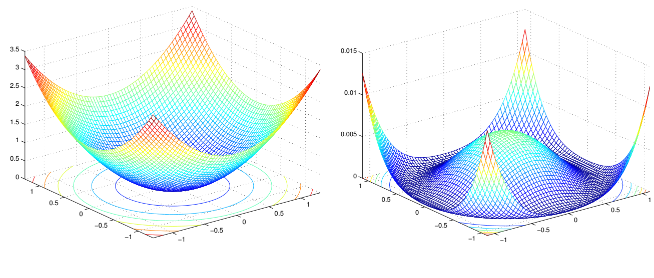
Due to their frequent appearance in multi-agent systems, dynamic networks have already received considerable attention. In [1], a measure of local connectedness of a network is introduced that under certain conditions is sufficient for global connectedness. Distributed maintenance of nearest neighbor links in formation stabilization is addressed in [2], while in [3], a controllability framework for state-dependent dynamic graphs is developed. In [4], the problem of maximizing the second smallest eigenvalue of a graph Laplacian matrix is investigated, while a decentralized approach to this problem that makes use of a supergradient algorithm and distributed eigenvector computation is considered in [5]. Network connectivity for double integrator agents is investigated in [6], where existential as well as optimal controller design results are discussed. Closely related to the topics discussed in this paper, is also work in ad-hoc sensor networks, involving cone-based topology control [7] as well as distributed algorithms that do not assume exact knowledge of agent positions [8], which, however, focuses more on the power consumption and routing problem than the actuation and control.

Motivated by the significance of connectivity in many applications involving multi-agent systems and mobile sensor networks [9], [10], [11], [12], [13], [14], in this paper, we consider graph connectivity as our primary objective. Unlike centralized [4], [15], [16], or distributed open loop [5], [6] approaches to the problem, we propose a distributed *feedback* control framework based on a novel control decomposition. In particular, motion control of the agents is performed in the continuous state space, where local potential fields are used to maintain existing proximity links in the network. On the other hand, topology control of the mobile network takes place in the discrete graph space by means of distributed coordination protocols consisting of two major components. First, locally updated estimates of the abstract network topology allow every agent to have a *rough* picture of the network. Second, distributed auctions enable tie breaking whenever simultaneous link deletions may violate connectivity, which is captured by the graph Laplacian matrix and its second smallest eigenvalue. Introducing a *hysteresis* in addition of new links in the network, allows integration of the above controllers in a hybrid model for each agent. In the presence of certain secondary objectives on the agents and under the assumption that the initial network is connected, the overall system is shown to guarantee connectivity of the mobile network, while reconfiguring towards its secondary objective. Communication time delays in the network as well as collision avoidance among the agents can also be efficiently handled, while due to its polynomial memory requirements and worst case polynomial complexity of eigenvalues computation [19], [20], our approach possesses the desired scalability properties and can be implemented on large mobile networks.

The rest of this paper is organized as follows. In Section II we define the problem of connectivity control in the presence of secondary objectives and develop the necessary graph theoretic background. In Section III, we elaborate on the controller specifications in both the continuous state-space and the discrete graph-space and discuss their integration into a hybrid model for every agents. Finally, in Section IV, we state and verify through computer simulations, nontrivial connectivity tasks that best illustrate our approach.

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



(a) $f(x,y) = \frac{1}{2}(x^2+y^2)$ (b) $f(x,y) = \frac{1}{4}(1-x^2-y^2)^2$

Fig. 1. Possible secondary objectives: Converging to the origin (a) or to a unit circle (b).

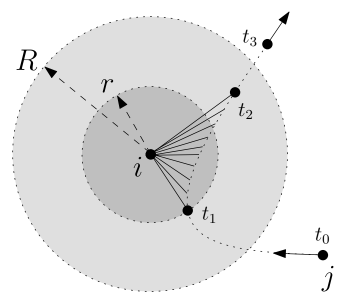


Fig. 2. Link (solid lines) dynamics according to Definition 2.1. Note that the *candidate* link $i \sim j$ is deleted at time t_2.

II. Problem Formulation

Consider n mobile agents in $\mathbb{R}^p$ and denote by $x_i(t) \in \mathbb{R}^p$ the position of agent i at time t. Let $\mathbf{x}(t) = [x_1(t) \ \dots \ x_n(t)]^T$ denote the $np \times 1$ stack vector of all agent positions and assume fully actuated agents i, such that,

$$\dot{x}_i(t) = -\nabla_{x_i} f_i(x_i(t)) + u_i(\mathbf{x}(t)) \tag{1}$$

where $f_i \geq 0$ corresponds to a continuous radially unbounded potential indicating a *secondary objective* (Fig. 1) and $u_i \in \mathbb{R}^p$ is a *primary* control input to be determined. The system of agents described in system (1), gives rise to a *dynamic graph* $\mathcal{G}(t)$, which we define as follows.

Definition 2.1 (Dynamic Graphs): We call $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ a dynamic graph consisting of a set of vertices $\mathcal{V} = \{1, \dots, n\}$ indexed by the set of agents and a time varying set of links $\mathcal{E}(t) = \{(i,j) \mid i,j \in \mathcal{V}\}$ such that, for any $0 < r < R$,

- if $(i,j) \notin \mathcal{E}(t)$ and $0 < \|x_i(t) - x_j(t)\|_2 < r$ then, (i,j) is a candidate link to be *added* to $\mathcal{E}(t)$,
- if $(i,j) \in \mathcal{E}(t)$ and $r \leq \|x_i(t) - x_j(t)\|_2 < R$ then, (i,j) is a candidate link to be *deleted* from $\mathcal{E}(t)$,
- if $R \leq \|x_i(t) - x_j(t)\|_2$ then, $(i,j) \notin \mathcal{E}(t)$.

Dynamic graphs $\mathcal{G}(t)$ such that $(i,j) \in \mathcal{E}(t)$ if and only if $(j,i) \in \mathcal{E}(t)$ are called *undirected* and consist the main focus of this paper. Moreover, any vertices i and j of an undirected graph $\mathcal{G}(t)$, that are joined by a link $(i,j) \in \mathcal{E}(t)$, are called adjacent or neighbors at time t and are denoted by $i \sim j$. Definition 2.1 implies that all links in $\mathcal{G}(t)$ are essentially controllable. In particular, the neighborhood of every vertex in $\mathcal{G}(t)$ is partitioned into two disjoint sets in $\mathbb{R}^p$, i.e., an open ball and an annulus, where addition and deletion of links, respectively, takes place (Fig. 2).[1] Note that the particular partitioning of the state-space neighborhood of every vertex in $\mathcal{G}(t)$, introduces a *hysteresis* in addition of new links in $\mathcal{G}(t)$, which is critical in integrating topology control of the network with motion control of the agents (Section III).

Given any dynamic graph $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}(t))$, we say that a graph $\mathcal{G}_i(t) = (\mathcal{V}_i(t), \mathcal{E}_i(t))$ is a *subgraph* of $\mathcal{G}(t)$, if $\mathcal{V}_i(t) \subseteq \mathcal{V}(t)$ and $\mathcal{E}_i(t) \subseteq \mathcal{E}(t)$. If $\mathcal{V}_i(t) = \mathcal{V}(t)$, we call $\mathcal{G}_i(t)$ a *spanning subgraph* of $\mathcal{G}(t)$. A subgraph $\mathcal{G}_i(t)$ of $\mathcal{G}(t)$ is called an *induced subgraph* if two vertices of $\mathcal{V}_i(t)$ are adjacent in $\mathcal{G}_i(t)$ if and only if they are adjacent in $\mathcal{G}(t)$. A topological invariant of graphs that also corresponds to the *primary objective* for system (1), is graph connectivity.

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

Then, the problem addressed in this paper is,

Problem 1 (Distributed Connectivity Control): Given the set of connected graphs $\mathcal{C}_n$ on n vertices, determine distributed control laws $u_i(\mathbf{x}(t))$ for all agents i so that if $\mathcal{G}(t_0) \in \mathcal{C}_n$, then $\mathcal{G}(t) \in \mathcal{C}_n$ for all time.

Problem 1 equivalently implies that we want the set $\mathcal{C}_n$ to be an invariant of motion for system (1). We achieve this goal by choosing an equivalent formulation, using the algebraic representation of the dynamic graph $\mathcal{G}(t)$. In particular, the structure of any dynamic graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ can be equivalently represented by a dynamic *Laplacian* matrix,

$$L(t) = \Delta(t) - A(t) \tag{2}$$

where $A(t) = (a_{ij}(t))$ corresponds to the *Adjacency matrix* of the graph $\mathcal{G}(t)$, which is such that $a_{ij}(t) = 1$ if $(i,j) \in \mathcal{E}(t)$ and $a_{ij}(t) = 0$ otherwise and $\Delta(t) = \text{diag}\left(\sum_{j=1}^n a_{ij}(t)\right)$ denotes the *Valency matrix*.[2] Note that for undirected graphs, the Adjacency matrix is a symmetric matrix and hence, so is the Laplacian matrix. The spectral properties of the Laplacian matrix are closely related to graph connectivity. In particular, we have the following lemma.

Lemma 2.3 ([17]): Let $\lambda_1(L(t)) \leq \lambda_2(L(t)) \leq \cdots \leq \lambda_n(L(t))$ be the ordered eigenvalues of the Laplacian matrix $L(t)$. Then, $\lambda_1(L(t)) = 0$ for all t, with corresponding eigenvector $\mathbf{1}$, i.e., the vector of all entries equal to 1. Moreover, $\lambda_2(L(t)) > 0$ if and only if $\mathcal{G}(t)$ is connected.

Note that computation of the spectrum of a matrix has worst case complexity $O(n^3)$, where n is the size of the matrix [19]. This complexity can, however, be reduced to $O(n)$ for sparse symmetric matrices [20], as is the Laplacian matrix $L(t)$ in the case of large networks, commonly appearing in the proposed setting. Consequently, dealing with eigenvalues does not introduce significant computational overhead, which makes our approach scalable to large size networks.

[1] State-dependent dynamic graphs $\mathcal{G}(t)$ as in Definition 2.1, are sometimes also called *proximity graphs*.

[2] Since we do not allow self-loops, we define $a_{ii}(t) = 0$ for all i.

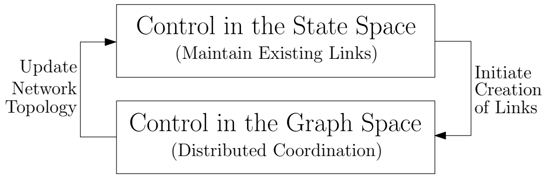


Fig. 3. Control Decomposition.

III. Distributed Topology Control

Motivated by the inherently discrete nature of graphs as combinatorial objects, in this paper we employ a novel control decomposition in the *continuous* state space and *discrete* graph space, respectively (Fig. 3). In particular, motion control of the agents is performed in the continuous state space and aims at maintaining existing links in the network. On the other hand, topology control of the discrete graph structure, namely addition and, more important, deletion of links in $\mathcal{G}(t)$, is due to distributed coordination protocols in the discrete graph space. State-dependence of the network topology (Definition 2.1) and in particular, the hysteresis in creation of new links, enables integration of the resulting controllers in a hybrid model for every agent.

A. Maintaining Existing Links in the Network

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ indicate any given topology of the network and denote by $\mathcal{N}_i = \{j \mid (i,j) \in \mathcal{E}\}$ the neighbors of agent i in the graph $\mathcal{G}$. The goal in this section is to design distributed control laws $u_i(t)$ for all agents i that guarantee that all links in $\mathcal{G}$ are maintained and that collisions among adjacent agents are avoided. The construction is based on potential fields that "blow up" whenever the state of the system violates any of these specifications. In particular, for every agent i, let $\varphi_i(\mathbf{x}) = \sum_{j \in \mathcal{N}_i} \varphi_{ij}(x_{ij})$ where,

$$\varphi_{ij}(x_{ij}) = \frac{1}{\|x_{ij}\|_2^2} + \frac{1}{R^2 - \|x_{ij}\|_2^2}$$

Then, we have the following result.[3]

Theorem 3.1 ([18]): For all agents i, assume secondary objectives described by continuous potentials $f_i : \mathbb{R}^p \to \mathbb{R}_+$ such that $\lim_{\|x_i\|_2 \to \infty} f_i(x_i) = \infty$ (radially unbounded) and let the control inputs be defined by,[4]

$$u_i(\mathbf{x}) := -K \nabla_{x_i} \varphi_i(\mathbf{x}) \tag{3}$$

Then, the closed loop system (1) guarantees that all links in $\mathcal{G}$ are maintained and collisions among agents are avoided.

B. Distributed Coordination

Having developed a distributed motion control framework that maintains existing links among adjacent agents, we now propose distributed coordination protocols to control the discrete structure of the network. The two major components of our approach are locally updated *spanning subgraphs* of the abstract network topology that allow every agent to have a *rough* picture of the network structure and a *market-based* coordination algorithm that enables tie breaking whenever simultaneous link deletions may violate connectivity.

[3]We emphasize the fact that the potentials $f_i(x_i)$ consist *secondary* objectives and there is no guarantee that they will be achieved.

[4]We denote by $\mathbb{R}_+$ the set $[0, \infty)$.

TABLE I

$a_{jk}^{[i]}(t)$	$v_{jk}^{[i]}(t)$	$a_{jk}^{[i]}(t+1)$
1	1	0
1	0	1
0	1	1
0	0	0

1) Locally Updated Spanning Subgraphs: Since connectivity is a global property of the graph $\mathcal{G}(t)$, it is necessary that every agent has sufficient knowledge of the structure of $\mathcal{G}(t)$ in order to be able to safely delete a link with a neighbor without violating connectivity. One can easily imagine scenarios where lack of such information might result in violation of connectivity, i.e., if an agent is not aware that the graph is a tree. Hence, we assume that every agent i can locally estimate a *spanning subgraph* $\mathcal{G}_i(t) = (\mathcal{V}, \mathcal{E}_i(t))$ of the graph $\mathcal{G}(t)$, using information from its nearest neighbors $\mathcal{N}_i(t) = \{j \mid (i,j) \in \mathcal{E}(t)\}$ only.[5] In particular, let $A_i(t) = \left(a_{jk}^{[i]}(t)\right)$ denote the Adjacency matrix corresponding to the graph $\mathcal{G}_i(t)$ at time t. Then, the dynamics of a link $j \sim k$ can be expressed as (Table I),[6]

$$a_{jk}^{[i]}(t+1) := \neg\left(a_{jk}^{[i]}(t) \leftrightarrow v_{jk}^{[i]}(t)\right) \tag{4}$$

where $v_{jk}^{[i]}(t) \in \{0,1\}$ indicates a control input, such that $v_{jk}^{[i]}(t) = 1$ indicates an action to create or delete a link $j \sim k$.[7] In matrix form, the dynamics in equation (4) become,

$$A_i(t+1) := H_i(t) := \neg\left(A_i(t) \leftrightarrow V_i(t)\right) \tag{5}$$

where the control input $V_i(t) = \left(v_{jk}^{[i]}(t)\right)$ is a symmetric matrix ensuring that, if $A_i(t_0)$ is symmetric, then $A_i(t)$ is also symmetric for all time $t \geq t_0$.

Let $E_i = \vee_{j \neq i}(e_i e_j^T \vee e_j e_i^T)$, where e_i is a column vector of all entries equal to zero but the i-th entry which is equal to one. Then, the expression $E_i \wedge (\neg A_i(t))$ indicates links $i \sim j$ that agent i can create with agents $j \notin \mathcal{N}_i(t)$. Moreover, let $A_i^{(1)}(t) = \vee_{j \in \mathcal{N}_i(t)} A_j(t)$ indicate existing links in the network, available by the 1-hop neighbors $\mathcal{N}_i(t)$ of agent i. Then, the expression $(\neg A_i(t)) \wedge A_i^{(1)}(t)$ indicates existing links in the network that agent i is not currently aware of. Hence, the expression,

$$F_i(t) := \left((\neg A_i(t)) \wedge A_i^{(1)}(t)\right) \vee \left(E_i \wedge (\neg A_i(t))\right) \tag{6}$$

indicates, either existing links in the network that agent i is not aware of, or links that agent i can create with agents $j \notin \mathcal{N}_i(t)$. Clearly, $F_i(t)$ captures all new links that agent i can add to $A_i(t)$. On the other hand, the Adjacency matrix $A_i(t)$ includes all existing links in $\mathcal{G}_i(t)$, which are also candidates

[5]The requirement that $\mathcal{G}_i(t)$ is a *spanning subgraph* of $\mathcal{G}(t)$ is necessary to guarantee connectivity of the graph $\mathcal{G}(t)$ for all $t \geq t_0$ [18].

[6]See Appendix for an overview of Boolean Operations.

[7]The discrete time semantics in (4) are associated with transitions of a hybrid automaton modeling every agent, defined in [18].

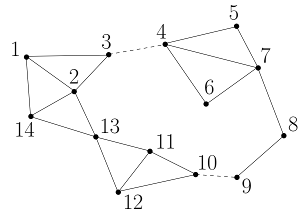


Fig. 4. Deleting any of the links (3,4) or (9,10) individually does not violate connectivity, but deleting them both results in a disconnected graph. None of the involved agents are adjacent.

to be deleted. Since $F_i(t) \wedge A_i(t) = 0$, we can decouple the control input $V_i(t)$ into a component $F_i(t) \wedge V_i^c(t)$ regulating link *creations* and a component $A_i(t) \wedge V_i^d(t)$ regulating link *deletions*. Hence, the control input becomes,

$$V_i(t) := \big(F_i(t) \wedge V_i^c(t)\big) \vee \big(A_i(t) \wedge V_i^d(t)\big) \qquad (7)$$

Note that the component $F_i(t) \wedge V_i^c(t)$ prevents updating $A_i(t)$ with new links $j \sim k$, where $j, k \neq i$, if agent i is not informed about them by its neighbors. On the other hand, if $\big((\neg A_i(t)) \wedge A_i^{(1)}(t)\big) \rightarrow V_i^c(t)$, then $A_i(t)$ is updated with all existing links in the network that agent i is not aware of. It can be shown that if all estimates $\mathcal{G}_i(t)$ are explicitly initialized with nearest neighbor links, the overall network $\mathcal{G}(t)$ is fixed and connected and the update rule for all agents is as in (5), then all agents achieve *consensus* on their estimates $A_i(t)$ in finite time. Hence, estimates $\mathcal{G}_i(t)$ provide the agents with a *rough* picture of the overall network, as desired [18].

2) Market-Based Coordination: Given the local network dynamics (5), the main challenge is to determine control inputs $V_i^c(t)$ and $V_i^d(t)$ that ensure connectivity of each subgraph $\mathcal{G}_i(t)$ for all time t. Since $\mathcal{G}_i(t) \subseteq \mathcal{G}(t)$ [18], connectivity of $\mathcal{G}_i(t)$ for all i implies connectivity of the overall network $\mathcal{G}(t)$. Regarding creation of links in $\mathcal{G}_i(t)$ by agent i, we define the control input $V_i^c(t) = \big(v_{jk}^{[i]c}\big)$ as,

$$v_{jk}^{[i]c}(t) := \neg\big((j = i) \vee (k = i)\big) \vee \big(x_k(t) \in \mathcal{B}_r(x_j(t))\big) \quad (8)$$

where,

$$\mathcal{B}_r(x_i(t)) = \{\ y(t) \in \mathbb{R}^p \mid \|y(t) - x_i(t)\|_2 < r\ \}$$

denotes an open ball of radius $r > 0$ (as in Definition 2.1) centered at $x_i(t) \in \mathbb{R}^p$. Clearly, $(j = i) \vee (k = i)$ guarantees the local nature of the controller (note that $v_{ij}^{[i]c}(t_1) = 1$ in Fig. 2). Note that $v_{jk}^{[i]c}(t) = 1$ for all $j, k \neq i$, which implies that $\big((\neg A_i(t)) \wedge A_i^{(1)}(t)\big) \rightarrow V_i^c(t)$ and so by equation (5), $A_i(t)$ is updated with all existing links in the network that agent i is not aware of.

Deletion of links $i \sim j$ by any agent i is, however, nontrivial due to the possibility for simultaneous link deletions by multiple non-adjacent agents that may generate a disconnected graph (Fig. 4). To avoid such scenarios, we require that at most one link can be deleted from $\mathcal{G}(t)$ at every time instant t and employ a *market-based* coordination framework to achieve consensus of all agents regarding the

Algorithm 1 Auction Mechanism for Agent i

Compute the set of *safe* neighbors $\mathcal{S}_i(t)$ which is such that if the link $i \sim j$ with $j \in \mathcal{S}_i(t)$ is deleted from $\mathcal{E}_i(t)$, then $\mathcal{G}_i(t)$ remains connected, i.e., $\mathcal{S}_i(t) := \{j \in \mathcal{N}_i(t) \backslash \mathcal{N}_i^r(t) \mid \lambda_2(L(\mathcal{E}_i(t) \backslash \{i \sim j\})) > 0\}$

Initialize a request for a link deletion $r_i = [\ i \quad g(\mathcal{S}_i(t)) \quad bid(\mathcal{S}_i(t), b)\ ]^T \in \mathbb{R}^3$ that consists of the link $i \sim g(\mathcal{S}_i(t))$ that is to be deleted and a bid $b \in \mathbb{R}_+$ indicating how "important" this request is.

Initialize a set of max-bids $\mathcal{M}_i(t) \in 2^{\mathbb{R}^3}$ and a binary vector of tokens $T_i(t) \in \{0, 1\}^n$ indicating the start of an auction, by $\mathcal{M}_i(t) := \{r_i(t)\}$ and $T_{ii} = 1$, respectively.

while $(\wedge_{j=1}^n T_{ij}(t)) = 0$ **do**

  Collect tokens from adjacent agents only, i.e., $T_i(t+1) := T_i(t) \vee \big(\vee_{j \in \mathcal{N}_i(t)} T_j(t)\big)$

  Apply a max-consensus update on $\mathcal{M}_i(t)$, i.e., $\mathcal{M}_i(t+1) := \big\{r_j \mid j = \underset{r_k \in \cup_{l \in \{\mathcal{N}_i(t), i\}} \mathcal{M}_l(t)}{\operatorname{argmax}} \{r_{k3}\}\big\}$

end while

Compute the *winner* link of the auction $w_i(t)$, i.e., $w_i(t) := \{r_{j1} \sim r_{j2} \mid r_j \in \mathcal{M}_i(t)\}$

deleted link. For this, let $\mathcal{N}_i^r(t) = \{j \mid x_j(t) \in \mathcal{B}_r(x_i(t))\}$ denote the set of r-neighbors of agent i at time t. Then, $\mathcal{N}_i^r(t) \subseteq \mathcal{N}_i(t)$ and links $i \sim j$ with neighbors $j \in \mathcal{N}_i(t) \backslash \mathcal{N}_i^r(t)$ are all candidates to be deleted, by Definition 2.1. Define further the functions $bid : 2^{\mathbb{N}} \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$ and $g : 2^{\mathbb{N}} \rightarrow \mathbb{N}$ with,

$$bid(X, b) := \begin{cases} b \in X & \text{if } X \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

and,

$$g(X) := \begin{cases} i \in X & \text{if } X \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

where $i \in X$ can be chosen according to any policy, deterministic or not. Then, the proposed auction mechanism is described in Algorithm 1 and the corresponding market-based coordination scheme consists of multiple such auctions taking place sequentially.

Given the winning link $w_i(t)$ corresponding to the highest bid in the auction in Algorithm 1, the control input that regulates link deletions $V_i^d(t) = \big(v_{jk}^{[i]d}(t)\big)$, becomes (note that $v_{ij}^{[i]d}(t_2) = 1$ in Fig. 2),

$$v_{jk}^{[i]d}(t) := (w_i(t) = j \sim k) \wedge (|w_i(t)| = 1) \qquad (9)$$

Note that whenever $|w_i(t)| > 1$, there is either a tie in the bids or no bids were sent in the network. In any such case, $V_i^d(t) = \mathbf{0}$ for all agents i and no link is deleted from any set $\mathcal{E}_i(t)$. To ensure correctness of the above market-based coordination scheme, all agents need to be synchronized in the same auction for all time [18]. This requirement becomes even more important in the presence of communication time delays [18].

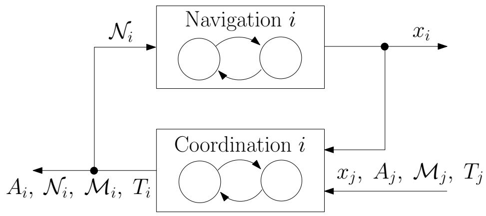


Fig. 5. Closed loop hybrid model for agent i.

Remark 3.2: Note that any positive real numbers can serve as bids in Algorithm 1. However, letting $b \geq 0$ be a function of the distance $\|x_i(t) - x_{g(\mathcal{S}_i)}(t)\|_2$ or the size of the neighbor set $|\mathcal{N}_i(t)|$ is a rather natural choice that can also be associated with signal strength or power consumption properties of the overall network.

Remark 3.3: The condition $(\wedge_{j=1}^{n} T_{ij}(t)) = 1$ indicating the end of the *while* loop in Algorithm 1, clearly implies convergence of the max-consensus algorithm on the sets $\mathcal{M}_i(t)$ to the maximum over all agents. The motivation for using tokens $T_i(t)$ to indicate the end of an auction, instead of a fixed $(n-1)$-step termination condition, which for connected networks is sufficient for convergence of the max-consensus algorithm, is twofold. First, it utilizes the structure of the network resulting in more efficient updating.[8] Second, it can deal with communication time delays in the system, where the time required for $n-1$ steps of Algorithm 1 can be significantly different for distinct agents, preventing convergence of all agents to a common outcome. It is also worth noting that the memory and communication cost for transmitting the binary tokens is minimal.

C. Closed Loop Hybrid Agent

Integration of the motion control laws of Section III-A with the discrete topology control protocols of Section III-B.2 leads to a hybrid model for every agent. This model consists of a *navigation* and a *coordination* automaton, implementing motion and market-based connectivity control, respectively (Fig. 5). *Input* to the navigation automaton of agent i consists the neighbor set $\mathcal{N}_i(t)$ and the states $x_j(t)$ of all agents. Although all agents' states are considered *shared* in this hybrid system, only the ones associated with agents in $\mathcal{N}_i(t)$ are practically required. The *output* of the navigation automaton is then the agent's updated state $x_i(t)$ obtained by integrating (1)-(3). *Inputs* to the coordination automaton of agent i are the states $x_j(t)$ of all agents, their network estimates $A_j(t)$, their max-bid sets $\mathcal{M}_j(t)$, and their token vectors $T_j(t)$. Similarly, all these variables are shared, however only those corresponding to $j \in \mathcal{N}_i(t)$ are required by agent i. The *outputs* of the coordination automaton are then the agent's updated states $\mathcal{N}_i(t)$, $A_i(t)$, $\mathcal{M}_i(t)$ and $T_i(t)$. The two automata are synchronized into a single closed loop hybrid system [18].

Remark 3.4 (Scalability): Note that memory and communication requirements for every agent are polynomial and in particular of the order $O(n^2)$, due to handling and transmitting the adjacency matrices $A_i(t)$.

[8]For a complete graph, one step is sufficient for convergence of the max-consensus algorithm.

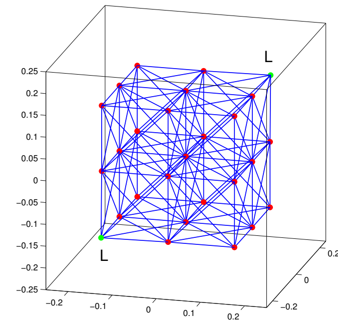


Fig. 6. 12 agents with 2 leaders (Initial Configuration).

IV. CONNECTIVITY TASKS

In this section we illustrate the proposed algorithm in non-trivial connectivity tasks and show that it has the desired connectivity, collision avoidance and scalability properties. We assume a 3-dimensional workspace and classify the agents into a set $\mathcal{L} \subseteq \{1,\ldots,n\}$ of *leaders* having a non-trivial secondary objective and a set of *followers* $\{1,\ldots,n\}\backslash\mathcal{L}$ having no secondary objective. In particular, for all leaders $i \in \mathcal{L}$ we assume secondary objectives as in Fig. 1(b), where we also add unit angular velocities, slightly abusing the secondary objective specifications in equation (1).

The connectivity task we consider consists of $n = 27$ agents initialized as in Fig. 6. We assume two leaders, namely $\mathcal{L} = \{1,2\}$, with secondary objective parameters chosen to *stretch* the network and observe whether it is able to reconfigure while maintaining connectivity. The link ranges in Definition 2.1 are $r = .25$ and $R = .4$.

Fig. 7 shows the evolution of the system at two consecutive time instants. Agents are indicated by dots, while the leaders are also labeled by the letter "L". Strong links between the agents, i.e., links with inter-agent distances less than r, are indicated by solid lines, while candidate links for deletion, by dotted lines (Definition 2.1). Solid curves attached attached to every agent indicate the recently traveled paths and give an idea of the agents' motion. Note that, under the proposed connectivity control laws, the overall network remains connected, while the leaders "do their best" to achieve their secondary objectives. Note also that in the absence of strong links in the network our algorithm generates a minimally connected network, i.e., a tree structure.

V. CONCLUSIONS

In this paper, we considered the problem of controlling a group of agents so that the resulting motion always preserves the connectivity property of the underlying network. Our approach was based on a novel control decomposition, where motion control of the agents was performed in the continuous state-space, while topology control of the network was embedded in the discrete graph-space and relied on abstract network information, locally updated by every agent, and market-based coordination. Integration of the above

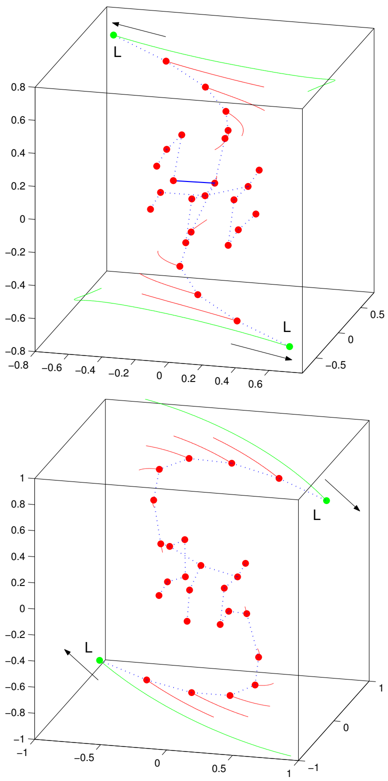


Fig. 7. Connectivity control for 27 agents with 2 leaders (for two consecutive time instants t).

controllers resulted in a hybrid model for every agent and, in the presence of certain secondary objectives, the overall system was shown to always guarantee connectivity of the network. Communication time delays in the network as well as collision avoidance among the agents were also efficiently handled, while our approach was illustrated through a class of interesting problems that could be achieved while preserving connectivity.

REFERENCES

[1] D. P. Spanos and R. M. Murray. *Robust Connectivity of Networked Vehicles*, Proceedings of the 43rd IEEE Conference on Decision and Control, pp. 2893-2898, Bahamas, Dec. 2004.

[2] M. Ji and M. Egerstedt. *Distributed Formation Control while Preserving Connectedness*, Proceedings of the 45th IEEE Conference on Decision and Control, pp. 5962-5967, San Diego, CA, Dec. 2006.

[3] M. Mesbahi. *On State-dependent Dynamic Graphs and their Controllability Properties*, IEEE Transactions on Automatic Control, vol. 50(3), pp. 387-392, March 2005.

[4] Y. Kim and M. 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] M. C. DeGennaro and A. Jadbabaie. *Decentralized Control of Connectivity for Multi-Agent Systems*, Proceeding of the 45th IEEE Conference on Decision and Control, pp. 3628-3633, San Diego, CA, Dec. 2006.

[6] G. Notarstefano, K. Savla, F. Bullo and A. Jadbabaie. *Maintaining Limited-Range Connectivity among Second-Order Agents*, American Control Conference, pp. 2124-2129, Minneapolis, MN, June 2006.

[7] L. Li, J. Y. Halpern, P. Bahl, Y. M. Wang and R. Wattenhofer. *A Cone-Based Distributed Topology-Control Algorithm for Wireless Multi-Hop Networks*, IEEE/ACM Transactions on Networking, Vol. 13(1), pp. 147-159, Feb. 2005.

[8] R. Wattenhofer and A. Zollinger. *XTC: A Practical Topology Control Algorithm for Ad-Hoc Networks*, Proceedings of the 18th IEEE International Parallel and Distributed Processing Symposium, pp. 216-223, Santa Fe, New Mexico, April 2004.

[9] 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, June 2003.

[10] 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(9), pp. 1520-1533, Sep. 2004.

[11] H. Tanner, A Jadbabaie and G. Pappas. *Flocking in Fixed and Switching Networks*, IEEE Transactions on Automatic Control, vol. 52(5), pp. 863-868, May 2007.

[12] R. Sepulchre, D. Paley, N. E. Leonard. *Stabilization of Planar Collective Motion: All-to-All Communication*, IEEE Transactions on Automatic Control, vol. 52(5), pp. 811-824, May 2007.

[13] W. Ren and R. Beard. *Consensus of Information under Dynamically changing Interaction Topologies*, Proceedings of the American Control Conference, pp. 4939-4944, June 2004.

[14] 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, Aug. 2006.

[15] 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.

[16] M. M. Zavlanos and G. J. Pappas. *Potential Fields for Maintaining Connectivity of Mobile Networks*, IEEE Transactions on Robotics, vol. 23(4), pp. 812-816, Aug. 2007.

[17] Norman Biggs, *Algebraic Graph Theory*, Cambridge University Press, 1993.

[18] M. M. Zavlanos and G. J. Pappas. *Distributed Connectivity Control of Mobile Networks*, Technical Report, University of Pennsylvania, March 2007. (`http://www.seas.upenn.edu/~zavlanos/`)

[19] G. H. Golub and C. F. Van Loan. *Matrix Computations*, Johns Hopkins University Press, 1996.

[20] Shing-Tung Yau and Ya Yan Lu. *A New Approach to Sparse Matrix Eigenvalues*, Proceedings of the 1st IEEE Conference on Regional Aerospace Control Systems, pp. 132-137, May 1993.

APPENDIX I
BOOLEAN OPERATIONS

Definition 1.1 (Boolean Operations on Boolean Variables): Given *boolean variables* $x, y \in \{0, 1\}$, we define the operations $\neg x$, $x \wedge y$, $x \vee y$, $x \rightarrow y$ and $x \leftrightarrow y$ as in Table II, where the symbols $\neg$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$ stand for *not*, *and*, *or*, *if, then* and *if and only if*, respectively.

TABLE II

BOOLEAN OPERATIONS

x	y	$\neg x$	$x \wedge y$	$x \vee y$	$x \rightarrow y$	$x \leftrightarrow y$
1	1	0	1	1	1	1
1	0	0	0	1	0	0
0	1	1	0	1	1	0
0	0	1	0	0	1	1

In a similar way, we can define boolean operations on *boolean matrices* $X, Y \in \{0, 1\}^{n \times n}$.

Definition 1.2 (Boolean Operations on Boolean Matrices): Let $X = (x_{ij})$ and $Y = (y_{ij})$ be $n \times n$ boolean matrices. Then, the boolean operations $\neg$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$ on the matrices X and Y are defined *elementwise* on their entries.

Hence, the boolean matrix $X \wedge Y$ is defined as $X \wedge Y := (x_{ij} \wedge y_{ij})$ and in a similar way we can define any other boolean operation on matrices.