

# Distributed Connectivity Control of Mobile Networks

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

**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. Motivated by the inherently discrete nature of graphs as combinatorial objects, we address this challenge using a key control decomposition. First, connectivity control of the network structure is performed in the *discrete* space of graphs and relies on local estimates of the network topology used, along with algebraic graph theory, to verify link deletions with respect to connectivity. Tie breaking when multiple such link deletions can violate connectivity is achieved by means of gossip algorithms and distributed market-based control. Second, motion control is performed in the *continuous* configuration space, where nearest neighbor potential fields are used to maintain existing links in the network. Integration of the above controllers results in a distributed, multi-agent, hybrid system, for which we show that the resulting motion always ensures connectivity of the network, while it reconfigures towards certain secondary objectives. Our approach can also account for communication time delays as well as collision avoidance, and is illustrated in nontrivial computer simulations.

**Index Terms**—Dynamic networks, distributed control, hybrid systems, graph connectivity.

## I. INTRODUCTION

RECENT advances in communication and computation have given rise to distributed control of multi-agent systems which, compared to classical control, provides increased efficiency, performance, scalability and robustness. A great challenge in this new field is achieving a global coordinated objective while using only local information [1]–[14]. The objective investigated in this paper is that of maintaining connectivity of a dynamic network consisting of multiple mobile agents [15]–[26].

Due to their frequent appearance in multi-agent systems, dynamic networks have already received considerable attention. In [15], a controllability framework for state-dependent dynamic graphs is introduced. A measure of local connectedness of a network that under certain conditions is sufficient for global connectedness is proposed in [16], while distributed maintenance of nearest neighbor links in formation stabilization is addressed in [17]. In [18], 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 [19]. Network connectivity for double integrator agents is investigated in [20], 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 [21], [22] and distributed algorithms that do not assume exact knowledge of agent positions [23], [24]. This work, however, focuses more on the power consumption and routing problem than the actuation and control.

Unlike centralized [18], [25], [26], distributed open loop [19], [20], or approaches that essentially restrict connectivity control to link additions [16]–[20], [25], we propose a distributed *feedback* and *provably correct* control framework that imposes no restrictions on the network topology other than the desired connectivity specification. To this end, connectivity control of the network structure is performed in the *discrete* space of graphs and relies on two key ideas. First, local estimates of the network topology provide every agent with a *rough* picture of the network structure used, along with notions from algebraic graph theory, to verify link deletions with respect to connectivity. Second, gossip algorithms and distributed market-based control allow tie breaking whenever multiple such link deletions can violate connectivity. On the other hand, motion control of the agents is performed in the *continuous* configuration space by means of local potential fields used to maintain existing links in the network. Integration of the above controllers is possible due to a novel representation of the network topology by a class of proximity graphs that impose a hysteresis in link additions, and results in a hybrid model for every agent [27]. Under the assumption that the initial network is connected, the overall hybrid system is shown to guarantee connectivity of the mobile network for all time, while it reconfigures towards certain secondary objectives. Communication time delays in the network as well as collision avoidance can also be efficiently handled, while our approach is illustrated through a class of interesting problems that can be achieved while preserving connectivity.

The rest of this paper is organized as follows. In Section II we define the problem of controlling connectivity of dynamic networks and develop a necessary graph theoretic background. In Section III we discuss the control challenges associated with a distributed solution to the problem and propose market-based control to efficiently address them. The resulting distributed hybrid agent is defined in Section IV, while properties as well as correctness of the overall system are discussed in Section V. Integration with agent mobility is considered in Section VI and, finally, in Section VII, nontrivial connectivity tasks are discussed 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. Preliminary versions of this work can be found in [32].

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. Dynamic Networks

Consider a network of  $n$  agents with integrated wireless communication capabilities and denote by  $(i, j)$  a communication link between agents  $i$  and  $j$ . We assume that communication links between the agents can be enabled and disabled in time due to power constraints, agent mobility, or networking specifications such as routing and throughput. This gives rise to the notion of a *dynamic* graph  $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ , where  $\mathcal{V} = \{1, \dots, n\}$  consists the set of vertices indexed by the set of agents and  $\mathcal{E}(t) = \{(i, j) \mid i, j \in \mathcal{V}\}$  denotes a time varying set of links. We assume bidirectional communication links and so  $(i, j) \in \mathcal{E}(t)$  if and only if  $(j, i) \in \mathcal{E}(t)$ . Such graphs are called *undirected* and consist the main focus of this paper. 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  $t$ . Hence, we can define the set of neighbors of agent  $i$  at time  $t$ , by  $\mathcal{N}_i(t) = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}(t)\}$ . An important topological invariant of graphs is *graph connectivity*.

**Definition 2.1 (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  $\mathcal{G}(t)$ .

Although the agents' primary task is detection of certain physical changes within their proximity, their communication capabilities enable them to share the individually collected data with their peers in order to achieve a global coordinated objective, such as *consensus* on their local measurements [1]–[6]. Consequently, network connectivity becomes a critical requirement. Motivated by these observations, in this paper we address the following problem.

**Problem 1 (Distributed Connectivity Control):** Given an initially connected network  $\mathcal{G}(t_0)$  consisting of  $n$  agents, determine local control laws that regulate addition and deletion of links between adjacent agents so that the dynamic network  $\mathcal{G}(t)$  is connected for all time.

Note that Problem 1 focuses on the *safety* specification associated with control of network connectivity, rather than satisfaction of possible global coordinated objectives, which, for the purposes of this work, are considered *secondary*. Therefore, our goal is to ensure invariance of the network  $\mathcal{G}(t)$  with respect to connectivity. We achieve this goal by choosing an equivalent formulation, using the algebraic representation of a dynamic graph. 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*

$$L(t) = \Delta(t) - A(t), \quad (1)$$

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  $\Delta(t) = \text{diag}(\sum_{j=1}^n a_{ij}(t))$  denotes the *valency matrix*.<sup>1</sup> 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.

<sup>1</sup>Since we do not allow self-loops, we define  $a_{ii}(t) = 0$  for all  $i$ .

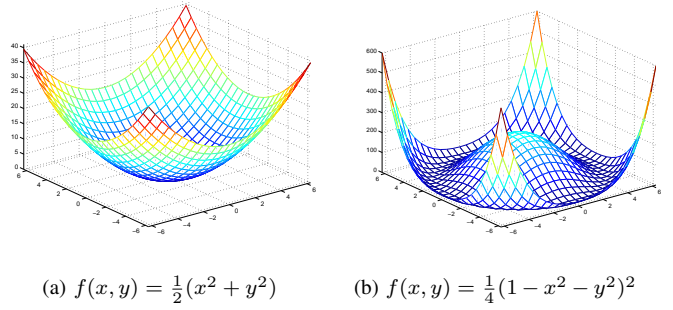


Fig. 1. Drifts capturing convergence to a point (a) or to a unit circle (b).

**Lemma 2.2 ([28]):** Let  $\lambda_1(L(t)) \leq \lambda_2(L(t)) \leq \dots \leq \lambda_n(L(t))$  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.

**Remark 2.3 ( $k$ -Connectivity):** Given any graph  $\mathcal{G}$ , the vertex connectivity  $\kappa(\mathcal{G})$  of  $\mathcal{G}$  is defined as the minimum number of vertices that, if deleted from  $\mathcal{G}$ , increase the number of connected components of the graph [28]. Similarly, the edge connectivity  $\eta(\mathcal{G})$  of  $\mathcal{G}$  can be defined as the minimum number of edges that, if deleted from  $\mathcal{G}$ , increase its number of connected components. For any graph  $\mathcal{G}$ , its vertex connectivity, edge connectivity and second smallest eigenvalue of its laplacian matrix satisfy  $\lambda_2(L(\mathcal{G})) \leq \kappa(\mathcal{G}) \leq \eta(\mathcal{G})$  [28]. Furthermore, for any  $k \leq \kappa(\mathcal{G})$  the graph  $\mathcal{G}$  is called  $k$ -connected. Clearly, if  $\lambda_2(L(\mathcal{G})) > k - 1$ , then  $\mathcal{G}$  is  $k$ -connected. Note that for  $k = 1$  we get  $\lambda_2(L(\mathcal{G})) > 0$  which corresponds to graph connectivity as in Definition 2.1.

### B. Dynamic Mobile Networks

As discussed in Section II-A, dynamic networks are due to either power constraints, networking specifications or agent mobility. In this paper, we focus on the latter and attempt to redefine the notion of a dynamic network  $\mathcal{G}(t)$  so that it captures this new specification. In particular, 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  $t$ . Assume, further, fully actuated agents  $i$ , such that

$$\dot{x}_i(t) = -\nabla_{x_i} f_i(t) + u_i(t), \quad (2)$$

where  $f_i(t) \geq 0$  is a global secondary objective (drift) modeled by a twice differentiable, radially unbounded potential (Fig. 1) and  $u_i(t) \in \mathbb{R}^p$  is a control input associated with the connectivity specification of the network (Problem 1). The system of agents described in system (2), gives rise to the following definition of a *dynamic graph*.

**Definition 2.4 (Dynamic Graphs):** We call  $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$  a dynamic graph consisting of a set of vertices  $\mathcal{V} = \{1, \dots, n\}$  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,  $(i, j) \notin \mathcal{E}(t)$ .

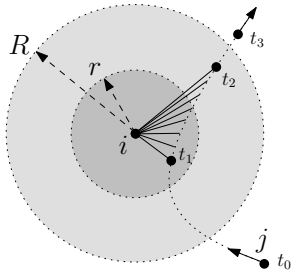


Fig. 2. Link (solid line) dynamics according to Definition 2.4. Note the *hysteresis* in addition of new links  $(i, j)$  due to the proposed partitioning of agent  $i$ 's neighborhood. Note also the imposed proximity condition, necessary for addition or deletion of links. However, the final decision is controller dependent, hence, the notion of a *candidate* link in Definition 2.4.

Definition 2.4 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 takes place, respectively (Fig. 2).<sup>2</sup> Note that this partitioning of the 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 VI). Hence, Problem 1 can be restated as: *determine local motion controllers  $u_i(t) \in \mathbb{R}^p$  for all agents  $i$  so that the underlying network  $\mathcal{G}(t)$  is always connected.*

### III. DISTRIBUTED TOPOLOGY CONTROL: CHALLENGES & MACHINERY

Consider a dynamic graph  $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$  defined by the time varying set of edges  $\mathcal{E}(t)$ .<sup>3</sup> The goal in this section is to design *local* control laws that allow every agent to add or delete nearest neighbor links without violating connectivity of  $\mathcal{G}(t)$ . Although addition of links can only increase connectivity and does not introduce any significant challenge in controlling the topology of  $\mathcal{G}(t)$ , deletion of links is a nontrivial task. Since, connectivity is a global graph property, it is necessary that every agent has sufficient knowledge of the network structure in order to safely delete a link with a neighbor (Fig. 3). Such knowledge can be obtained through local *estimates* of the network topology (Section III-A), which, along with a *tie breaking* mechanism obtained by means of *gossip algorithms* and distributed *market-based* control (Section III-B), ensure connectivity even when combinations of multiple deletion requests could possibly violate it (Fig. 4).

<sup>2</sup>Dynamic graphs  $\mathcal{G}(t)$  as in Definition 2.4, are also called *proximity graphs*.

<sup>3</sup>In what follows,  $\mathcal{G}(t)$  is treated as a generic combinatorial object as in Section II-A. Integration with mobility and the underlying configuration space will be discussed in Section VI.

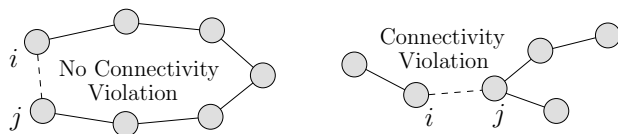


Fig. 3. Control challenges requiring knowledge of the network structure. Without such knowledge, deletion of a link  $(i, j)$  can either violate connectivity (right) or not (left).

TABLE I  
ELEMENTWISE LINK DYNAMICS.

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

#### A. Local Estimates of the Network Topology

As previously discussed, the goal in this section is to develop a distributed control framework that allows every agent  $i$  to obtain a local *spanning subgraph*<sup>4</sup> estimate  $\mathcal{G}_i(t) = (\mathcal{V}, \mathcal{E}_i(t))$  of the global network  $\mathcal{G}(t)$ , using information from its nearest neighbors  $\mathcal{N}_i(t) = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}(t)\}$  only.<sup>5</sup> For this, let  $A_i(t) = (a_{jk}^{[i]}(t))$  denote the adjacency matrix associated with the graph  $\mathcal{G}_i(t)$  at time  $t$ . Then, the dynamics of a link  $(j, k)$  can be expressed as (Table I)<sup>6</sup>

$$a_{jk}^{[i]}(t+1) = \neg(a_{jk}^{[i]}(t) \leftrightarrow v_{jk}^{[i]}(t)), \quad (3)$$

where  $v_{jk}^{[i]}(t) \in \{0, 1\}$  is such that  $v_{jk}^{[i]}(t) = 1$  if a control action is taken to add or delete link  $(j, k)$ .<sup>7</sup> In matrix form, the dynamics in equation (3) become

$$A_i(t+1) = \neg(A_i(t) \leftrightarrow V_i(t)), \quad (4)$$

where the control input  $V_i(t) = (v_{jk}^{[i]}(t))$  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  $E_i = \bigvee_{j \neq i} (e_i e_j^T \vee e_j e_i^T)$ , 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  $E_i \wedge (\neg A_i(t))$  captures new links  $(i, j)$  that agent  $i$  can create with agents  $j \notin \mathcal{N}_i(t)$ . Moreover, let  $A_1^{[i]}(t) \triangleq \bigvee_{j \in \mathcal{N}_i(t)} A_j(t)$  indicate existing links in the network, available by the 1-hop neighbors  $\mathcal{N}_i(t)$  of agent  $i$ . Then, the expression  $(\neg A_i(t)) \wedge A_1^{[i]}(t)$  captures existing links in the network that agent  $i$  is not aware of and are available by its neighbors. Hence, the expression (Table II)

$$F_i(t) \triangleq ((\neg A_i(t)) \wedge A_1^{[i]}(t)) \vee (E_i \wedge (\neg A_i(t))) \quad (5)$$

<sup>4</sup>Given any dynamic graph  $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ , we say that a graph  $\mathcal{G}_i(t) = (\mathcal{V}_i, \mathcal{E}_i(t))$  is a *subgraph* of  $\mathcal{G}(t)$ , if  $\mathcal{V}_i \subseteq \mathcal{V}$  and  $\mathcal{E}_i(t) \subseteq \mathcal{E}(t)$ . If  $\mathcal{V}_i = \mathcal{V}$ , we call  $\mathcal{G}_i(t)$  a *spanning subgraph* of  $\mathcal{G}(t)$ .

<sup>5</sup>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  $t \geq t_0$  (Section V).

<sup>6</sup>See Appendix I for an overview of boolean operations.

<sup>7</sup>The discrete time semantics in (3) are associated with *transition resets* in a resulting hybrid automaton for agent  $i$  and will be discussed in detail in Section IV.



Fig. 4. Control challenges due to multiple link deletions. In the absence of an agreement protocol, simultaneous deletion of links  $(i, j)$  and  $(k, l)$  violates connectivity.

TABLE II  
LINK DYNAMICS IN MATRIX FORM.

Entry	$A_i$	$A_1^{[i]}$	$\neg A_i$	$(\neg A_i) \wedge A_1^{[i]}$	$E_i$	$F_i$
$(i, j)$	1	1	0	0	1	0
	1	0	0	0	1	0
	0	1	1	1	1	1
	0	0	1	0	1	1
$(j, k)$ $j, k \neq i$	1	1	0	0	0	0
	1	0	0	0	0	0
	0	1	1	1	0	1
	0	0	1	0	0	0

captures all new links that agent  $i$  can add to  $A_i(t)$ , consisting of either existing links that agent  $i$  is not aware of, or links that agent  $i$  can create with agents  $j \notin \mathcal{N}_i(t)$ . On the other hand, the adjacency matrix  $A_i(t)$  includes all links in  $\mathcal{G}_i(t)$  that are candidates to be deleted from the network. Since  $F_i(t) \wedge A_i(t) = 0$ , we can decouple the control input  $V_i(t)$  into a component  $F_i(t) \wedge V_i^a(t)$  regulating link *additions* and a component  $A_i(t) \wedge V_i^d(t)$  regulating link *deletions*. Hence, the control input becomes

$$V_i(t) \triangleq (F_i(t) \wedge V_i^a(t)) \vee (A_i(t) \wedge V_i^d(t)). \quad (6)$$

Note that the component  $F_i(t) \wedge V_i^a(t)$  prevents  $A_i(t)$  from being updated with new links  $(j, k)$ , where  $j, k \neq i$ , if these are not provided by agent  $i$ 's neighbors. On the other hand, if  $((\neg A_i(t)) \wedge A_1^{[i]}(t)) \rightarrow V_i^a(t)$ , then all neighboring information is used to update  $A_i(t)$ . The following proposition shows that the local network dynamics (4)-(6) are essentially a *consensus* (with inputs) on the adjacency matrix estimates  $A_i(t)$ , providing every agent with a *rough* picture of the overall network, as desired.

*Proposition 3.1* ([33]): Assume a fixed network  $\mathcal{G}$  with corresponding adjacency matrix  $A$  and initialize all network estimates  $A_i(t_0)$  with nearest neighbor links, i.e.,  $A_i(t_0) = E_i \wedge A$ . Let, further,  $V_i^a(t) = ((\neg A_i(t)) \wedge A_1^{[i]}(t))$  and  $V_i^d(t) = \mathbf{0}$  for all time  $t \geq t_0$ , so that no new links are added or deleted in  $\mathcal{G}$ . Then,

$$A_i(t+1) = \bigvee_{j \in \mathcal{N}_i} (A_i(t) \vee A_j(t))$$

and  $A_i(t_0 + n - 1) = A$  for all agents  $i$ .

### B. Controlling Addition and Deletion of Links

Given the local network dynamics (4), the main challenge now is to determine control inputs  $V_i^a(t)$  and  $V_i^d(t)$  that ensure connectivity of each estimate  $\mathcal{G}_i(t)$  for all time  $t$ . If the spanning subgraph requirement  $\mathcal{G}_i(t) \subseteq \mathcal{G}(t)$  holds (Section V), connectivity of  $\mathcal{G}_i(t)$  for all  $i$  implies connectivity of the overall network  $\mathcal{G}(t)$ .

Regarding links that agent  $i$  can add in  $\mathcal{G}_i(t)$ , we require that the control  $V_i^a(t) = (v_{jk}^{[i]a}(t))$  satisfies  $((\neg A_i(t)) \wedge A_1^{[i]}(t)) \rightarrow V_i^a(t)$ , so that  $A_i(t)$  is updated with all existing links in the network that agent  $i$  is not aware of. Furthermore,  $V_i^a(t)$  should also capture new links that agent  $i$  can create with

### Algorithm 1 Auction Mechanism for Agent $i$

- 1: Compute the set of *safe* neighbors  $\mathcal{S}_i(t)$  such that if a link  $(i, j)$  with  $j \in \mathcal{S}_i(t)$  is deleted from  $\mathcal{E}_i(t)$ , then  $\mathcal{G}_i(t)$  remains connected (see Remark 2.3), i.e.,  $\mathcal{S}_i(t) \triangleq \{j \in \mathcal{N}_i^d(t) \mid \lambda_2(L(\mathcal{E}_i(t) \setminus (i, j))) > k - 1\}$ .
- 2: Initialize a request  $r_i \triangleq [i \ g(\mathcal{S}_i(t)) \ b]^T \in \mathbb{R}^3$  consisting of the link  $(i, g(\mathcal{S}_i(t)))$  to be deleted and a bid  $b \in \mathbb{R}$ , such that  $b > 0$  if  $\mathcal{S}_i(t) \neq \emptyset$  and  $b = 0$  otherwise, indicating how “important” this request is.
- 3: Initialize a set of max-bids  $\mathcal{M}_i(t) \triangleq \{r_i(t)\} \in 2^{\mathbb{R}^3}$  and a binary vector of tokens  $T_i(t) \triangleq [0 \ \dots \ 1_i \ \dots \ 0]^T \in \{0, 1\}^n$  indicating the start of an auction.
- 4: **while**  $(\bigwedge_{j=1}^n T_{ij}(t)) = 0$  **do**
- 5:   Collect tokens from neighbors only, i.e.,  
 $T_i(t+1) := T_i(t) \vee (\bigvee_{j \in \mathcal{N}_i(t)} T_j(t))$
- 6:   Apply a max-consensus update on  $\mathcal{M}_i(t)$ , i.e.,  
 $\mathcal{M}_i(t+1) := \{r_j \mid j = \underset{r_k \in \bigcup_{l \in \{\mathcal{N}_i(t), i\}} \mathcal{M}_l(t)}{\operatorname{argmax}} \{r_{k3}\}\}$
- 7: **end while**
- 8: Compute the *winner* link  $w_i(t)$  of the auction,  
 $w_i(t) \triangleq \{(r_{j1}, r_{j2}) \mid r_j \in \mathcal{M}_i(t)\}$

agents  $j \notin \mathcal{N}_i$ . These objectives can be achieved by letting

$$v_{jk}^{[i]a}(t) \triangleq \underbrace{((j \neq i) \wedge (k \neq i))}_{\text{add all existing links provided by neighbors}} \vee \underbrace{(x_k(t) \in \mathcal{B}_r(x_j(t)))}_{\text{maintain current neighbors and add new neighbors}}, \quad (7)$$

where  $\mathcal{B}_\rho(x) = \{y \in \mathbb{R}^p \mid \|y - x\|_2 < \rho\}$  denotes an open ball of radius  $\rho > 0$  centered at  $x \in \mathbb{R}^p$ , and  $r > 0$  is as in Definition 2.4.<sup>8</sup> Note that  $((\neg A_i(t)) \wedge A_1^{[i]}(t)) \rightarrow V_i^a(t)$ , hence,  $A_i(t)$  is updated with all neighbor information.

Unlike link additions, deletion of nearest neighbor links is a much more challenging task since, although knowledge of the estimate  $\mathcal{G}_i(t)$  allows every agent  $i$  to determine adjacent links that if deleted individually, network connectivity is preserved (Fig.3), it is not sufficient for dealing with simultaneous link deletions by multiple non-adjacent agents that may disconnect  $\mathcal{G}(t)$  (Fig. 4). For this, we require that at most one link can be deleted from  $\mathcal{G}(t)$  at a time<sup>9</sup> and employ *market-based* control to achieve agreement of all agents regarding the link that is to be deleted. In particular, let  $\mathcal{N}_i^d(t) \triangleq \mathcal{N}_i(t) \setminus \{j \in \mathcal{V} \mid x_j(t) \in \mathcal{B}_r(x_i(t))\}$  denote a subset of neighbors that agent  $i$  desires to delete a link with,<sup>10</sup> and define the function  $g : 2^{\mathbb{R}} \rightarrow \mathbb{R}$  with

$$g(X) \triangleq \begin{cases} x \in X & \text{if } X \neq \emptyset \\ 0 & \text{otherwise} \end{cases},$$

where  $x \in X \subseteq \mathbb{R}$  can be chosen according to any policy, deterministic or not. Then, the proposed auction algorithm for agent  $i$  is described in Algorithm 1 and with every iteration, it outputs a winning link  $w_i(t)$  corresponding to the highest

<sup>8</sup>Although not necessary to introduce Definition 2.4 of a proximity dynamic graph yet, we do so to avoid complicating further notations as well as to achieve a smoother introduction to the case of mobile agents (Section VI).

<sup>9</sup>Extension to multiple link deletions is considered in [24].

<sup>10</sup>The set  $\mathcal{N}_i^d(t)$  typically consists of neighbors that due to their distance from agent  $i$  require high communication power.

bid sent in the network. The control input that regulates link deletions  $V_i^d(t) = (v_{jk}^{[i]d}(t))$  can then be defined as

$$v_{jk}^{[i]d}(t) \triangleq (w_i(t) = (j, k)) \wedge (|w_i(t)| = 1). \quad (8)$$

Note that if  $|w_i(t)| > 1$ , then there is either a tie in the maximum bids or all bids sent in the network were zero. In every case, equation (8) implies that  $V_i^d(t) = \mathbf{0}$  for all agents  $i$  and no link is deleted from any estimate edge set  $\mathcal{E}_i(t)$ , which ensures the requirement of at most one link deletion with every auction. Other deletion requests are considered in subsequent auctions and so the proposed market-based control framework consists of a sequence of auctions, each one of which results in at most one link deletion from the network. Clearly, the existence of some notion of *synchronization* of all agents to the same auction is necessary for correctness of the proposed approach. This requirement becomes even more important in the presence of communication time delays (Section IV).

*Remark 3.2 (Choosing the Bids):* 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 constraint properties of the overall network.

*Remark 3.3 (Convergence of max-Consensus):* Note that the condition  $(\bigwedge_{j=1}^n T_{ij}(t)) = 1$  in Algorithm 1, clearly implies convergence of the max-consensus on sets  $\mathcal{M}_i(t)$  to the global maximum over the network. Alternatively, convergence of the max-consensus algorithm could be decided after  $n - 1$  updates of the sets  $\mathcal{M}_i(t)$ , where  $n - 1$  corresponds to the worst case diameter of a network on  $n$  agents. The advantage of using the first approach over the second one is twofold. First, using tokens exploits the network structure and can result in more efficient updating.<sup>11</sup> Second, the first approach can deal with communication time delays, where the time required for  $n - 1$  updates of Algorithm 1 can be significantly different for different agents, preventing convergence of all agents to a common outcome. It is also worth noting that the memory and communication overhead for transmitting the binary tokens is minimal (linear in the number of agents  $n$ ).

*Remark 3.4 (Computational Complexity):* Note that computation of the spectrum of a matrix has worst case complexity  $O(n^3)$ , where  $n$  is the size of the matrix [29]. This complexity can, however, be reduced to  $O(n)$  for sparse symmetric matrices [30], as is the laplacian matrix  $L(t)$  in the case of large networks, commonly appearing in the proposed framework. Consequently, dealing with eigenvalues does not introduce significant computational overhead, which makes our approach scalable to large size networks.

#### IV. MODELING THE AGENTS IN THE PRESENCE OF TIME DELAYS

The discrete topology control machinery introduced in Section III gives rise to a hybrid model for every agent  $i$  (Fig. 5), defined by the composition (or product)  $\mathbb{T}_i \times \mathbb{A}_i$  of a *topology*

<sup>11</sup>For instance, in the case of a complete graph, one update is sufficient for convergence of the max-consensus algorithm.

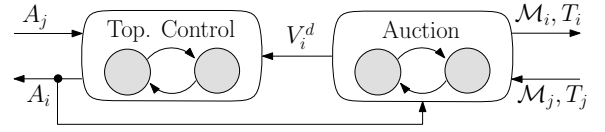


Fig. 5. Hybrid model  $\mathbb{T}_i \times \mathbb{A}_i$  for agent  $i$  that consists of the composition of a *topology control* and an *auction* automaton, resulting from the analysis in Section III.

*control*  $\mathbb{T}_i$  and an *auction* automaton  $\mathbb{A}_i$ , respectively [27]. The topology control automaton of agent  $i$  is responsible for updating its network estimate  $A_i$  with addition and deletion of links (Section III-A). For this, it requires the control input  $V_i^d$  that regulates link deletions, as well as the network estimates  $A_j$  of agent  $i$ 's neighbors in order to compute the control input  $V_i^a$  that regulates link additions (Section III-B). The control input  $V_i^d$  is provided by the auction automaton and is computed using the max-bid sets  $\mathcal{M}_j$  and tokens  $T_j$  of agent  $i$ 's neighbors (Algorithm 1). Note that in the proposed hybrid system, all variables are considered *shared* [27], however, the only variables that are practically needed are the ones provided by every agent's neighbors, which guarantees the local nature of the proposed control framework.

Implementation of the above hybrid system relies on information exchange between neighboring agents in the form of messages

$$\text{Msg}[i] \triangleq \{A_i, \mathcal{M}_i, T_i\}$$

containing their network estimates  $A_i$ , max-bid sets  $\mathcal{M}_i$  and tokens  $T_i$ .<sup>12</sup> Clearly, such messages are neither received simultaneously nor instantaneously. Instead, they are queued and are received with a time delay  $\tau_i > 0$  and in an order that may vary according to the frequency of transmission of each agent.<sup>13</sup> To address these challenges, a notion of *synchronization* is required among the two individual automata consisting a single hybrid agent, as well as among all hybrid agents consisting the overall hybrid multiagent system. In the absence of a common global clock, the desired synchronization is ideally *event triggered*, where by a triggering event we understand the time instant that a message  $\text{Msg}[j]$  is received by any of agent  $i$ 's neighbors  $j \in \mathcal{N}_i$ . The rest of this section is devoted in defining formally the aforementioned automata and discussing how they can be synchronized into a distributed multiagent system, in the presence of time delays.

##### A. The Hybrid Agent

The following notion of a *predicate* enables us to formally define the aforementioned automata.

*Definition 4.1 (Predicate):* Let  $X = \{x_1, \dots, x_n\}$  be a finite set of variables. We define a predicate  $\psi(X)$  over  $X$  to be a finite conjunction of strict or non-strict inequalities over  $X$ . We denote the set of all predicates over  $X$  by  $\text{Pred}(X)$ .

In other words, a predicate is a logical formula. For example, the predicate  $\psi(X) = (\|x - x_0\|_2 < r)$  over the set

<sup>12</sup>The memory and communication cost associated with messages  $\text{Msg}[i]$  is  $O(n^2)$ , hence, our approach is scalable to large networks.

<sup>13</sup>For instance, for a set of neighbors  $\mathcal{N}_i = \{1, 2, 3\}$ , the order of the messages received could be a sequence of the form  $\{1, 1, 2, 1, 3, 2, 2, 1, \dots\}$ .

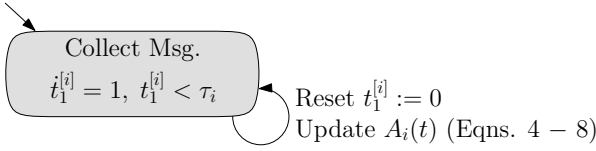


Fig. 6. Topology control automaton  $\mathbb{T}_i$  of agent  $i$  responsible for updating its network estimate  $A_i$ .

of variables  $X \in \mathbb{R}^p$  returns 1 if  $x$  belongs in the open ball  $\|x - x_0\|_2 < r$  and 0 otherwise. Hence, the topology control automaton, responsible for updating the network estimate  $A_i$  of agent  $i$ , can be defined as follows [27] (Fig. 6).<sup>14</sup>

*Definition 4.2 (Topology Control Automaton):* We define the topology control automaton of agent  $i$  by the tuple  $\mathbb{T}_i \triangleq (X_{\mathbb{T}_i}, V_{\mathbb{T}_i}, E_{\mathbb{T}_i}, \text{inv}, \text{init}, \text{guard}, \text{reset}, \text{flow})$ , where

- $X_{\mathbb{T}_i} \triangleq \{t_1^{[i]}, A_i\}$  denotes the set of owned state variables with  $t_1^{[i]} \in \mathbb{R}_+$  and  $A_i \in \{0, 1\}^{n \times n}$ .
- $V_{\mathbb{T}_i} \triangleq \{C, D\}$  denotes the finite set of control modes.<sup>15</sup>
- $E_{\mathbb{T}_i} \triangleq \{(C, C)\}$  denotes the set of control switches.
- $\text{inv} : V_{\mathbb{T}_i} \rightarrow \text{Pred}(\cup_{j=1}^n (X_{\mathbb{T}_j} \cup X_{\mathbb{A}_j}))$  with  $\text{inv}(C) \triangleq \{t_1^{[i]} < \tau_i\}$  denotes the invariant conditions.
- $\text{init} : V_{\mathbb{T}_i} \rightarrow \text{Pred}(X_{\mathbb{T}_i})$  with  $\text{init}(C) \triangleq \text{true}$  denotes the set of initial conditions.
- $\text{guard} : E_{\mathbb{T}_i} \rightarrow \text{Pred}(\cup_{j=1}^n (X_{\mathbb{T}_j} \cup X_{\mathbb{A}_j}))$  with  $\text{guard}((C, C)) \triangleq \{t_1^{[i]} = \tau_i\}$  denotes the set of transition guards.
- $\text{reset} : E_{\mathbb{T}_i} \rightarrow \text{Pred}(\cup_{j=1}^n (X_{\mathbb{T}_j} \cup X_{\mathbb{A}_j}))$  with  $\text{reset}((C, C)) \triangleq \{t_1^{[i]} := 0, A_i := \text{Eqn. 4}\}$  denotes the set of transition resets.
- $\text{flow} : V_{\mathbb{T}_i} \rightarrow \text{Pred}(\dot{X}_{\mathbb{T}_i} \cup_{j=1}^n (X_{\mathbb{T}_j} \cup X_{\mathbb{A}_j}))$  with  $\text{flow}(C) \triangleq \{t_1^{[i]} = 1\}$ , denotes the flow conditions of the hybrid automaton that constrain the first time derivatives of the system variables in mode  $v \in V_{\mathbb{T}_i}$ .

As discussed above, transitions in  $\mathbb{T}_i$  are *event triggered*, where the triggering event corresponds to the time instant when agent  $i$  receives messages  $\text{Msg}[j]$  from its neighbors  $j \in \mathcal{N}_i$ . The elapsed time between any two such events is strictly positive and is denoted by  $\tau_i > 0$ , while the information contained in these messages is used to compute the control input  $V_i^a$  that regulates link additions. The control input  $V_i^d$  responsible for link deletions is provided by the auction automaton defined as follows (Fig. 7).

*Definition 4.3 (Auction Automaton):* We define the auction automaton of agent  $i$  by the tuple  $\mathbb{A}_i \triangleq (X_{\mathbb{A}_i}, V_{\mathbb{A}_i}, E_{\mathbb{A}_i}, \text{inv}, \text{init}, \text{guard}, \text{reset}, \text{flow})$ , where

- $X_{\mathbb{A}_i} \triangleq \{t_2^{[i]}, T_i, \mathcal{M}_i\}$  denotes the set of owned state variables with  $t_2^{[i]} \in \mathbb{R}_+$ ,  $T_i \in \mathbb{B}^n$  and  $\mathcal{M}_i \in 2^{\mathbb{R}^3}$ .
- $V_{\mathbb{A}_i} \triangleq \{S, B, U, D\}$  denotes the finite set of control modes.<sup>16</sup>

<sup>14</sup>To simplify notation, we hereafter drop the dependence of the state variables on time  $t$ .

<sup>15</sup>The shorthand notation stands for  $C \triangleq \text{Collect Msg.}$  and  $D \triangleq \text{Dummy.}$

<sup>16</sup>The shorthand notation stands for  $S \triangleq \text{Start}$ ,  $B \triangleq \text{Bid}$ ,  $U \triangleq \text{Update}$  and  $D \triangleq \text{Decide.}$

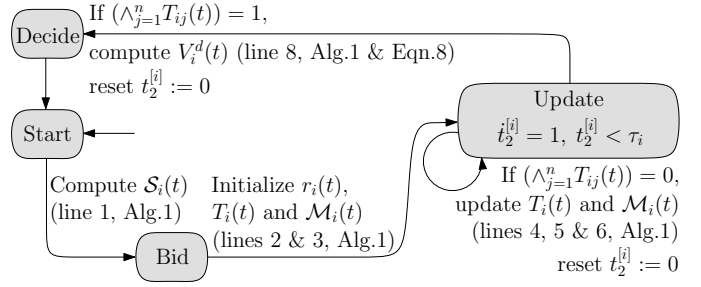


Fig. 7. Auction automaton  $\mathbb{A}_i$  of agent  $i$  responsible for providing the topology control automaton with the control input  $V_i^d$  that regulates link deletions.

- $E_{\mathbb{A}_i} \triangleq \{(S, B), (B, U), (U, U), (U, D), (D, S)\}$  denotes the set of control switches.
- $\text{inv} : V_{\mathbb{A}_i} \rightarrow \text{Pred}(\cup_{j=1}^n (X_{\mathbb{T}_j} \cup X_{\mathbb{A}_j}))$  with  $\text{inv}(U) \triangleq \{t_2^{[i]} < \tau_i\}$  and  $\text{inv}(e) \triangleq \text{true}$  for all  $e \in V_{\mathbb{A}_i} \setminus \{U\}$  denotes the invariant conditions.
- $\text{init} : V_{\mathbb{A}_i} \rightarrow \text{Pred}(X_{\mathbb{A}_i})$  with  $\text{init}(S) \triangleq \text{true}$  denotes the set of initial conditions.
- $\text{guard} : E_{\mathbb{A}_i} \rightarrow \text{Pred}(\cup_{j=1}^n (X_{\mathbb{T}_j} \cup X_{\mathbb{A}_j}))$  with,
  - $\text{guard}((U, U)) \triangleq \{(\wedge_{j=1}^n T_{ij}) = 0\}$ ,
  - $\text{guard}((U, D)) \triangleq \{(\wedge_{j=1}^n T_{ij}) = 1\}$ ,denotes the set of transition guards.
- $\text{reset} : E_{\mathbb{A}_i} \rightarrow \text{Pred}(\cup_{j=1}^n (X_{\mathbb{T}_j} \cup X_{\mathbb{A}_j}))$  with,
  - $\text{reset}((S, B)) \triangleq \{S_i := \text{Alg. 1}\}$ ,
  - $\text{reset}((B, U)) \triangleq \{r_i, T_i, \mathcal{M}_i := \text{Alg. 1}\}$ ,
  - $\text{reset}((U, U)) \triangleq \{t_2^{[i]} := 0, T_i, \mathcal{M}_i := \text{Alg. 1}\}$ ,
  - $\text{reset}((U, D)) \triangleq \{t_2^{[i]} := 0, V_i^d := \text{Eqn. 8}\}$ ,
  - $\text{reset}((D, S)) \triangleq \{\text{clear variables for new auction}\}$ ,denotes the set of transition resets.
- $\text{flow} : V_{\mathbb{A}_i} \rightarrow \text{Pred}(\dot{X}_{\mathbb{A}_i} \cup_{j=1}^n (X_{\mathbb{T}_j} \cup X_{\mathbb{A}_j}))$  with  $\text{flow}(U) \triangleq \{t_2^{[i]} = 1\}$ , denotes the flow conditions of the hybrid automaton that constrain the first time derivatives of the system variables in mode  $v \in V_{\mathbb{A}_i}$ .

Note that the auction automaton  $\mathbb{A}_i$  of agent  $i$  consists an implementation of Algorithm 1 that provides  $\mathbb{T}_i$  with the control input  $V_i^d$  that regulates link deletions. As with the topology control automaton  $\mathbb{T}_i$ , transitions in the auction automaton  $\mathbb{A}_i$  are triggered upon receipt of a message and the elapsed time between any two such events is  $\tau_i > 0$ . This implies that transitions  $(C, C)_{\mathbb{T}_i}$  of the topology control automaton  $\mathbb{T}_i$  are always *synchronized* with either transitions  $(U, U)_{\mathbb{A}_i}$  or  $(U, D)_{\mathbb{A}_i}$  of the auction automaton  $\mathbb{A}_i$ , synchronizing the two automata, as desired. Composition of the automata  $\mathbb{A}_i$  and  $\mathbb{T}_i$  captures control in the discrete space of graphs.

## B. Synchronization of the Hybrid Agents

The presence of time delays in the network, as well as the network topology which imposes multi-hop communication patterns between non-neighboring agents, can result in the agents reaching a decision on an auction asynchronously. This would imply that initialization of all subsequent auctions would also be asynchronous and could result in mixing of information between consecutive auctions, preventing all

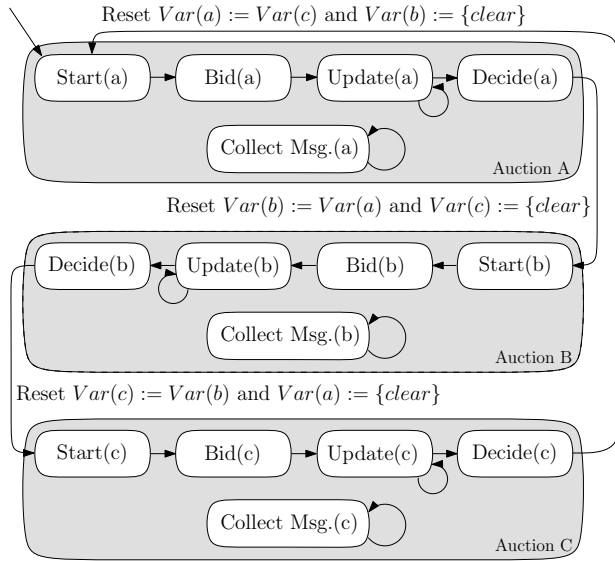


Fig. 8. Creating three copies of the same auction guarantees synchronization of all agents to the same auction and common auction outcomes for all agents.  $Var(z)$  denotes the set of all variables associated with auction  $z = a, b, c$ .

agents from reaching a common outcome for the same auction. Hence, correctness of the proposed distributed control framework relies on some notion of synchronization of all agents to the same auction. In the absence of a common global clock, this synchronization can be achieved by creating three identical copies of the auction protocol described in Algorithm 1 that only differ on their labels (Fig. 8). In particular, we create three copies of all variables in  $\mathbb{A}_i$  and  $\mathbb{T}_i$ , label them in the set  $\{a, b, c\}$  and require that a sequence of auctions is always of the form  $\{a, b, c, a, b, c, \dots\}$  by imposing the transitions  $(D_a, S_b)_{\mathbb{A}_i}$ ,  $(D_b, S_c)_{\mathbb{A}_i}$  and  $(D_c, S_a)_{\mathbb{A}_i}$ , according to Fig. 8.

To understand how this scheme ensures synchronization of all agents on the same auction, consider any agent  $i$  and suppose it is in auction  $a$ . Clearly, for agent  $i$  to transition to auction  $b$  it is necessary that all other agents are also in auction  $a$ , since otherwise agent  $i$  will be missing bids from the agents that are not in auction  $a$  yet (currently in auction  $c$ ) and Algorithm 1 will not be able to converge. Once agent  $i$  transitions to auction  $b$ , it initializes all variables for that auction with the latest values from auction  $a$ , while it maintains the variables of auction  $a$  for agents that are still in auction  $a$  and it clears all variables of auction  $c$  since, no agent is in this auction any more. In other words, *faster* agents are forced to wait for their *slower* peers before transitioning to a subsequent auction, which guarantees synchronization of all agents to the same auction.

## V. CORRECTNESS OF THE OVERALL SYSTEM

Composition of all elementary agents  $\mathbb{T}_i \times \mathbb{A}_i$  results in the overall product system

$$\mathbb{S} \triangleq \times_{i=1}^n (\mathbb{T}_i \times \mathbb{A}_i),$$

which captures properties of the whole dynamic network.<sup>17</sup> In particular, in Section V-A we show that the proposed

distributed hybrid system  $\mathbb{S}$  ensures synchronization of all agents to the same auction and agreement on the link that is to be deleted, while in Section V-B we show that the overall system  $\mathbb{S}$  guarantees connectivity of the mobile network.

### A. Synchronization & Market-Based Control

Let  $\{z_k\}_{k=1}^\infty = \{a, b, c, a, b, c, \dots\}$  denote the sequence of auctions for any agent  $\mathbb{T}_i \times \mathbb{A}_i$  (Fig. 8). Due to possible time delays, the product system  $\mathbb{S}$  is not necessarily synchronized in the same auction for all agents and all time, which can cause the overall market-based coordination framework to fail. This section is devoted in studying the *auction dynamics* in system  $\mathbb{S}$  and showing that they guarantee synchronization of all agents to the same auction for all time. We achieve this goal by studying the associated *token dynamics*, which explicitly indicate the beginning and termination of the corresponding auction from the perspective of agent  $i$ . In particular, let

$$T_{\vee i}^{z_k} \triangleq \vee_{j=1}^n T_{ij}^{z_k} \quad \text{and} \quad T_{\wedge i}^{z_k} \triangleq \wedge_{j=1}^n T_{ij}^{z_k}$$

indicate the existence of at least one token in  $T_i^{z_k}$  or the existence of exactly  $n$  tokens in  $T_i^{z_k}$ , respectively. Then, we have the following results.

*Proposition 5.1* ([33]): For any time  $t$ , suppose there exists an agent  $i$  such that  $T_{\vee i}^{z_k}(t) = 1$  for any auction  $z_k$ . Then,  $T_{\vee j}^{z_k}(t) = 1$  for all agents  $j$  with  $j \neq i$ .

*Proposition 5.2* ([33]): For any time  $t$ , suppose there exists an agent  $i$  such that  $T_{\vee i}^{z_k}(t) = 1$  and  $T_{\wedge i}^{z_k}(t) = 0$ , while  $T_{\vee j}^{z_{k+1}}(t) = 1$  for all agents  $j$  with  $j \neq i$  and any consecutive auctions  $z_k$  and  $z_{k+1}$ . Then,  $T_{\wedge j}^{z_{k+1}}(t) = 0$  and  $T_{\wedge j}^{z_k}(t) = 1$  for all  $j \neq i$ .

Proposition 5.2 equivalently states that *faster* agents periodically wait for their *slower* peers, which implies a form of synchronization among all agents. Now, let

$$T_{\vee}^{z_k}(t) \triangleq \vee_{i=1}^n T_{ii}^{z_k}(t) \quad \text{and} \quad T_{\wedge}^{z_k}(t) \triangleq \wedge_{i=1}^n T_{ii}^{z_k}(t)$$

denote the existence of at least one token or exactly  $n$  tokens in auction  $z_k$ , respectively, from the perspective of a global observer. Clearly, if  $T_{\vee}^{z_k}(t) = 1$  there exists at least one agent  $i$  with  $T_{\vee i}^{z_k}(t) = 1$ , while if  $T_{\vee}^{z_k}(t) = 0$ , then  $T_{\vee i}^{z_k}(t) = 0$  for all agents  $i$ . Denote, further, by  $t_{I_k}$  the time instant that the *first* token for auction  $z_k$  has just been sent and by  $t_{B_k}$  the time instant that the *last* token for auction  $z_k$  has just been sent. Similarly, let  $t_{F_k}$  denote the time instant that the *last* token for auction  $z_k$  has just been cleared (Fig. 8). In terms of the global variables  $T_{\vee}^{z_k}(t)$  and  $T_{\wedge}^{z_k}(t)$  for auction  $z_k$ , these time instants can be expressed as  $t_{I_k} \triangleq \min\{t \mid T_{\vee}^{z_k}(t) = 1\}$ ,  $t_{B_k} \triangleq \min\{t \mid T_{\wedge}^{z_k}(t) = 1\}$  and  $t_{F_k} \triangleq \max\{t \mid T_{\vee}^{z_k}(t) = 1\}$ , respectively. Then, Proposition 5.2 results in:

*Proposition 5.3* ([33]): Let  $z_{k-2}$ ,  $z_{k-1}$ ,  $z_k$  and  $z_{k+1}$  be any sequence of auctions. Then,  $t_{F_{k-2}} < t_{B_k} < t_{I_{k+1}} < t_{F_{k-1}}$ . Hence, the sequence of auctions  $\{z_k\}_{k=1}^\infty$  is according to

$$t_{I_k} \leq t_{F_{k-2}} \leq t_{B_k} \leq t_{I_{k+1}} \leq t_{F_{k-1}} \leq t_{B_{k+1}} \leq t_{I_{k+2}} \leq t_{F_k},$$

which, since  $t_{F_{k-2}} < t_{I_{k+1}}$ , implies that no contributions from past auctions can be made in updating the variables of current auctions, despite recycling of the auction labels according to  $z_{k-2} = z_{k+1} \pmod{3}$ . This observation can be used to

<sup>17</sup>For a formal definition of the overall system  $\mathbb{S}$ , see [33].

show correctness of the proposed market-based coordination framework. In particular, we have the following result.

**Proposition 5.4 ([33]):** For any agent  $i$  and any auction  $z_k$ , eventually  $T_{\wedge_i}^{z_k}(t) = 1$ . Moreover, all agents  $i$  with  $T_{\wedge_i}^{z_k}(t) = 1$  share identical max-bid sets  $\mathcal{M}_i^{z_k}(t)$ .

Proposition 5.4 equivalently implies that the max-consensus algorithm (Algorithm 1) on the max-bid sets  $\mathcal{M}_i^{z_k}(t)$  converges to a common link for all agents  $i$  and every auction  $z_k$  (Eqn. 8). Hence, synchronization guarantees correctness of market-based control, even in the presence of time delays.

### B. Correctness of Connectivity Control

As in Section V-A, observe that the transition resets  $\text{reset}((D_{z_k}, S_{z_{k+1}})_{\mathbb{A}_i})$  that clear all variables of auction  $z_{k-1}$  (Fig. 8), combined with synchronization of the transitions  $(C_{z_k}, C_{z_k})_{\mathbb{T}_i}$  and  $(U_{z_k}, D_{z_k})_{\mathbb{A}_i}$  in agent  $i$ , result in  $A_i^{z_k}(t) = \mathbf{0}$  for all agents  $i$  and all time  $t > t_{F_k}$ . Hence, although  $z_{k-3} = z_k \pmod{3}$ , the network estimates  $A_i^{z_k}(t)$  are updated exclusively from auction  $z_k$ , as desired. The following result shows that the outcome of every auction  $z_k$  is at most one link  $w_{z_k}^{[i]}(t_s)$  in  $\mathcal{E}_i^{z_k}(t_s)$  that is deleted from the network estimate  $\mathcal{E}_i^{z_{k+1}}(t_{s+1})$  of the subsequent auction, i.e.,  $w_{z_k}^{[i]}(t_s) \notin \mathcal{E}_i^{z_{k+1}}(t_{s+1})$ .

**Proposition 5.5 ([33]):** For any time  $t_s$  and any agent  $i$ , consider the transition  $(v_S^s, v_S^{s+1})$  that is due to  $(D_{z_k}, S_{z_{k+1}})_{\mathbb{A}_i}$ . Then,  $V_{z_k}^{[i]d}(t_s) \rightarrow (\neg A_i^{z_{k+1}}(t_{s+1}))$ .

We next provide a relation between the edge sets  $\mathcal{E}_i^{z_k}(t_s)$  and  $\mathcal{E}_i^{z_k}(t_{s+1})$  after any transition  $(v_S^s, v_S^{s+1})$  that is due to a self-transition  $(C_{z_k}, C_{z_k})_{\mathbb{T}_i}$ .

**Proposition 5.6 ([33]):** For any time  $t_s$  and any agent  $i$ , consider the transition  $(v_S^s, v_S^{s+1})$  that is due to  $(C_{z_k}, C_{z_k})_{\mathbb{T}_i}$ . Then,  $A_i^{z_k}(t_s) \rightarrow (A_i^{z_k}(t_{s+1}) \vee V_{z_k}^{[i]d}(t_s))$ .

Proposition 5.6 equivalently implies that with every transition  $(v_S^s, v_S^{s+1})$  of the product system  $\mathbb{S}$  that is due to a transition  $(C_{z_k}, C_{z_k})_{\mathbb{T}_i}$ , we have  $\mathcal{E}_i^{z_k}(t_s) \subseteq \mathcal{E}_i^{z_k}(t_{s+1}) \cup \{w_{z_k}^{[i]}(t_s)\}$ . This result can be used to derive similar results for the *global* edge set  $\mathcal{E}(t)$ . For this, we need to introduce some further notation. In particular, for all time  $t \in [t_{B_k}, t_{B_{k+1}})$  we can define the *global* winning link of auction  $z_k$  by

$$w_{z_k}(t) \triangleq \{(r_{i1}, r_{i2}) \mid i = \operatorname{argmax}_{1 \leq j \leq n} \{r_{j3} \mid r_j \in \cup_{l=1}^n \mathcal{M}_l^{z_k}(t)\}\}$$

and, as in equation (8), denote by  $Q_{z_k}(t) = (q_{jk}^{z_k}(t))$ , where  $q_{jk}^{z_k}(t) \triangleq (w_{z_k}(t) = (j, k)) \wedge (|w_{z_k}(t)| = 1)$ , the boolean matrix indicating the existence of a winning link for auction  $z_k$ . By Propositions 5.3 and 5.4, the set  $\cup_{l=1}^n \mathcal{M}_l^{z_k}(t)$  is fixed for all time  $t \in [t_{B_k}, t_{F_k}]$  and hence, for all time  $t \in [t_{B_k}, t_{B_{k+1}})$ , since  $[t_{B_k}, t_{B_{k+1}}) \subseteq [t_{B_k}, t_{F_k}]$ . Thus, the winning link  $w_{z_k}(t)$  and matrix  $Q_{z_k}(t)$  are also fixed for all time  $t \in [t_{B_k}, t_{B_{k+1}})$  and exclusively associated with the outcome of auction  $z_k$ . Furthermore,  $V_{z_k}^{[i]d}(t) \rightarrow Q_{z_k}(t)$  for all time  $t \in [t_{B_k}, t_{B_{k+1}})$  and any agent  $i$ . Finally, let  $A(t) \triangleq \vee_{i=1}^n (E_i \wedge A_i^{z_k}(t))$  denote the adjacency matrix of the *global* graph. Then, we have the following result.

**Proposition 5.7 ([33]):** For any time  $t_s \in [t_{B_k}, t_{B_{k+1}})$  and any agent  $i$ , consider the transition  $(v_S^s, v_S^{s+1})$  that is due to  $(C_{z_k}, C_{z_k})_{\mathbb{T}_i}$ . Then,  $A(t_s) \rightarrow (A(t_{s+1}) \vee Q_{z_k}(t_s))$ .

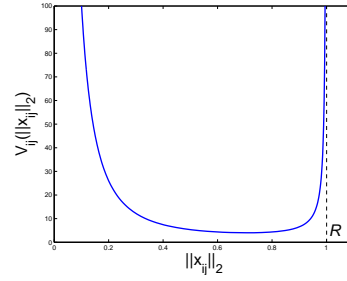


Fig. 9. Plot of the potential  $\varphi_{ij}(x_{ij})$  for  $R = 1$ .

Proposition 5.7 implies that any transition  $(v_S^s, v_S^{s+1})$  of the product system  $\mathbb{S}$  that is due to a transition  $(C_{z_k}, C_{z_k})_{\mathbb{T}_i}$  at time  $t_s \in [t_{B_k}, t_{B_{k+1}})$ , results in  $\mathcal{E}(t_s) \subseteq \mathcal{E}(t_{s+1}) \cup \{w_{z_k}(t_s)\}$ . This leads to the following result.

**Proposition 5.8 ([33]):** For any time  $t_s \in [t_{B_k}, t_{B_{k+1}})$  and any agent  $i$ , consider the transition  $(v_S^s, v_S^{s+1})$  that is due to  $(C_{z_k}, C_{z_k})_{\mathbb{T}_i}$ . Then,  $A_i^{z_k}(t_{s+1}) \rightarrow (A(t_{s+1}) \vee Q_{z_k}(t_s))$ . Moreover, if  $t = t_{s+1}$  is such that  $v_{\mathbb{A}_i}^{s+1} = S_{z_k}$ , then  $A_i^{z_k}(t_{s+1}) \rightarrow A(t_{s+1})$ .

Proposition 5.8 equivalently implies that  $\mathcal{E}_i^{z_k}(t_s) \subseteq \mathcal{E}(t_s)$  for any time instant  $t_s \in [t_{B_k}, t_{B_{k+1}})$  such that  $v_{\mathbb{A}_i}^s = S_{z_k}$ . In other words, when agent  $i$  selects a link  $(i, j)$  to delete from  $\mathcal{E}_i(t_s)$ , the estimate  $\mathcal{G}_i(t_s)$  is a spanning subgraph of  $\mathcal{G}(t_s)$ . This leads to our main result.

**Theorem 5.9:** Assuming that all changes in the network topology  $\mathcal{G}(t)$  are controllable (due to  $\mathbb{S}$ ) as well as that  $\mathcal{G}(t_0)$  is initially connected, the product system  $\mathbb{S}$  guarantees connectivity of the dynamic network  $\mathcal{G}(t)$  for all time  $t \geq t_0$ .

*Proof:* See Appendix II. ■

## VI. INTEGRATION WITH AGENT MOBILITY

A critical requirement for correctness of the discrete topology controller  $\mathbb{S}$  (Section V) is that the network structure does not change between consecutive updates in the network topology. In the case of stationary agents, this assumption translates to the absence of random link additions or failures, while in the presence of agent mobility and proximity networks as in Definition 2.4, supplementary motion constraints should be introduced that constrain the pairwise distances between the agents and maintain all links created by the discrete topology controller  $\mathbb{S}$ . In particular, in the presence of communication time delays  $\tau_i > 0$ , a positive *dwelt time* is introduced between consecutive updates in the network topology that allows us to define continuous agent motion during these intervals that maintains the underlying network structure. We achieve this goal, as well as the dual interagent collision avoidance objective, using potential fields that *blow up* whenever the state of the system tends to violate any of these specifications.

As before, let  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  indicate a fixed topology of the network between any two consecutive switches and denote by  $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$  the set of neighbors of agent  $i$ . Define further the stack vectors  $\mathbf{x} \triangleq [\dots x_i^T \dots]^T \in \mathbb{R}^{pm}$  and  $\hat{\mathbf{x}} \triangleq [\dots x_{ij}^T \dots]^T \in \mathbb{R}^{pm(n-1)}$ , where  $x_{ij} \triangleq x_i - x_j$ , and for



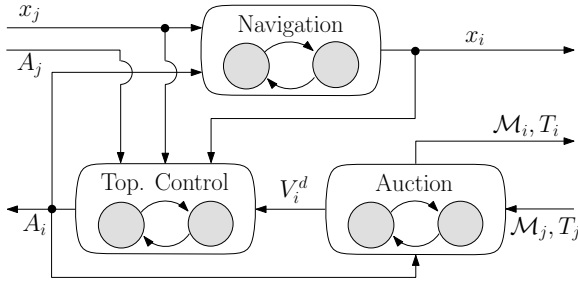


Fig. 10. Illustration of the hybrid automaton  $\mathbb{T}_i \times \mathbb{A}_i \times \mathbb{N}_i$  of a mobile agent  $i$  that consists of the composition of a topology control  $\mathbb{T}_i$ , an auction  $\mathbb{A}_i$  and a navigation automaton  $\mathbb{N}_i$ .

every agent  $i$  let  $\varphi_i \triangleq \sum_{j \in \mathcal{N}_i} \varphi_{ij}$ , where (Fig. 9)

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

Then, we have the following result.

**Theorem 6.1:** For all agents  $i$ , assume secondary objectives described by  $C^2$  potentials  $f_i : \mathbb{R}^p \rightarrow \mathbb{R}_+$  such that  $\lim_{\|x_i\|_2 \rightarrow \infty} f_i(x_i) = \infty$  (radially unbounded).<sup>18</sup> Then, the closed loop system defined by equation (2) and the control

$$u_i(t) \triangleq -K \nabla_{x_i} \varphi_i(t) \quad (9)$$

guarantees that all links in  $\mathcal{G}$  are maintained, collisions are avoided and all agent velocities are bounded.

*Proof:* See Appendix II. ■

Introducing mobile agents gives rise to the notion of a *navigation automaton*  $\mathbb{N}_i$  for every agent  $i$ , which coordinates with the associated topology control and auction automaton to obtain the agent's set of neighbors  $\mathcal{N}_i$ , which it uses, along with their positions  $x_j$  for  $j \in \mathcal{N}_i$ , to update its own position  $x_i$  (Eqns. 2 and 9). The updated agent positions are then provided to the topology control automaton that further updates agent  $i$ 's network estimate  $A_i$  and the resulting set of neighbors  $\mathcal{N}_i$ . Composition of all three automata  $\mathbb{T}_i$ ,  $\mathbb{A}_i$  and  $\mathbb{N}_i$  results in the hybrid model  $\mathbb{T}_i \times \mathbb{A}_i \times \mathbb{N}_i$  for a *mobile agent*  $i$ , shown in Fig. 10. Formally, the navigation automaton of agent  $i$  can be defined as follows (Fig. 11).

**Definition 6.2 (Navigation Automaton):** We define the navigation automaton of agent  $i$  by the tuple  $\mathbb{N}_i \triangleq (X_{\mathbb{N}_i}, V_{\mathbb{N}_i}, E_{\mathbb{N}_i}, inv, init, guard, reset, flow)$ , where,

- $X_{\mathbb{N}_i} \triangleq \{x_i, t_3^{[i]}\}$  denotes the set of owned state variables with  $x_i \in \mathbb{R}^p$  and  $t_3^{[i]} \in \mathbb{R}_+$ .
- $V_{\mathbb{N}_i} \triangleq \{N\}$  denotes the finite set of control modes.<sup>19</sup>

<sup>18</sup>We denote by  $\mathbb{R}_+$  the set  $[0, \infty)$ .

<sup>19</sup>The shorthand notation stands for  $N := \text{Navigate}$ .

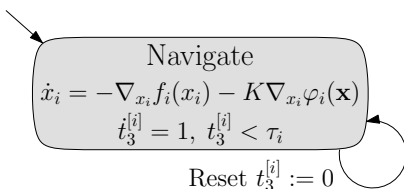


Fig. 11. Navigation automaton for agent  $i$  responsible for agent motion without violating the network structure.

- $E_{\mathbb{N}_i} \triangleq \{(N, N)\}$ , denotes the set of control switches.
- $inv : V_{\mathbb{N}_i} \rightarrow \text{Pred}(\cup_{j=1}^n (X_{\mathbb{T}_j} \cup X_{\mathbb{A}_j} \cup X_{\mathbb{N}_j}))$  with  $inv(N) \triangleq \{t_3^{[i]} < \tau_i\}$  denotes the invariant conditions of the hybrid automaton.
- $init : V_{\mathbb{N}_i} \rightarrow \text{Pred}(X_{\mathbb{N}_i})$  with  $init(N) \triangleq true$  denotes the set of initial conditions.
- $guard : E_{\mathbb{N}_i} \rightarrow \text{Pred}(\cup_{j=1}^n (X_{\mathbb{T}_j} \cup X_{\mathbb{A}_j} \cup X_{\mathbb{N}_j}))$  with  $guard((N, N)) \triangleq \{t_3^{[i]} = \tau_i\}$  denotes the set of transition guards.
- $reset : E_{\mathbb{N}_i} \rightarrow \text{Pred}(\cup_{j=1}^n (X_{\mathbb{T}_j} \cup X_{\mathbb{A}_j} \cup X_{\mathbb{N}_j}))$  with  $reset((N, N)) \triangleq \{t_3^{[i]} := 0\}$  denotes the set of transition resets.
- $flow : V_{\mathbb{N}_i} \rightarrow \text{Pred}(\dot{X}_{\mathbb{N}_i} \cup \cup_{j=1}^n (X_{\mathbb{T}_j} \cup X_{\mathbb{A}_j} \cup X_{\mathbb{N}_j}))$  with  $flow(N) \triangleq \{\dot{x}_i = -\nabla_{x_i} f_i(x_i) - K \nabla_{x_i} \varphi_i(x), t_3^{[i]} = 1\}$ , denotes the flow conditions of the hybrid automaton that constrain the first time derivatives of the system variables in mode  $v \in V_{\mathbb{N}_i}$ .

Observe that the navigation automaton  $\mathbb{N}_i$  consists of a single mode including agent  $i$ 's dynamics as in Theorem 6.1, while self-transitions in  $\mathbb{N}_i$  are associated with updates in the agent's set of neighbors  $\mathcal{N}_i$  and are message triggered and synchronized with transitions of  $\mathbb{T}_i$  and  $\mathbb{A}_i$ , as before.<sup>20</sup> Correctness of the resulting mobile multiagent system  $\mathbb{S} \triangleq \times_{i=1}^n (\mathbb{T}_i \times \mathbb{A}_i \times \mathbb{N}_i)$  follows directly from Theorems 5.9 and 6.1. In particular, we have the result.

**Theorem 6.3:** Assuming that  $\mathcal{G}(t_0)$  is initially connected, the product system  $\mathbb{S}$  guarantees connectivity of the mobile network  $\mathcal{G}(t)$  for all time  $t \geq t_0$ .

## VII. CONNECTIVITY TASKS

In this section we illustrate the proposed distributed topology control algorithm in nontrivial connectivity tasks involving both stationary and mobile agents and show that it has the desired connectivity maintenance, collision avoidance and scalability properties. In particular, we first consider  $n = 50$  stationary agents (dots), randomly distributed in a square of unit area, such that the distance between any two of them is greater than  $r = .05$ , and define a link (line) between any two such agents if their pairwise distance is less than  $R = .2$  (Fig. 12(a)). According to Definition 2.4, this implies that no new links can be added in the network, while all existing links are candidates for deletion, i.e.,  $\mathcal{N}_i^d(t) = \mathcal{N}_i(t)$  for all agents  $i$ , as in Section III-B. Observe that the distributed topology control algorithm succeeds in deleting links while maintaining connectivity of the network, captured by a positive second smallest eigenvalue of the graph laplacian matrix  $\lambda_2 > 0$  (Fig. 12(d)). The algorithm terminates when the final network corresponds to a minimally connected *tree* structure (Fig. 12(b)), while the number of links in the network as a function of time is illustrated in Fig. 12(c).

In our next connectivity scenario, we consider  $n = 12$  *mobile* agents in  $\mathbb{R}^3$  and compare  $k$ -connectivity control

<sup>20</sup>To define the continuous motion dynamics, we make the simplifying assumption that the neighbor positions  $x_j$  for  $j \in \mathcal{N}_i$  are transmitted in much higher frequencies than the messages  $\text{Msg}[j]$  so that they can be approximated by a continuous signal. In practice, this assumption can be relaxed by discrediting the motion dynamics. Details can be found in [31], where the proposed algorithm is implemented on a real robotic platform.

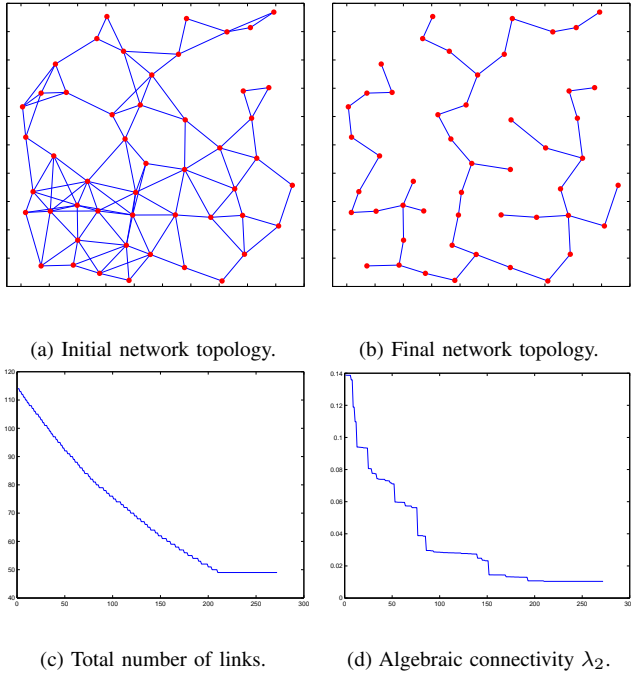


Fig. 12. Dynamic network consisting of 50 stationary agents. Distributed topology control guarantees link deletions while maintaining connectivity. The final network topology corresponds to a minimally connected *tree* structure.

for  $k = 1, 2$  and for the same initial configuration of the agents. We classify the agents into a set of so called *leaders*  $\mathcal{L} = \{1, 2\}$  labeled by the letter “L” and having a nontrivial secondary objective and a set of *followers*  $\{1, \dots, n\} \setminus \mathcal{L}$  having no secondary objective. In particular, for all leaders  $i \in \mathcal{L}$  we assume secondary objectives as in Fig. 1(b), with an additional unit angular velocity term (slightly abusing the secondary objective specifications in equation (2)), designed to *stretch* the network and observe whether it can reconfigure while maintaining connectivity. Interagent links that are within  $r = .25$  are denoted by solid lines, while candidate links for deletion, i.e., links that are within  $R = .4$ , are denoted by dotted lines (Definition 2.4). Solid curves attached to every agent indicate the recently traveled paths and give an idea of the agents’ motion. Figs. 13 compare the evolution of the system at four consecutive time instants for  $k = 1, 2$ . Note that, under the proposed connectivity control laws, the overall network remains  $k$ -connected, while the leaders *do their best* to achieve their secondary objectives.

### VIII. 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. For this, we proposed a distributed feedback and provably correct control framework that, unlike most prior work, imposed no restrictions on the network topology other than the desired connectivity specification. Our approach was based on a key control decomposition, where connectivity control of the network structure was performed in the discrete space of graphs and relied on local estimates of the network topology, algebraic graph theory and market-based control, while motion control

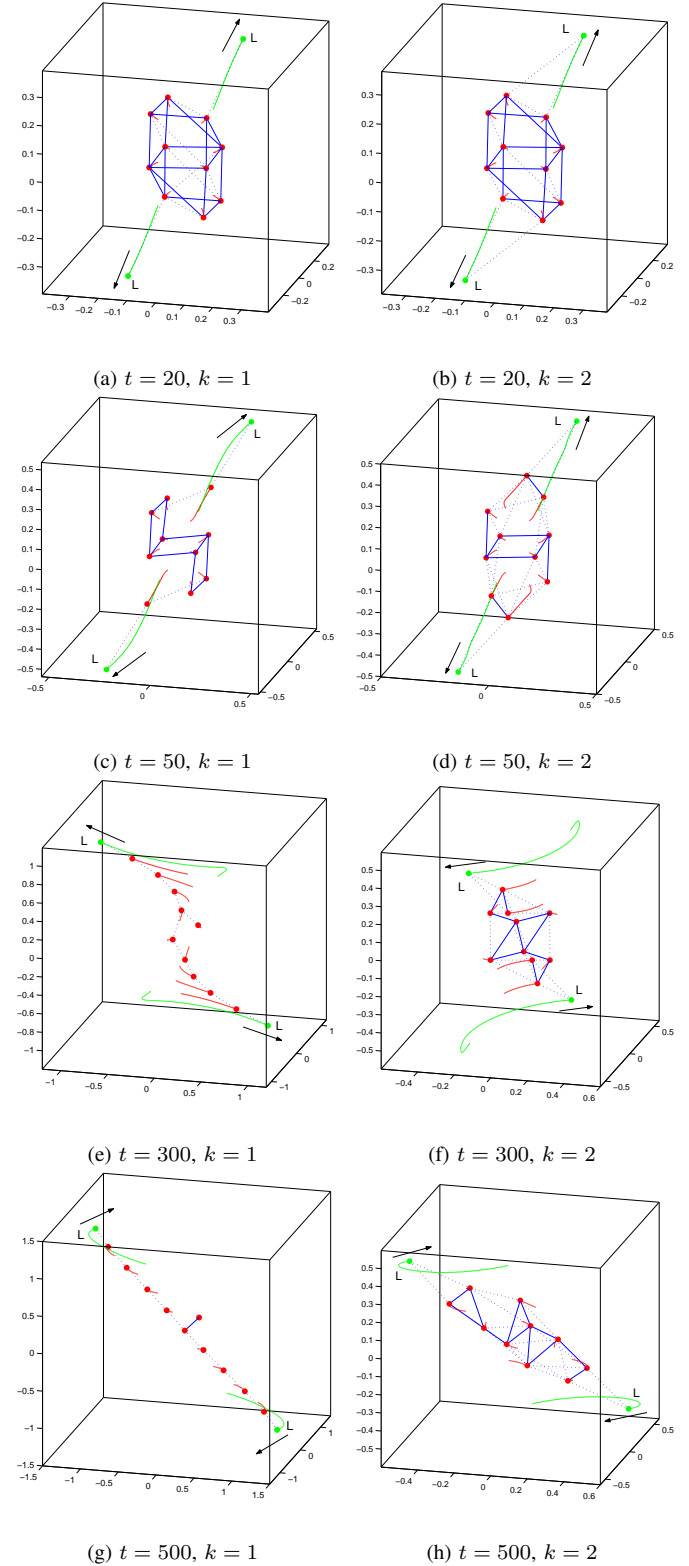


Fig. 13.  $k$ -Connectivity control for  $n = 12$  agents with 2 leaders. Compare figures (a,b), (c,d), (e,f) and (g,h).

TABLE III  
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

of the agents was performed in the continuous configuration space by means of local potential fields used to maintain nearest neighbor links. Integration of the above controllers resulted in a hybrid model for every agent, which was shown to always guarantee connectivity of the network, while it reconfigures towards certain secondary objectives. Communication time delays in the network as well as collision avoidance among adjacent agents were also efficiently handled, while our approach was illustrated through a class of interesting problems that could be achieved while preserving connectivity.

## APPENDIX I

### A. Boolean Operations

*Definition 1.1 (Boolean Operations on Scalars):* 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 III, where the symbols  $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\rightarrow$  and  $\leftrightarrow$  stand for *not*, *and*, *or*, *if, then* and *if and only if*, respectively.

Similarly, we can define boolean operations on boolean matrices  $X, Y \in \{0, 1\}^{n \times n}$ .

*Definition 1.2 (Boolean Operations on 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  $X \wedge Y \triangleq (x_{ij} \wedge y_{ij})$  and in a similar way we can define any other boolean operation on matrices.

## APPENDIX II

### A. Proof of Theorem 5.9

Since all changes in the topology of  $\mathcal{G}(t)$  are due to  $\mathbb{S}$  (no random link additions or failures), we only need to show that connectivity is maintained at the transition time instants of  $\mathbb{S}$ , when the network structure is updated. However, since addition of links does not endanger network connectivity, we only need to show that connectivity is not violated when a link is deleted from  $\mathcal{G}(t)$ .

Note first that Propositions 5.4 and 5.5 imply that the outcome of every auction is common for all agents  $i$  and is at most one link that is eventually deleted from all edge sets  $\mathcal{E}_i(t)$ . Hence, we need to show that the selection of this link is safe with respect to connectivity. For this, consider any agent  $i$  such that  $v_{\mathbb{A}_i}^s = S_{z_k}$  for  $t = t_s$ . Then, at  $t = t_{s+1}$  the transition  $(v_{\mathbb{A}_i}^s, v_{\mathbb{A}_i}^{s+1})$  results in  $v_{\mathbb{A}_i}^{s+1} = B_{z_k}$  and the corresponding reset initializes a set of candidate neighbors  $\mathcal{S}_i(t_{s+1})$  with which agent  $i$  can safely delete a link. Since the network estimates  $\mathcal{G}_i(t)$  are updated with information from the

same auction and  $\mathcal{G}_i(t_s)$  is a spanning subgraph of  $\mathcal{G}(t_s)$ , i.e.,  $\mathcal{E}_i(t_s) \subseteq \mathcal{E}(t_s)$  by Proposition 5.8, any deletion of a link  $(i, j)$  with  $j \in \mathcal{S}_i(t_{s+1}) \neq \emptyset$  does not violate connectivity of the local network estimate  $\mathcal{G}_i(t_{s+1}) = (\mathcal{V}, \mathcal{E}_i(t_{s+1}))$  and, hence, neither does it violate connectivity of the overall network  $\mathcal{G}(t_{s+1}) = (\mathcal{V}, \mathcal{E}(t_{s+1}))$ .

### B. Proof of Theorem 6.3

Consider the potential function  $\varphi_{\mathcal{G}} : D_{\mathcal{G}} \times \mathbb{R}^{pn} \rightarrow \mathbb{R}_+$  such that

$$\varphi_{\mathcal{G}} \triangleq \sum_{i=1}^n f_i + \frac{K}{2} \sum_{i=1}^n \varphi_i, \quad (10)$$

where  $D_{\mathcal{G}} \triangleq \{\hat{\mathbf{x}} \in \mathbb{R}^{pn(n-1)} \mid \|x_{ij}\|_2 \in (0, R), \forall (i, j) \in \mathcal{E}\}$  and for any  $c > 0$ , define the set  $\Omega_{\mathcal{G}} \triangleq \{(\hat{\mathbf{x}}, \mathbf{x}) \in D_{\mathcal{G}} \times \mathbb{R}^{pn} \mid \varphi_{\mathcal{G}} \leq c\}$ . Observe, further, that

$$\Omega_{\mathcal{G}} \subseteq \left( \bigcap_{i=1}^n f_i^{-1}([0, c]) \right) \cap \left( \bigcap_{(i,j) \in \mathcal{E}} \varphi_{ij}^{-1}([0, c]) \right) \triangleq \Omega.$$

The sets  $f_i^{-1}([0, c])$  are closed by continuity of the potentials  $f_i$  in  $\mathbb{R}^p$ . They are also bounded; to see this, suppose that there exists an  $i$  for which  $f_i^{-1}([0, c])$  is unbounded. Then, for any choice of  $N > 0$ , there exists an  $x_i \in f_i^{-1}([0, c])$  such that  $\|x_i\|_2 > N$ . Allowing  $N \rightarrow \infty$  and given that  $\lim_{\|x_i\|_2 \rightarrow \infty} f_i(x_i) = \infty$ , it follows that for any  $M > 0$ , there is an  $N > 0$  such that  $f_i(x_i) > M$ . If we pick  $M > c$  we reach a contradiction, since by definition  $x_i \in f_i^{-1}([0, c]) = \{x_i \mid f_i(x_i) \leq c\}$ . Thus, all sets  $f_i^{-1}([0, c])$  are bounded and hence, compact. Similarly, for all  $(i, j) \in \mathcal{E}$  the sets  $V_{ij}^{-1}([0, c])$  are closed by continuity of  $V_{ij}$  in the interval  $(0, R)$ . They are also bounded; to see this, suppose there exist indices  $i$  and  $j$  for which  $V_{ij}^{-1}([0, c])$  is unbounded. Then, for any choice of  $N \in (0, R)$ , there exists an  $x_{ij} \in V_{ij}^{-1}([0, c])$  such that  $\|x_{ij}\|_2 > N$ . Allowing  $N \rightarrow R$ , and given that  $\lim_{\|x_{ij}\|_2 \rightarrow R} V_{ij} = \infty$ , it follows that for any  $M > 0$ , there is a  $N > 0$  such that  $V_{ij} > M$ . If we pick  $M > c$  we reach a contradiction, since by definition  $x_{ij} \in V_{ij}^{-1}([0, c]) = \{x_{ij} \mid V_{ij}(x_{ij}) \leq c\}$ . Thus, all sets  $\varphi_{ij}^{-1}([0, c])$  are bounded and hence, compact. Therefore, the set  $\Omega$  is compact as a finite intersection of compact sets. It follows that  $\Omega_{\mathcal{G}}$  is also compact, as a closed subset of a compact set.

The time derivative of  $\varphi_{\mathcal{G}}$  in the set  $\Omega_{\mathcal{G}}$  becomes  $\dot{\varphi}_{\mathcal{G}} = \sum_{i=1}^n \dot{f}_i + \frac{K}{2} \sum_{i=1}^n \dot{\varphi}_i$ , where

$$\begin{aligned} \sum_{i=1}^n \dot{V}_i &= \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} \dot{x}_{ij}^T \nabla_{x_{ij}} \varphi_{ij} \\ &= \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} \left( \dot{x}_i^T \nabla_{x_i} \varphi_{ij} + \dot{x}_j^T \nabla_{x_j} \varphi_{ij} \right) \\ &= 2 \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} \dot{x}_i^T \nabla_{x_i} \varphi_{ij} = 2 \sum_{i=1}^n \dot{x}_i^T \nabla_{x_i} \varphi_i \end{aligned}$$

by symmetry of the functions  $\varphi_{ij}$ . Thus,

$$\begin{aligned} \dot{\varphi}_{\mathcal{G}} &= \sum_{i=1}^n \dot{x}_i^T \nabla_{x_i} f_i + K \sum_{i=1}^n \dot{x}_i^T \nabla_{x_i} \varphi_i \\ &= - \sum_{i=1}^n \|\nabla_{x_i} f_i + K \nabla_{x_i} \varphi_i\|_2^2 \leq 0, \end{aligned}$$

which implies that the level sets  $\Omega_G$  of  $\varphi_G$  are also positively invariant. The invariance of  $\Omega_G$  implies that all links in the network are maintained and that no collisions between agents occur. On the other hand, compactness and positive invariance of  $\Omega_G$  also implies that  $(\hat{\mathbf{x}}, \mathbf{x}) \in D_G \times \mathbb{R}^{pn}$  remains bounded for all time  $t$  between any two consecutive switches in  $\mathcal{G}$ . Moreover, since  $\varphi_G \in C^2$  inside  $D_G \times \mathbb{R}^{pn}$ , the right-hand-side of the closed loop system defined in equations (2) and (9) is locally Lipschitz, which implies that  $\dot{\mathbf{x}}$  is bounded. Hence, all agent velocities are bounded. Note, that the *hysteresis* in Definition 2.4 ensures that if a link  $(i, j) \notin \mathcal{E}$  is added to  $\mathcal{E}$ , then the associated potential  $\varphi_{ij}$  is bounded and hence, so is the new potential  $\varphi_G$ . This observation allows us to define level sets of the potentials  $\varphi_G$ .

## REFERENCES

- [1] 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.
- [2] 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.
- [3] W. Ren and R. Beard. *Consensus of Information under Dynamically changing Interaction Topologies*, Proceedings of the American Control Conference, Boston, MA, June 2004, pp. 4939–4944.
- [4] 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.
- [5] 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, Maui, Hawaii, Dec. 2003, pp. 1508–1513.
- [6] H. G. Tanner, A. Jadbabaie and G. J. Pappas. *Flocking in Fixed and Switching Networks*, IEEE Transactions on Automatic Control, vol. 52(5), pp. 863–868, May 2007.
- [7] 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.
- [8] M. M. Zavlanos and G. J. Pappas. *Dynamic Assignment in Distributed Motion Planning with Local Coordination*, IEEE Transactions on Robotics, vol. 24(1), pp. 232–242, Feb. 2008.
- [9] 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, Dec. 2001.
- [10] 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.
- [11] 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.
- [12] P. Ogren, M. Egerstedt 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.
- [13] K. E. Bekris, A. A. Argyros and L. E. Kavraki. *New Methods for Reaching the Entire Plane with Angle-Based Navigation*, Proc. of The IEEE International Conference on Robotics and Automation, New Orleans, LA, April 2004, pp. 2373–2378.
- [14] S. Poduri and G. S. Sukhatme. *Constrained Coverage for Mobile Sensor Networks*, in Proc. IEEE Int. Conf. Robot. Autom., New Orleans, LA, May 2004, pp. 165–172.
- [15] M. Mesbahi. *On State-dependent Dynamic Graphs and their Controllability Properties*, IEEE Transactions on Automatic Control, vol. 50(3), pp. 387–392, March 2005.
- [16] D. P. Spanos and R. M. Murray. *Robust Connectivity of Networked Vehicles*, Proceedings of the 43rd IEEE Conference on Decision and Control, Bahamas, Dec. 2004, pp. 2893–2898.
- [17] M. Ji and M. Egerstedt. *Distributed Formation Control while Preserving Connectedness*, Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, Dec. 2006, pp. 5962–5967.
- [18] 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.
- [19] M. C. DeGennaro and A. Jadbabaie. *Decentralized Control of Connectivity for Multi-Agent Systems*, Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, Dec. 2006, pp. 3628–3633.
- [20] G. Notarstefano, K. Savla, F. Bullo and A. Jadbabaie. *Maintaining Limited-Range Connectivity among Second-Order Agents*, American Control Conference, Minneapolis, MN, June 2006, pp. 2124–2129.
- [21] 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.
- [22] R. D’Souza, D. Galvin, C. Moore and D. Randall. *Global Connectivity from Local Geometric Constraints for Sensor Networks with Various Wireless Footprints*, Proceedings of the 5th International Conference on Information Processing in Sensor Networks, Nashville, TN, April 2006, pp. 19–26.
- [23] 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, Santa Fe, NM, April 2004, pp. 216–223.
- [24] M. M. Zavlanos, A. Tahbaz-Salehi, A. Jadbabaie and G. J. Pappas. *Distributed Topology Control of Dynamic Networks*, American Control Conference, Seattle, WA, June 2008, pp. 2660–2665.
- [25] 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, August 2007.
- [26] M. M. Zavlanos and G. J. Pappas. *Controlling Connectivity of Dynamic Graphs*, Proc. of the 44th IEEE Conference on Decision and Control, Seville, Spain, Dec. 2005, pp. 6388–6393.
- [27] T. A. Henzinger. *The Theory of Hybrid Automata*, Proc. of the 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, NJ, July 1996, pp. 278–292.
- [28] C. Godsil and G. Royle. *Algebraic Graph Theory*, Springer-Verlag, New York, 2001.
- [29] G. H. Golub and C. F. Van Loan. *Matrix Computations*, Johns Hopkins University Press, 1996.
- [30] Shing-Tung Yau and Ya Yan Lu. *A New Approach to Sparse Matrix Eigenvalues*, Proc. of the 1st IEEE Conference on Regional Aerospace Control Systems, Westlake Village, CA, May 1993, pp. 132–137.
- [31] N. Michael, M. M. Zavlanos, V. Kumar and G. J. Pappas. *Maintaining Connectivity in Mobile Robot Networks*, Proc. of the 11th International Symposium on Experimental Robotics, Athens, Greece, July 2008.
- [32] M. M. Zavlanos and G. J. Pappas. *Distributed Connectivity Control of Mobile Networks*, Proc. of the 46th IEEE Conference on Decision and Control, New Orleans, LA, Dec. 2007, pp. 3591–3596.
- [33] M. M. Zavlanos and G. J. Pappas. *Distributed Connectivity Control of Mobile Networks*, Technical Report MS-CIS-07-08, Dept. of Computer and Information Science, University of Pennsylvania, March 2007. (<http://www.seas.upenn.edu/~zavlanos/>)