Distributed Connectivity Control of Mobile Networks

Michael M. Zavlanos and George J. Pappas

Technical Report MS-CIS-07-08

Department of Computer and Information Science, University of Pennsylvania

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 is illustrated in nontrivial computer simulations.

Index Terms

Mobile Networks, Graph Connectivity, Time Delays, Boolean Algebra, Hybrid Systems.

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

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

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.

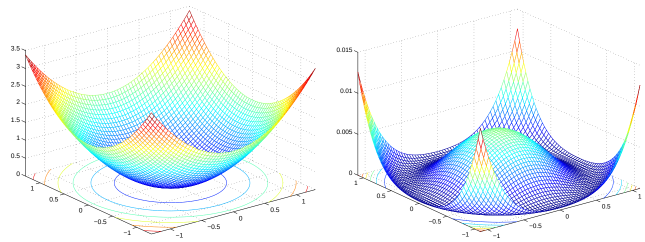
Motivated by the significance of connectivity in many applications involving multi-agent systems and mobile sensor networks [9] $-$ [19], in this paper, we consider graph connectivity as our primary objective. Unlike centralized [4], [20], [21], or distributed open loop [5], [6] approaches to the problem, 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 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. Integration of the above controllers is possible due to a novel representation of the network topology by means of proximity graphs, modified to include a hysteresis in link additions, and results in a hybrid model for each agent [23]. 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 our approach is finally 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 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. Integration of the resulting controllers that consist the hybrid models for the agents is discussed in Section IV, while the properties of the overall system are studied in Section V. Finally, in Section VI, we state and verify through computer simulations, nontrivial connectivity tasks that best illustrate our approach.

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,

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



(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).

where $f_i \geq 0$ corresponds to a continuous radially unbounded potential indicating a *secondary objective* (Figure 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.[1] Controlling links in $\mathcal{G}(t)$ is then decomposed into addition and deletion of links in the open ball and annulus, respectively (Figure 2). 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 crucial 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 is of particular interest to us, since it consists 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 $\mathcal{G}(t)$.

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

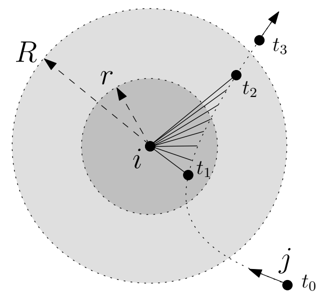


Fig. 2. Link (solid lines) dynamics according to Definition 2.1. Note that the *candidate* link $i \sim j$ is deleted at time t_2.

Hence, the problem addressed in this paper can be stated as follows.

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 of the problem, 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 $\Delta(t) = \text{diag}\big(\sum_{j=1}^{n} a_{ij}(t)\big)$ 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 ([22]): 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.

Remark 2.4 (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 [22]. 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 $\lambda_2(L(\mathcal{G})) \leq \kappa(\mathcal{G}) \leq \eta(\mathcal{G})$ [22]. 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 2.2.

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

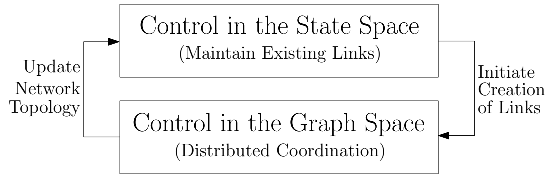


Fig. 3. Control Decomposition.

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 (Figure 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 the hysteresis in creation of new links, enable 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 with respect to 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 define the potential $\varphi_i(\mathbf{x}) = \sum_{j \in \mathcal{N}_i} \varphi_{ij}(x_{ij})$ where (Figure 4),

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

Then, we have the following result.

Theorem 3.1: 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).[3] Then, under control inputs,

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

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

Proof: Consider the potential function $\varphi_{\mathcal{G}} : D_{\mathcal{G}} \to \mathbb{R}_+$ such that,

$$\varphi_{\mathcal{G}}(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i) + \frac{K}{2} \sum_{i=1}^{n} \varphi_i(\mathbf{x})$$

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

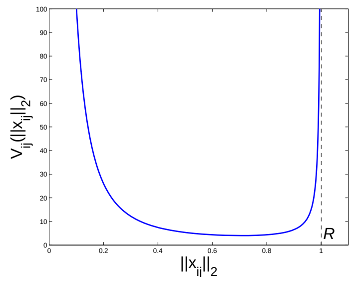


Fig. 4. Plot of the potential $\varphi_{ij}(x_{ij})$ for $R = 1$.

where $D_{\mathcal{G}} = \{\mathbf{x} \in \mathbb{R}^{pn} \mid \|x_{ij}\|_2 \in (0, R), \; \forall \; (i,j) \in \mathcal{E}\}$ and for any $c > 0$, define the set $\Omega_{\mathcal{G}} := \{\mathbf{x} \in D_{\mathcal{G}} \mid \varphi_{\mathcal{G}}(\mathbf{x}) \leq c\}$. Note that $\Omega_{\mathcal{G}}$ is closed by continuity of $\varphi_{\mathcal{G}}$ in its domain $D_{\mathcal{G}}$.[4] Moreover,

$$\Omega_{\mathcal{G}} \subseteq \left(\cap_{i=1}^n \{x_i \mid f_i(x_i) \leq c\} \right) \cap \left(\cap_{(i,j)\in\mathcal{E}} \{x_i, x_j \mid \varphi_{ij}(x_{ij}) \leq c\} \right) \subseteq \cap_{i=1}^n f_i^{-1}([0,c]) = \Omega$$

By continuity of the potentials f_i, all sets $f_i^{-1}([0,c])$ are closed, and so Ω is closed as an intersection of closed sets. Moreover, $f_i^{-1}([0,c])$ are also bounded for all i. To see this, suppose that there exists an i such that $f_i^{-1}([0,c])$ is unbounded. Then, for every $N > 0$, there exists an $x_i \in f_i^{-1}([0,c])$ such that $\|x_i\|_2 > N$. On the other hand, since $\lim_{\|x_i\|_2 \to \infty} f_i(x_i) = \infty$, for all $M > 0$ there exists an $N > 0$ such that $\|x_i\|_2 > N$ implies $f_i(x_i) > M$. Let $M > c > 0$. Then, there exists an $N > 0$ and $\tilde{x}_i \in f_i^{-1}([0,c])$ with $\|\tilde{x}_i\|_2 > N$ such that $f_i(\tilde{x}_i) > M > c$, by unboundedness of $f_i^{-1}([0,c])$. But, $\tilde{x}_i \in f_i^{-1}([0,c])$ implies $f_i(\tilde{x}_i) \in [0,c]$, which results in a contradiction. Hence, all sets $f_i^{-1}([0,c])$ are bounded, which implies that Ω is also bounded. We conclude that Ω is compact, since it is both closed and bounded, and so $\Omega_{\mathcal{G}}$ is compact too, as a closed subset of the compact set Ω. Now, the time derivative of $\varphi_{\mathcal{G}}$ in the set $\Omega_{\mathcal{G}}$ becomes,

$$\dot{\varphi}_{\mathcal{G}}(\mathbf{x}) = \sum_{i=1}^n \dot{f}_i(x_i) + \frac{K}{2} \sum_{i=1}^n \dot{\varphi}_i(\mathbf{x})$$

where,

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

by symmetry of the functions φ_{ij}. Hence,

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

[4]Since $\varphi_{\mathcal{G}}$ is continuous in $D_{\mathcal{G}}$, the preimage $\varphi_{\mathcal{G}}^{-1}([0,c])$ of the closed set $[0,c]$ is closed.

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

which implies that $\Omega_{\mathcal{G}}$ is positively invariant. Hence, all inter-agent potentials φ_{ij} with $(i,j) \in \mathcal{E}$ remain bounded.[5] But, $\lim_{\|x_{ij}\|_2 \to 0} \varphi_{ij}(x_{ij}) = \infty$ and $\lim_{\|x_{ij}\|_2 \to R} \varphi_{ij}(x_{ij}) = \infty$ for any $(i,j) \in \mathcal{E}$. Thus, for all time t, we have $\|x_{ij}\|_2 \in (0, R)$ for all $(i,j) \in \mathcal{E}$. In other words, all links in $\mathcal{G}$ are maintained and collisions among adjacent agents are avoided. Note, finally, that the *hysteresis* in Definition 2.1 ensures that if a link $(i,j) \notin \mathcal{E}$ is added to $\mathcal{E}$, then the associated potential φ_{ij} is bounded and hence, so is the new potential $\varphi_{\mathcal{G}}$. This observation allows us to define level sets of the potentials $\varphi_{\mathcal{G}}$. ∎

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 braking whenever simultaneous link deletions may violate network connectivity.

1) Locally Updated Spanning Subgraphs: Let $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ denote the dynamic graph corresponding to the mobile network at time $t \geq t_0$. 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 a link is deleted from a graph that 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.[6] 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),[7]

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

[5]By continuity, the potentials f_i are bounded in any compact subset of $\mathbb{R}^p$, hence they are bounded in $\Omega_{\mathcal{G}}$.

[6]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 $t \geq t_0$, as it will be shown in Section V.

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

TABLE II

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

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 $j \sim k$.[8] In matrix form, the dynamics in equation (4) become,

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

where the control input $V_i(t) = \big(v_{jk}^{[i]}(t)\big)$ 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 (Table II),

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

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 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,

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

where, in general, $V_i^c(t) \neq V_i^d(t)$. 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 they are not provided by adjacent agents. On the other hand, if $\big((\neg A_i(t)) \wedge A_i^{(1)}(t)\big) \to V_i^c(t)$, then $A_i(t)$ is updated with all existing links in the network that agent i is not aware of. The following proposition shows that if the estimates $\mathcal{G}_i(t)$ of all agents are exclusively initialized with nearest neighbor links and the overall network $\mathcal{G}(t)$ is fixed and connected, then the agents will achieve *consensus* on their estimates $A_i(t)$

[8]The discrete time semantics in (4) will be made clear in Section IV.

in finite time. Hence, the proposed local dynamics (5) allow every agent to have a *rough* picture of the overall network, as desired.

Proposition 3.2: Assume a fixed graph $\mathcal{G}(t_0)$ with corresponding adjacency matrix $A(t_0)$ and initialize the estimates $A_i(t_0)$ of all agents such that $A_i(t_0) = E_i \wedge A(t_0)$. Let $V_i^c(t) = \big((\neg A_i(t)) \wedge A_i^{(1)}(t)\big)$ and $V_i^d(t) = \mathbf{0}$ for all time $t \geq t_0$. Then, $A_i(t_0 + n - 1) = A(t_0)$ for all agents i.

Proof: Observe that,

$$F_i(t) \wedge V_i^c(t) \leftrightarrow F_i(t) \wedge \big((\neg A_i(t)) \wedge A_i^{(1)}(t)\big) \leftrightarrow \big((\neg A_i(t)) \wedge A_i^{(1)}(t)\big)$$

and the dynamical system (5) becomes,[9]

$$A_i(t+1) = \neg\big(A_i(t) \leftrightarrow \big((\neg A_i(t)) \wedge A_i^{(1)}(t)\big)\big) \leftrightarrow \big(A_i^{(1)}(t) \vee A_i(t)\big) \tag{8}$$

Define, further, existing links in the network provided by the k-hop neighbors of agent i,

$$A_i^{(k)}(t) := \vee_{j_1 \in \mathcal{N}_i \ldots j_k \in \mathcal{N}_{j_{k-1}}} A_{j_k}(t)$$

and for any $1 \leq l \leq k$ observe that,

$$A_i^{(k)}(t) \leftrightarrow \vee_{j_1 \in \mathcal{N}_i \ldots j_l \in \mathcal{N}_{j_{l-1}}} A_{j_l}^{(k-l)}(t)$$

where by convention $j_0 = i$. Hence, using the dynamical system (8), we get the recursion,

$$A_i^{(k)}(t+1) \leftrightarrow \vee_{j_1 \in \mathcal{N}_i \ldots j_k \in \mathcal{N}_{j_{k-1}}} A_{j_k}(t+1) \leftrightarrow \vee_{j_1 \in \mathcal{N}_i \ldots j_k \in \mathcal{N}_{j_{k-1}}} \big(A_{j_k}(t) \vee A_{j_k}^{(1)}(t)\big) \leftrightarrow A_i^{(k)}(t) \vee A_i^{(k+1)}(t)$$

and so,

$$A_i(t) \leftrightarrow A_i(t-1) \vee A_i^{(1)}(t-1) \leftrightarrow A_i(t-2) \vee A_i^{(1)}(t-2) \vee A_i^{(2)}(t-2) \leftrightarrow \vee_{j=0}^{k} A_i^{(j)}(t-k)$$

which models the information propagation in the network. Clearly, if the graph is connected, then,

$$A_i(t_0 + n - 1) \leftrightarrow \vee_{j=0}^{n-1} A_i^{(j)}(t_0) \leftrightarrow \vee_{j=1}^{n} A_j(t_0) \leftrightarrow A(t_0)$$

which completes the proof. ■

2) Market-Based Coordination: Having defined the network estimates $\mathcal{G}_i(t)$ that allow every agent to have a rough picture of the overall network and their local dynamics (5), the main challenge is to determine control inputs $V_i^c(t)$ and $V_i^d(t)$ for all agents i that ensure connectivity of each subgraph $\mathcal{G}_i(t)$ for all time t. Then, in Section V, we show that 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,

$$v_{jk}^{[i]c}(t) := \neg\big((j = i) \vee (k = i)\big) \vee \big(x_k(t) \in \mathcal{B}_r(x_j(t))\big) \tag{9}$$

[9]Note that equation (8) can be rewritten as $A_i(t+1) := \vee_{j \in \mathcal{N}_i}\big(A_i(t) \vee A_j(t)\big)$, which is exactly a consensus equation on the estimates $A_i(t)$.

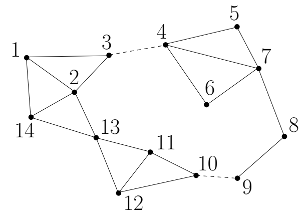


Fig. 5. Deleting any of the links (3,4) or (9,10) individually does not violate connectivity, but deleting them both results in a disconnected graph. None of the involved agents are adjacent.

where $r > 0$ is as in Definition 2.1 and $(j = i) \vee (k = i)$ guarantees the local nature of the controller (note that $v_{ij}^{[i]c}(t_1) = 1$ in Figure 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 (Figure 5). 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 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, where $r > 0$ is according to Definition 2.1 and,

$$\mathcal{B}_r(x_i(t)) = \{\ y(t) \in \mathbb{R}^p \mid \|y(t) - x_i(t)\|_2 < r\ \} \tag{10}$$

denotes an open ball of radius $r > 0$ centered at $x_i(t) \in \mathbb{R}^p$. 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 Figure 2),

$$v_{jk}^{[i]d}(t) := (w_i(t) = j \sim k) \wedge (|w_i(t)| = 1) \tag{11}$$

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 $\mathcal{E}_i(t)$. Clearly, the existence of some notion of *synchronization* of all agents to the same auction for all time is necessary for correctness of the above market-based

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., $\mathcal{S}_i(t) := \{j \in \mathcal{N}_i(t) \backslash \mathcal{N}_i^r(t) \mid \lambda_2(L(\mathcal{E}_i(t) \backslash \{i \sim j\})) > k-1\}$, according to Remark 2.4.

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., $T_i(t+1) := T_i(t) \vee \left(\vee_{j \in \mathcal{N}_i(t)} T_j(t) \right)$

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

end while

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

coordination scheme. This requirement becomes even more important in the presence of communication time delays in the system (Section IV).

Remark 3.3: 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 3.4: 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.[10] 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.

IV. Modeling the Agents in the Presence of Time Delays

Integration of the continuous time control laws in Subsection III-A with the discrete coordination protocols in Subsection III-B gives rise to a hybrid model for every agent [23] that consists of a *navigation* and a *coordination* automaton, as shown in Figure 6. *Input* to the navigation automaton of agent i consists the neighbor set $\mathcal{N}_i(t)$, while *output* is its updated state $x_i(t)$. On the other hand, *inputs* to the coordination automaton of agent i consist the states $x_j(t)$, network estimates $A_j(t)$, max-bid sets $\mathcal{M}_j(t)$ and token vectors $T_j(t)$ of all agents, while *outputs* are

[10]For a complete graph 1 step is sufficient for convergence of the max-consensus algorithm.

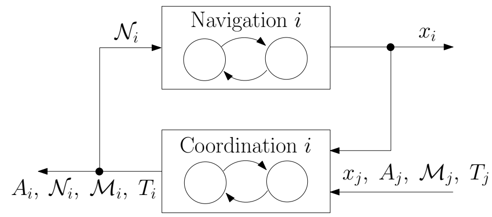


Fig. 6. Model for Agent i.

its own updated neighbor set $\mathcal{N}_i(t)$, network estimate $A_i(t)$, max-bid set $\mathcal{M}_i(t)$ and token vector $T_i(t)$.[11] Updates in both navigation and coordination automata depend explicitly on nearest neighbors $\mathcal{N}_i(t)$, hence the distributed nature of the approach. The two automata are synchronized and together consist the model of agent i. 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 $Pred(X)$.

In other words, a predicate is a logical formula. For example, the predicate $\psi(X) = \big(\|x - x_0\|_2 < r\big)$ over the set of variables $X \in \mathbb{R}^N$ returns 1 if x belongs in the open ball $\|x - x_0\|_2 < r$ and 0 otherwise. Hence, the navigation automaton for agent i can be defined as follows.[12]

Definition 4.2 (Navigation Hybrid Automaton): We define the navigation hybrid automaton of agent i to be the tuple $N_i = (X_{N_i}, V_{N_i}, E_{N_i}, \Sigma_{N_i}, sync, inv, init, guard,\ reset, flow)$, where,

- $X_{N_i} = \{x_i\}$ denotes the set of owned state variables with $x_i \in \mathbb{R}^p$.
- $V_{N_i} = \{N\}$ denotes the finite set of control modes.[13]
- $E_{N_i} = \{(N, N)\}$, denotes the set of control switches.
- $\Sigma_{N_i} = \{update_i\}$ denotes the set of synchronization labels.
- $sync : E_{N_i} \to \Sigma_{N_i}$ with, $sync((N, N)) = update_i$, denotes the synchronization map mapping each control switch to a synchronization label.
- $inv : V_{N_i} \to Pred(X_{N_i})$ with $inv(N) = true$, denotes the invariant conditions of the hybrid automaton.
- $init : V_{N_i} \to Pred(X_{N_i})$ with $init(N) = true$, denotes the set of initial conditions.
- $guard : E_{N_i} \to Pred(X_{N_i})$ with $guard((N, N)) = true$, denotes the set of guards of the hybrid automaton.
- $reset : E_{N_i} \to X_{N_i}$ with $x_i := reset((N, N)) = x_i$, denotes the set of resets associated with the guards of the hybrid automaton.
- $flow : V_{N_i} \to \dot{X}_{N_i}$ with $\dot{x}_i := flow(N) = -\nabla_{x_i} f_i(x_i) - K \nabla_{x_i} \varphi_i(\mathbf{x})$, denotes the flow conditions of the

[11] Technically, the model we propose for the agents does not correspond to an Input/Output hybrid automaton [24], but to a composition of synchronized automata. In our framework, the terms *input* and *output* are used for presentation purposes, exclusively.

[12] To simplify notation, we hereafter drop the dependence of the state variables on time t.

[13] The shorthand notation stands for $N := Navigate$.

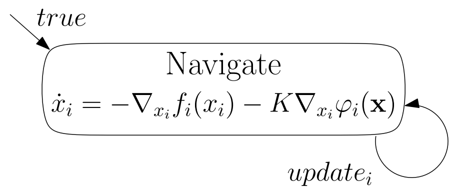


Fig. 7. Navigation Automaton for Agent i.

hybrid automaton that constrain the first time derivatives of the system variables in mode $v \in V_{N_i}$.

Figure 7 shows the graph representation of the hybrid automaton N_i. Observe that N_i consists of a single mode including agent i's dynamics as in Theorem 3.1, while self transitions in N_i are synchronized with transitions of agent i's coordination automaton due to the synchronization labels $sync((N, N)) = update_i$. Such transitions are triggered whenever the set of neighbors $\mathcal{N}_i$ of agent i is updated with new agents. In particular, we define the indicator function of an update in the neighbor set $\mathcal{N}_i$ by,

$$\mathcal{I}_{\mathcal{N}_i}^{new} := \neg\big(\wedge_{j=1}^{n} (a_{ij}^{[i]} \leftrightarrow h_{ij}^{[i]})\big)$$

where $h_{ij}^{[i]}(t)$ denotes the (i,j)-th entry of the matrix $H_i(t)$ in equation (5). Clearly, if $\mathcal{I}_{\mathcal{N}_i}^{new} = 0$, then no update has been done in $\mathcal{N}_i$.

On the other hand, the coordination automaton C_i of agent i implements the auction mechanism in Algorithm 1, while it also computes the set of neighbors $\mathcal{N}_i$ and updates the spanning subgraph estimate A_i. Note that, in order to account for communication time delays, transitions of the coordination automaton are *time triggered*. In particular, C_i stays in each mode for time $\tau_i \geq 0$, which has to be finite but may vary after every transition according to any policy, deterministic or not. Moreover, we define the quantities,

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

indicating the existence of at least one token in T_i or the existence of exactly n tokens in T_i, respectively. Hence, we have the following definition.

Definition 4.3 (Coordination Hybrid Automaton): We define the coordination hybrid automaton of agent i to be the tuple $C_i = (X_{C_i}, V_{C_i}, E_{C_i}, \Sigma_{C_i}, sync, inv, init,\ guard, reset, flow)$, where,

- $X_{C_i} = \{t, t_0^i, r_i, w_i, T_i^z, \mathcal{M}_i^z, A_i^z, \mathcal{N}_i^r, \mathcal{N}_i, \mathcal{S}_i \mid z = a, b, c\}$ denotes the set of owned state variables with $t, t_0^i \in \mathbb{R}_+$, $r_i \in \mathbb{R}^3$, $\mathcal{N}_i^r, \mathcal{N}_i, \mathcal{S}_i \in 2^{\mathbb{N}}$, $w_i \in \mathbb{N}^2$ and $T_i^z \in \{0,1\}^n$, $\mathcal{R}_i^z \in 2^{\mathbb{R}^3}$, $A_i^z \in \{0,1\}^{n\times n}$ for $z = a, b, c$.
- $V_{C_i} = \{Init, S_z, B_z, U_z, D_z \mid z = a, b, c\}$ denotes the finite set of control modes.[14]
- $E_{C_i} = \{(Init, B_a), (S_z, B_z), (B_z, U_z), (U_z, U_z),\ (U_z, D_z), (D_z, S_{z+1(\text{mod}3)}) \mid z = a, b, c\}$, denotes the set of control switches.
- $\Sigma_{C_i} = \{update_i\}$ denotes the set of synchronization labels.

[14]The shorthand notation stands for $B := Bid$, $U := Update$, $D := Decision$, $S := Start$.

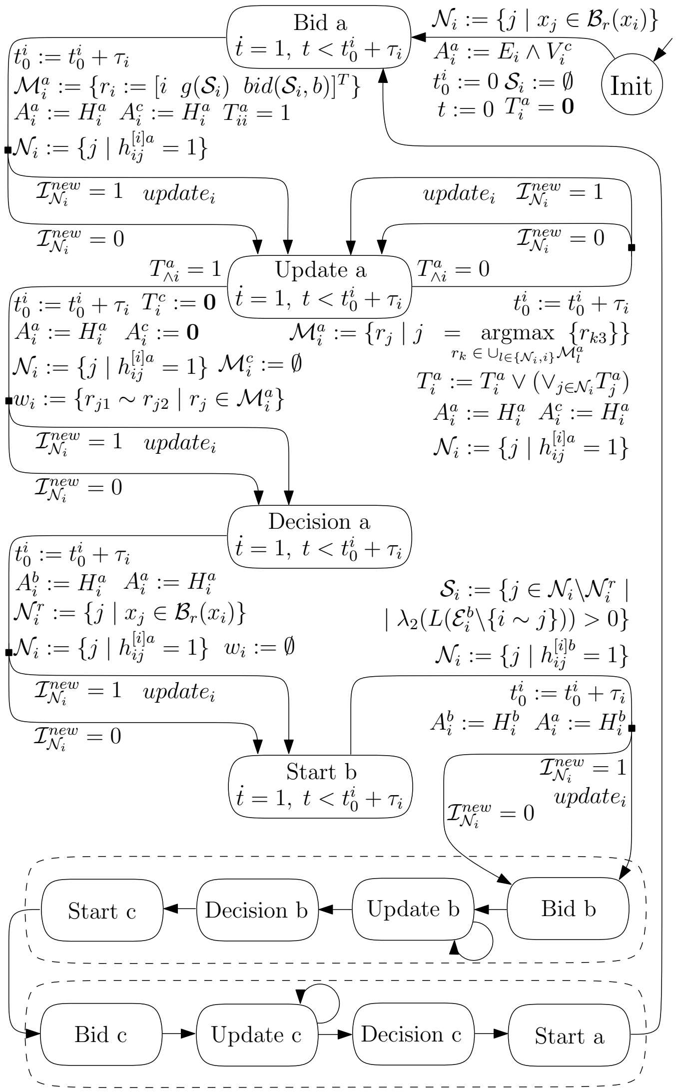


Fig. 8. Coordination Automaton for Agent i.

- $sync : E_{C_i} \rightarrow \Sigma_{C_i}$ with, $sync(e) = update_i$ if $\mathcal{I}^{new}_{\mathcal{N}_i} = 1$ for all $e \in E_{C_i}$, denotes the synchronization map mapping each control switch to a synchronization label.
- $inv : V_{C_i} \rightarrow Pred(X_{C_i})$ with $inv(v) = (t < t^i_0 + \tau_i)$ for all $v \in V_{C_i} \backslash \{Init\}$, denotes the invariant conditions of the hybrid automaton.
- $init : V_{C_i} \rightarrow Pred(X_{C_i})$ with $init(Init) = true$, denotes the set of initial conditions.
- $guard : E_{C_i} \rightarrow Pred(X_{C_i})$ with,
  - $guard\big((U_z, U_z)\big) = (T^z_{\wedge i} = 0)$ for $z = a, b, c$,
  - $guard\big((U_z, D_z)\big) = (T^z_{\wedge i} = 1)$ for $z = a, b, c$,
  - $guard(e) = true$, otherwise,

denotes the set of guards of the hybrid automaton.

- $reset : E_{C_i} \to X_{C_i}$ with, $[t\ t_0^i\ r_i\ w_i\ T_i^z\ \mathcal{M}_i^z\ A_i^z\ \mathcal{N}_i^r\ \ \mathcal{N}_i\ \mathcal{S}_i]_{z=a,b,c} := reset(e)$ such that,
  - if $e = (Init, B_a)$ we have, $t := 0$, $t_0^i := 0$, $T_i^a := \mathbf{0}$, $A_i^a := E_i \wedge V_i^c$, $\mathcal{N}_i := \{j \mid x_j \in \mathcal{B}_r(x_i)\}$ and $\mathcal{S}_i := \emptyset$.
  - if $e = (B_z, U_z)$ we have, $t_0^i := t_0^i + \tau_i$, $T_{ii}^z =: 1$, $\mathcal{M}_i^z := \{r_i := [i\ g(\mathcal{S}_i)\ bid(\mathcal{S}_i, b)]^T\}$, $A_i^z := H_i^z$, $A_i^{z-1(\text{mod}3)} := H_i^z$ and $\mathcal{N}_i := \{j \mid h_{ij}^{[i]z} = 1\}$.
  - if $e = (U_z, U_z)$ we have, $t_0^i := t_0^i + \tau_i$, $T_i^z := T_i^z \vee \big(\vee_{j\in\mathcal{N}_i} T_j^z\big)$, $\mathcal{M}_i := \big\{r_j \mid j = \text{argmax}_{r_k \in \cup_{l\in\{\mathcal{N}_i, i\}}\mathcal{M}_l}\{r_{k3}\}\big\}$, $A_i^z := H_i^z$, $A_i^{z-1(\text{mod}3)} := H_i^z$ and $\mathcal{N}_i := \{j \mid h_{ij}^{[i]z} = 1\}$.
  - if $e = (U_z, D_z)$ we have, $t_0^i := t_0^i + \tau_i$, $w_i := \{r_{j1} \sim r_{j2} \mid r_j \in \mathcal{M}_i^z\}$, $T_i^{z-1(\text{mod}3)} := \mathbf{0}$, $\mathcal{M}_i^{z-1(\text{mod}3)} := \emptyset$, $A_i^z := H_i^z$, $A_i^{z-1(\text{mod}3)} := \mathbf{0}$ and $\mathcal{N}_i := \{j \mid h_{ij}^{[i]z} = 1\}$.
  - if $e = (D_z, S_z)$ we have, $t_0^i := t_0^i + \tau_i$, $w_i := \emptyset$, $A_i^{z+1(\text{mod}3)} := H_i^z$, $A_i^z := H_i^z$ $\mathcal{N}_i^r := \{j \mid x_j \in \mathcal{B}_r(x_i)\}$ and $\mathcal{N}_i := \{j \mid h_{ij}^{[i]z} = 1\}$.
  - if $e = (S_z, B_{z+1(\text{mod}3)})$ we have, $t_0^i := t_0^i + \tau_i$, $A_i^{z+1(\text{mod}3)} := H_i^{z+1(\text{mod}3)}$, $A_i^z := H_i^{z+1(\text{mod}3)}$, $\mathcal{N}_i := \{j \mid h_{ij}^{[i]z+1(\text{mod}3)} = 1\}$ and $\mathcal{S}_i := \{j \in \mathcal{N}_i \backslash \mathcal{N}_i^r \mid \lambda_2(L(\mathcal{E}_i^{z+1(\text{mod}3)} \backslash \{i \sim j\})) > 0\}$.

  denotes the set of resets associated with the guards of the hybrid automaton.

- $flow : V_{C_i} \to \dot{X}_{C_i}$ with $[\dot{t}\ \dot{t}_0^i\ \dot{r}_i\ \dot{w}_i\ \dot{T}_i^z\ \dot{\mathcal{M}}_i^z\ \dot{A}_i^z\ \dot{\mathcal{N}}_i^r\ \dot{\mathcal{N}}_i\ \ \dot{\mathcal{S}}_i]_{z=a,b,c} := flow(v) = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$ for all $v \in V_{C_i}$, denotes the flow conditions of the hybrid automaton that constrain the first time derivatives of the system variables in mode $v \in V_{C_i}$.

By Definition 4.3, in order to achieve the desired synchronization of all coordination automata on the same auction, the coordination automata consist of a triple of identical auctions labeled a, b and c respectively, such that all variables (i.e., max-bids and links) of automaton C_i in any auction $z = a, b, c$ are updated with variables from automata C_j that are in the same auction z. Furthermore, the outcome of one auction is the input to the other. Hence, a sequence of auctions is always of the form $\{a, b, c, a, b, c, \dots\}$ and, as it is shown in Section V, if all agents are initialized in auctions with same labels, then they are always synchronized in auctions with same labels, despite any possible time delays. Figure 8 shows the graph representation of hybrid automaton C_i.

V. Integration of the Overall System

Having defined the coordination automata of the agents we now proceed with their composition [23] and study the properties of the overall system.

Definition 5.1 (Product System): We define the product of the hybrid automata $N_1, \dots, N_n, C_1, \dots, C_n$ by the tuple $S = (X_S, V_S, E_S, \Sigma_S, sync, inv, init, guard, reset,\ flow)$, where,

- $X_S = X_{N_1} \cup \cdots \cup X_{N_n} \cup X_{C_1} \cup \cdots \cup X_{C_n}$ denotes the set of state variables.
- $V_S = V_{N_1} \times \cdots \times V_{N_n} \times V_{C_1} \times \cdots \times V_{C_n}$ denotes the finite set of control modes.
- $E_S = \{e_S\}$ denotes the set of control switches such that,
  - $e_S = e_{N_i} \| e_{C_i} \in E_S$ is defined as the set of control switches of S corresponding to control switches $e_{N_i} \in E_{N_i}$ and $e_{C_i} \in E_{C_i}$, with $sync(e_{N_i}) = sync(e_{C_i}) = update_i$,

  - $e_S = (v_S, v'_S) \in E_S$ otherwise.

  Hence, the only variables that change with every transition $v_S \overset{e_S}{\to} v'_S$ are the ones involved in the control switch e_S through the corresponding automata.

- $\Sigma_S = \Sigma_{N_1} \cup \cdots \cup \Sigma_{N_n} \cup \Sigma_{C_1} \cup \cdots \cup \Sigma_{C_n}$ denotes the set of synchronization labels.
- $sunc : E_S \to \Sigma_S$ denotes the synchronization map mapping each control switch to a synchronization label.
- $inv : V_S \to Pred(X_S)$ with $inv(v_S) = inv(v_{N_1}) \wedge \cdots \wedge inv(v_{N_n}) \wedge inv(v_{C_1}) \wedge \cdots \wedge inv(v_{C_n})$ for all $v_S \in V_S$, denotes the invariant conditions of the product automaton.
- $init : V_S \to Pred(X_S)$ with $init(v_S) = init(v_{N_1}) \wedge \cdots \wedge init(v_{N_n}) \wedge init(v_{C_1}) \wedge \cdots \wedge init(v_{C_n})$ for all $v_S \in V_S$, denotes the set of initial conditions.
- $guard : E_S \to Pred(X_S)$ such that $guard(e_S) = guard(e_{N_i}) \wedge guard(e_{C_i})$ if $e_S = e_{N_i} \| e_{C_i} \in E_S$, and $guard(e_S) = guard\big((v_S, v'_S)\big)$ otherwise, denotes the set of guards (or transitions) of the hybrid automaton.
- $reset : E_S \to X_S$ such that $reset(e_S) = reset(e_{N_i}) \wedge reset(e_{C_i})$ if $e_S = e_{N_i} \| e_{C_i} \in E_S$, and $reset(e_S) = reset\big((v_S, v'_S)\big)$ otherwise, denotes the set of resets associated with the guards of the hybrid automaton.
- $flow : V_S \to \dot{X}_S$ with $flow(v_S) = flow(v_{N_1}) \cup \cdots \cup flow(v_{N_n}) \cup flow(v_{C_1}) \cup \cdots \cup flow(v_{C_n})$ for all $v_S \in V_S$, denotes the flow conditions of the hybrid automaton that constrain the first time derivatives of the system variables in mode $v_S \in V_S$.

Clearly, the product system S, being the composition of all elementary automata N_i and C_i, models the interconnection between them. Hence, studying S, we can identify the properties of the whole mobile network. In particular, in Subsection V-A we show that the market-based coordination framework implemented in S is sound, while in Subsection V-B we show correctness of the overall system S, namely, that it guarantees connectivity of the mobile network.

A. *Soundness of Market-Based Coordination*

Let $\{z_k\}_{k=1}^{\infty} = \{a, b, c, a, b, c, \dots\}$ denote the sequence of auctions for any automaton C_i. Due to the possible time delays that are modeled in every automaton C_i, the product system S is not necessarily synchronized in the same auction for all automata C_i and all time. This lack of synchronization can cause the overall market-based coordination framework to fail. In view of this observation, we can define a *sound* coordination framework as follows.

Definition 5.2 (Soundness of Market-Based Coordination): We say that the market-based coordination framework implemented in the product system S is *sound* if there are no deadlocks in S and if the outcome of very auction z_k is common for all automata C_i.

Note that, due to time delays in our system, it is possible that there exist distinct automata C_i and C_j that at the same time instant are in distinct auctions z_k and z_{k+1}.[15] Clearly, if $T_{\wedge j}^{z_k} = 0$ for all automata C_j that are in auction z_{k+1}, while $T_{ij}^{z_k} = 0$ for all automata C_i in auction z_k, then the guards $guard((U_{z_k}, D_{z_k})) = (T_{\wedge i}^{z_k} = 1)$

[15] We say that automaton C_i is in auction z_k at time t if $v_{C_i}(t) = S_{z_k}, B_{z_k}, U_{z_k}, D_{z_k}$.

in the automata C_i can not be enabled, or equivalently the max-consensus algorithm on the max-bid sets $\mathcal{M}_i^{z_k}$ for auction z_k can not terminate, and so the system will deadlock.

On the other hand, for any auction z_k, suppose that $v_{C_i}^s = U_{z_k}$ for $t = t_s$ and that at time $t = t_{s+1}$ automaton C_i transitions to mode $v_{C_i}^{s+1} = U_{z_k}$. Then, the corresponding reset becomes, $\mathcal{M}_i^{z_k}(t_{s+1}) := reset\big((v_{C_i}^s, v_{C_i}^{s+1})\big) = \big\{r_j \mid j = \text{argmax}_{r_p \in \cup_{l \in \{\mathcal{N}_i(t_s), i\}} \mathcal{M}_l^{z_k}(t_s)}\{r_{p3}\}\big\}$, i.e., the set of maximum bids $\mathcal{M}_i^{z_k}(t_s)$ is updated with bids from auction z_k exclusively. However, since $z_{k-3} = z_k \pmod 3$, auction z_{k-3} can contribute to the updates of $\mathcal{M}_i^{z_k}$ too. Clearly, such a scenario, prevents all automata from converging to a common outcome for every auction.

These observations imply that soundness of the proposed market-based coordination framework is necessary for correctness of the overall distributed system S. On the other hand, the token dynamics of any automaton C_i explicitly indicate the beginning and termination of the corresponding auction from the perspective of automaton C_i. Hence, we can study soundness of the proposed coordination framework through the underlying token dynamics. In particular, we have the following results.

Proposition 5.3: For any time t, suppose there exists an automaton C_i such that $T_{\wedge i}^{z_k}(t) = 1$ for any auction z_k. Then, $T_{\vee j}^{z_k}(t) = 1$ for all C_j with $j \neq i$.

Proof: To show this result we use induction on the sequence of auctions z_k. Suppose that for $t = t_s$ there exists an automaton C_i such that $T_{\wedge i}^{z_1}(t_s) = 1$. Then, since z_1 is the first auction to take place, we necessarily have that $T_{\vee j}^{z_1}(t_s) = 1$ for all automata C_j with $j \neq i$.

Suppose, now, that if at any time instant $t_{s+1} > t_s$ there exists an automaton C_i such that $T_{\wedge i}^{z_k}(t_{s+1}) = 1$, then $T_{\vee j}^{z_k}(t_{s+1}) = 1$ for all C_j with $j \neq i$ and any auction z_k. But for all $j \neq i$, the induction hypothesis $T_{\vee j}^{z_k}(t_{s+1}) = 1$ implies that in the worst case $v_{C_j}^{s+1} = U_{z_k}$ (C_j can also be in any subsequent mode). Hence, $T_{\wedge j}^{z_{k-2}}(t_{s+1}) = 0$ for all j due to the transition $U_{z_{k-1}} \to D_{z_{k-1}}$ that has taken place in each C_j at some time instant $t < t_{s+1}$, which implies that $T_{\wedge j}^{z_{k-2}}(t) = 0$ for all j and all time $t > t_{s+1}$ (note that in the worst case $v_{C_i}^{s+1} = D_{z_k}$, hence the result holds for automaton C_i too).

Now, let $t_{s+2} > t_{s+1}$ denote a time instant such that $T_{\wedge i}^{z_{k+1}}(t_{s+2}) = 1$ for any automaton C_i and note that z_{k-2} is the latest auction such that $z_{k+1} = z_{k-2} \pmod 3$. But, by induction hypothesis $T_{\wedge j}^{z_{k-2}}(t) = 0$ for all j and all time $t > t_{s+1}$. Hence, for all time $t_{s+1} < t < t_{s+2}$ such that $v_{C_i}^{s+2} = U_{z_{k+1}}$, the token vectors $T_j^{z_{k-2}}(t)$ for $j \neq i$ can not contribute to the updates of $T_i^{z_{k+1}}(t)$, which implies that a necessary condition for $T_{\wedge i}^{z_{k+1}}(t_{s+2}) = 1$ is that $T_{\vee j}^{z_{k+1}}(t_{s+2}) = 1$ for all $j \neq i$. ■

Proposition 5.4: For any time t, suppose there exists an automaton C_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 C_j with $j \neq i$ and any 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 $j \neq i$.

Proof: Suppose that for any time instant t there exists an automaton C_j with $j \neq i$ such that $T_{\wedge j}^{z_{k+1}}(t) = 1$. Then, by Proposition 5.3, $T_{\vee l}^{z_{k+1}}(t) = 1$ for all $l \neq j$. But by assumption, there exists an automaton C_i such that $T_{\vee i}^{z_k}(t) = 1$ and $T_{\wedge i}^{z_k}(t) = 0$ or equivalently $T_{\wedge i}^{z_{k+1}}(t) = 0$, since automaton C_i is not in mode $U_{z_{k+1}}$ yet, which is clearly a contradiction. Hence, $T_{\vee j}^{z_{k+1}}(t) = 1$ and $T_{\wedge j}^{z_{k+1}}(t) = 0$ for all $j \neq i$.

Now, since $T_{\vee j}^{z_{k+1}}(t) = 1$ and $T_{\wedge j}^{z_{k+1}}(t) = 0$, we have that $v_{C_j} = U_{z_{k+1}}$ for all $j \neq i$, which clearly implies that $T_{\wedge j}^{z_k}(t) = 1$ for all $j \neq i$ (since the token vectors $T_j^{z_k}(t)$ are reset to zero with the transition $U_{z_{k+1}} \to D_{z_{k+1}}$ which takes place when $T_{\wedge j}^{z_{k+1}}(t) = 1$). ■

In other words, Proposition 5.4 says that if there exists an automaton C_i that due to possible time delays is still in auction z_k, while all other automata C_j with $j \neq i$ are in auction z_{k+1}, then all automata C_j with $j \neq i$ will remain in auction z_{k+1} until automaton C_i receives all tokens for auction z_k, while they will also maintain their max-bid sets $\mathcal{M}_j^{z_k}$ to make sure that the max-bid set $\mathcal{M}_i^{z_k}$ of automaton C_i for auction z_k will eventually converge too. Note that this result also implies a form of synchronization among all automata, since the *faster* ones periodically wait for their *slower* peers, preventing the occurrence of any deadlocks. Now, let,

$$T_{\vee}^{z_k}(t) := \vee_{i=1}^{n} T_{ii}^{z_k}(t) \quad \text{and} \quad T_{\wedge}^{z_k}(t) := \wedge_{i=1}^{n} T_{ii}^{z_k}(t) \tag{12}$$

denote the existence of at least one token or exactly n tokens in auction z_k, respectively. Clearly, if $T_{\vee}^{z_k}(t) = 1$ there exists at least one automaton C_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 automata C_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. 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 $t_{I_k} = \min\{t \mid T_{\vee}^{z_k}(t) = 1\}$, $t_{B_k} = \min\{t \mid T_{\wedge}^{z_k}(t) = 1\}$ and $t_{F_k} = \max\{t \mid T_{\vee}^{z_k}(t) = 1\}$, respectively. We hence have the following result.

Proposition 5.5: Let z_{k-2}, z_{k-1}, z_k and z_{k+1} be any sequence of auctions. Then, $t_{F_{k-2}} < t_{B_k} < t_{I_{k+1}} < t_{F_{k-1}}$.

Proof: For any time instant t let C_i be the *first* automaton such that $T_{\wedge i}^{z_k}(t) = 1$ for any auction z_k. Then, clearly $T_{\vee j}^{z_{k+1}}(t) = 0$ for all j which implies that $t < t_{I_{k+1}}$. On the other hand, since $T_{\vee j}^{z_k}(t) = 1$ by Proposition 5.3 for all j, we have that $t_{B_k} < t$. Hence, $t_{B_k} < t_{I_{k+1}}$. Note also that in order that $T_{\vee j}^{z_k}(t) = 1$ for all j it is necessary that $T_{\vee j}^{z_{k-2}}(t) = 0$ for all j, which implies that $t_{F_{k-2}} < t_{B_k}$. Hence, $t_{F_{k-2}} < t_{B_k} < t_{I_{k+1}}$.

On the other hand, due to time delays in the system, it is possible that for some time instant t there exist automata C_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 $j \neq i$ and any auctions z_k and z_{k+1}. Clearly, since there exist automata i such that $T_{\vee i}^{z_k}(t) = 1$ and $T_{\wedge i}^{z_k}(t) = 0$ we have that $t < t_{F_{k-1}}$. Moreover, since $T_{\vee j}^{z_{k+1}}(t) = 1$ for all $j \neq i$ we also have that $t_{I_{k+1}} < t$. Hence, $t_{I_{k+1}} < t_{F_{k-1}}$ which completes the proof. ■

Hence, the sequence of auctions $\{z_k\}_{k=1}^{\infty}$ is according to the following scheme,

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

Note also that, since $t_{F_{k-2}} < t_{I_{k+1}}$, no contributions from past auctions can be made in updating the variables of current auctions, even though $z_{k-2} = z_{k+1} \pmod 3$. Hence, we have the following result.

Proposition 5.6: For any automaton C_i and any auction z_k, eventually $T_{\wedge i}^{z_k}(t) = 1$. Moreover, all automata C_i with $T_{\wedge i}^{z_k}(t) = 1$ share identical max-bid sets $\mathcal{M}_i^{z_k}(t)$.

Proof: Consider the transition $v_{C_i}^s \to v_{C_i}^{s+1}$ of automaton C_i at time $t = t_s$ with $v_{C_i}^s = B_{z_k}$ and $v_{C_i}^{s+1} = U_{z_k}$. Then, by Definition 4.3, the corresponding reset becomes $[T_{ii}^{z_k}(t_{s+1}) \ \mathcal{M}_i^{z_k}(t_{s+1})] := reset\big((v_{C_i}^s, v_{C_i}^{s+1})\big) =$

$[1\ \{r_i(t_{s+1})\}]$ and the guard $guard((v_{C_i}^{s+1}, v_{C_i}^{s+2})) = (T_{\wedge i}^{z_k}(t_{s+1}) = 0)$ with $v_{C_i}^{s+2} = U_{z_k}$ is enabled. Hence, the transition $v_{C_i}^{s+1} \to v_{C_i}^{s+2}$ results in,

$$T_i^{z_k}(t_{s+2}) := reset\big((v_{C_i}^{s+1}, v_{C_i}^{s+2})\big) = \big(T_i^{z_k}(t_{s+1}) \vee \big(\vee_{j \in \mathcal{N}_i(t_{s+1})} T_j^{z_k}(t_{s+1})\big)\big) \leftarrow T_i^{z_k}(t_{s+1})$$

Clearly, with every subsequent transition $v_{C_i}^{p} \to v_{C_i}^{p+1}$, $p \geq s+2$, of automaton C_i with $v_{C_i}^{p} = U_{z_k}$ and $v_{C_i}^{p+1} = U_{z_k}$, we have $T_i^{z_k}(t_p) \to T_i^{z_k}(t_{p+1})$ and in the absence of deadlocks (Proposition 5.4), eventually $T_{\wedge i}^{z_k}(t) = 1$ for a connected network.

To show that all automata C_i with $T_{\wedge i}^{z_k}(t) = 1$ share identical max-bid sets $\mathcal{M}_i^{z_k}(t)$, just note that for every transition $U_{z_k} \to U_{z_k}$ of any automaton C_i only auction z_k can contribute in updating $\mathcal{M}_i^{z_k}(t)$, by Proposition 5.5. ∎

Proposition 5.6 implies that consensus among all automata C_i is guaranteed for every auction z_k, while, by equation (11), the outcome of any auction z_k is common for all automata C_i. Hence, Propositions 5.4 and 5.6 imply that the proposed market-based coordination framework is sound.

B. Correctness of the Overall System

Having shown soundness of the proposed market-based coordination framework, our goal now is to study how every automaton selects a link and the associated bid, and show that the proposed selection process is *safe* with respect to connectivity. The rest of this section is devoted to the study of the dynamics (5) of the local variables $A_i(t)$ and the way the estimates $A_i(t)$ relate to the adjacency matrix of the global graph $A(t)$. Resulting from this analysis is the main contribution of this section, namely, correctness of the overall system S.

Observe first that the resets associated with the transitions $U_{z_k} \to D_{z_k}$ and the analysis in Subsection V-A imply that $A_j^{z_k}(t) = \mathbf{0}$ for all automata C_j and all time $t > t_{F_k}$. Hence, auction z_{k-3} can not contribute to the updating of the variables $A_i^{z_k}(t)$ in auction z_k, even though $z_{k-3} = z_k (\mathrm{mod} 3)$. In other words, updating of the variables $A_i^{z_k}(t)$ is sound too. In the following result we show that the winning link $w_i(t_s)$ of auction z_k is always deleted from the edge set $\mathcal{E}_i^{z_z}(t)$, i.e., that $w_i(t_s) \notin \mathcal{E}_i^{z_{k+1}}(t_{s+1})$.[16]

Proposition 5.7: Let t_s be such that $v_{C_i}^{s} = D_{z_k}$ for any automaton C_i and any auction z_k. Then, $V_i^d(t_s) \to (\neg A_i^{z_{k+1}}(t_{s+1}))$.

Proof: Let t_s be such that $v_{C_i}^{s} = D_{z_k}$ for any automaton C_i and any auction z_k. Then, at time t_{s+1} automaton C_i transitions to mode $v_{C_i}^{s+1} = S_{z_{k+1}}$ and the reset $A_i^{z_{k+1}}(t_{s+1}) := reset((v_{C_i}^{s}, v_{C_i}^{s+1})) = H_i^{z_k}(t_s)$ implies that,

$$\begin{aligned}
\big(V_i^d(t_s) \wedge (\neg A_i^{z_{k+1}}(t_{s+1}))\big) \quad &\leftrightarrow \quad \big(V_i^d(t_s) \wedge \big(A_i^{z_k}(t_s) \leftrightarrow V_i(t_s)\big)\big) \\
&\leftrightarrow \quad \big(V_i^d(t_s) \wedge \big((A_i^{z_k}(t_s) \wedge V_i(t_s)) \vee \big(\neg(A_i^{z_k}(t_s) \vee V_i(t_s))\big)\big)\big) \\
&\leftrightarrow \quad \big(A_i^{z_k}(t_s) \wedge V_i^d(t_s)\big) \leftrightarrow V_i^d(t_s)
\end{aligned}$$

since $V_i^d(t_s) \to V_i(t_s)$, $V_i^d(t_s) \to A_i^{z_k}(t_s)$ and $H_i^{z_k}(t_s) = \neg(A_i^{z_k}(t_s) \leftrightarrow V_i(t_s))$. The result follows by Lemma 1.4(f). ∎

[16]Note the direct relation between the adjacency matrix $A_i(t)$ of a graph and the associated set of edges $\mathcal{E}_i(t)$.

Clearly, Proposition 5.7 implies that the outcome of every auction is always the deletion of at most one link from $\mathcal{E}_i(t)$. The following result provides a relation between the edge sets $\mathcal{E}_i(t_s)$ and $\mathcal{E}_i(t_{s+1})$ after every transition $v^s_{C_i} \to v^{s+1}_{C_i}$ of automaton C_i.

Proposition 5.8: For any time t_s, consider transition $v^s_{C_i} \to v^{s+1}_{C_i}$ of automaton C_i such that $A_i(t_{s+1}) = H_i(t_s)$, as in equation (5). Then, $A_i(t_s) \to \big(A_i(t_{s+1}) \vee V_i^d(t_s)\big)$.

Proof: For any time t_s, consider the transition $v^s_{C_i} \to v^{s+1}_{C_i}$ of automaton C_i and observe that,

$$\big(A_i(t_s) \wedge \big(A_i(t_{s+1}) \vee V_i^d(t_s)\big)\big) \leftrightarrow \big(\big(A_i(t_s) \wedge A_i(t_{s+1})\big) \vee \big(A_i(t_s) \wedge V_i^d(t_s)\big)\big)$$

by Lemma 1.4(a) in the Appendix. Moreover, by equation (5) and Lemma 1.4(g) in the Appendix, we have that,

$$\begin{aligned}
\big(A_i(t_s) \wedge A_i(t_{s+1})\big) &\leftrightarrow \big(A_i(t_s) \wedge \big(\neg(A_i(t_s) \leftrightarrow V_i(t_s))\big)\big) \\
&\leftrightarrow \big(A_i(t_s) \wedge \big(\neg(A_i(t_s) \wedge V_i(t_s))\big) \wedge \big(A_i(t_s) \vee V_i(t_s)\big)\big) \\
&\leftrightarrow \big(A_i(t_s) \wedge \big(\neg(A_i(t_s) \wedge V_i^d(t_s))\big) \wedge \big(A_i(t_s) \vee V_i(t_s)\big)\big) \\
&\leftrightarrow \big(\big(\neg(A_i(t_s) \wedge V_i^d(t_s))\big) \wedge \big(A_i(t_s) \vee (A_i(t_s) \wedge V_i(t_s))\big)\big) \\
&\leftrightarrow \big(\big(\neg(A_i(t_s) \wedge V_i^d(t_s))\big) \wedge \big(A_i(t_s) \vee (A_i(t_s) \wedge V_i^d(t_s))\big)\big) \\
&\leftrightarrow \big(\big(\neg(A_i(t_s) \wedge V_i^d(t_s))\big) \wedge A_i(t_s)\big) \leftrightarrow \big(A_i(t_s) \wedge (\neg V_i^d(t_s))\big)
\end{aligned}$$

since $(A_i(t_s) \wedge V_i(t_s)) \leftrightarrow (A_i(t_s) \wedge V_i^d(t_s))$, by equation (7). Hence,

$$\begin{aligned}
\big(A_i(t_s) \wedge \big(A_i(t_{s+1}) \vee V_i^d(t_s)\big)\big) &\leftrightarrow \big(\big(A_i(t_s) \wedge (\neg V_i^d(t_s))\big) \vee \big(A_i(t_s) \wedge V_i^d(t_s)\big)\big) \\
&\leftrightarrow \big(A_i(t_s) \wedge \big(A_i(t_s) \vee (\neg V_i^d(t_s))\big) \wedge \big(A_i(t_s) \vee V_i^d(t_s)\big)\big) \\
&\leftrightarrow \big(A_i(t_s) \wedge \big(A_i(t_s) \vee V_i^d(t_s)\big)\big) \leftrightarrow A_i(t_s)
\end{aligned}$$

since $\big(A_i(t_s) \wedge \big(A_i(t_s) \vee (\neg V_i^d(t_s))\big)\big) \leftrightarrow A_i(t_s)$ and $\big(A_i(t_s) \wedge \big(A_i(t_s) \vee V_i^d(t_s)\big)\big) \leftrightarrow A_i(t_s)$ by Lemma 1.4(e) in the Appendix. ∎

In other words, Proposition 5.8 implies that with every transition $v^s_{C_i} \to v^{s+1}_{C_i}$ of automaton C_i we have $\mathcal{E}_i(t_s) \subseteq \mathcal{E}_i(t_{s+1}) \cup \{w_i(t_s)\}$. This fact 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 define the *winning* link of auction z_k by,

$$w(t) := \{r_{i1} \sim r_{i2} \mid r_i \in \mathcal{M}^{z_k}(t)\} \tag{13}$$

where $\mathcal{M}^{z_k}(t) := \big\{r_i \mid i = \text{argmax}_{r_j \in \cup_{l=1}^n \mathcal{M}_l^{z_k}(t)}\{r_{j3}\}\big\}$ denotes the global set of max-bids and denote by $Q(t) = (q_{jk}(t))$ the boolean matrix with entries,

$$q_{jk}(t) := (w(t) = j \sim k) \wedge (|w(t)| = 1) \tag{14}$$

indicating the existence of a winning link for auction z_k. By Propositions 5.5 and 5.6, note that the set $\mathcal{M}^{z_k}(t)$ is constant for all time $t \in [t_{B_k}, t_{F_k}]$ and hence, for all time $t \in [t_{B_k}, t_{B_{k+1}})$ as well, since $[t_{B_k}, t_{B_{k+1}}) \subseteq [t_{B_k}, t_{F_k}]$. Thus, the winning link $w(t)$ and, hence, the boolean matrix $Q(t)$, are constant for all time $t \in [t_{B_k}, t_{B_{k+1}})$ and are exclusively associated with the outcome of auction z_k. Note also that $V_i^d(t) \to Q(t)$ for any automaton C_i and all time $t \in [t_{B_k}, t_{B_{k+1}})$, since if the time instant t is such that $v_{C_i}(t) = D_{z_k}$, then $t \in [t_{B_k}, t_{B_{k+1}})$.[17] Finally, let,

$$A(t) := \vee_{i=1}^n (E_i \wedge A_i(t)) \tag{15}$$

denote the adjacency matrix of the global graph. Then,

Proposition 5.9: For any time instant $t_s \in [t_{B_k}, t_{B_{k+1}})$, consider transition $v_S^s \to v_S^{s+1}$ of the product system S, which is due to transition $v_{C_i}^s \to v_{C_i}^{s+1}$ of any automaton C_i such that $A_i(t_{s+1}) = H_i(t_s)$, as in equation (5). Then, $A(t_s) \to \big(A(t_{s+1}) \vee Q(t_s)\big)$.

Proof: Observe first that, since $\vee_i E_i \leftrightarrow \mathbf{1}_{n\times n}$ we have that $Q(t_s) \leftrightarrow \vee_i (E_i \wedge Q(t_s))$. Moreover, since $V_i^d(t_s) \to Q(t_s)$ Proposition 5.8 implies that $A_i(t_s) \to \big(A_i(t_{s+1}) \vee Q(t_s)\big)$ and so $\big(E_i \wedge A_i(t_s)\big) \to \big(E_i \wedge \big(A_i(t_{s+1}) \vee Q(t_s)\big)\big)$, by Lemma 1.5(a) in the Appendix. Hence, by Lemma 1.5(b) in the Appendix, we get,

$$\begin{aligned}
\big(A(t_s) \wedge \big(A(t_{s+1}) \vee Q(t_s)\big)\big) &\leftrightarrow \big((\vee_i (E_i \wedge A_i(t_s))) \wedge \big((\vee_i (E_i \wedge A_i(t_{s+1}))) \vee (\vee_i (E_i \wedge Q(t_s)))\big)\big) \\
&\leftrightarrow \big((\vee_i (E_i \wedge A_i(t_s))) \wedge \big(\vee_i \big((E_i \wedge A_i(t_{s+1})) \vee (E_i \wedge Q(t_s))\big)\big)\big) \\
&\leftrightarrow \big((\vee_i (E_i \wedge A_i(t_s))) \wedge \big(\vee_i \big(E_i \wedge (A_i(t_{s+1}) \vee Q(t_s))\big)\big)\big) \\
&\leftrightarrow \big(\vee_i \, (E_i \wedge A_i(t_s))\big) \leftrightarrow A(t_s)
\end{aligned}$$

which, by Lemma 1.4(f) in the Appendix completes the proof. ■

In other words, Proposition 5.9 implies that for all time $t_s \in [t_{B_k}, t_{B_{k+1}})$, the transition $v_S^s \to v_S^{s+1}$ of the product system S results in $\mathcal{E}(t_s) \subseteq \mathcal{E}(t_{s+1}) \cup \{w(t_s)\}$, which leads to the following result.

Proposition 5.10: For any time $t_s \in [t_{B_k}, t_{B_{k+1}})$, consider transition $v_{C_i}^s \to v_{C_i}^{s+1}$ of automaton C_i such that $A_i(t_{s+1}) = H_i(t_s)$, as in equation (5). Then, $A_i(t_{s+1}) \to \big(A(t_{s+1}) \vee Q(t_s)\big)$. Moreover, if $t = t_s$ is such that $v_{C_i}^s = S_{z_k}$, then $A_i(t_s) \to A(t_s)$.

Proof: By equations (6) and (7), for all time t we have,

$$\begin{aligned}
V_i(t) &\leftrightarrow \big((F_i(t) \wedge V_i^c(t)) \vee (A_i(t) \wedge V_i^d(t))\big) \\
&\leftrightarrow \big(\big(\big(((\neg A_i(t)) \wedge A_i^{(1)}(t)) \vee (E_i \wedge (\neg A_i(t)))\big) \wedge V_i^c(t)\big) \vee (A_i(t) \wedge V_i^d(t))\big) \\
&\leftrightarrow \big(\big((\neg A_i(t)) \wedge A_i^{(1)}(t) \wedge V_i^c(t)\big) \vee \big(E_i \wedge (\neg A_i(t)) \wedge V_i^c(t)\big) \vee (A_i(t) \wedge V_i^d(t))\big) \\
&\to \big(\big((\neg A_i(t)) \wedge A_i^{(1)}(t) \wedge V_i^c(t)\big) \vee \big(E_i \wedge (\neg A_i(t)) \wedge V_i^c(t)\big) \vee A_i(t)\big) \\
&\leftrightarrow \big(\big((A_i(t) \vee A_i^{(1)}(t)) \wedge (A_i(t) \vee V_i^c(t))\big) \vee \big(E_i \wedge (\neg A_i(t)) \wedge V_i^c(t)\big)\big) \\
&\to \big((A_i(t) \vee A_i^{(1)}(t)) \vee \big(E_i \wedge (\neg A_i(t)) \wedge V_i^c(t)\big)\big) \\
&\leftrightarrow \big(A_i^{(1)}(t) \vee \big((A_i(t) \vee E_i) \wedge (A_i(t) \vee V_i^c(t))\big)\big) \\
&\leftrightarrow \big(A_i(t) \vee A_i^{(1)}(t) \vee (E_i \wedge V_i^c(t))\big)
\end{aligned}$$

[17] Note that $w_i(t) \neq \emptyset$ only if $v_{C_i}(t) = D_{z_k}$, for any automaton C_i.

and so $V_i(t) \rightarrow \big(A_i(t) \vee A_i^{(1)}(t) \vee (E_i \wedge V_i^c(t))\big)$. Now, for any $t_s \in [t_{B_k}, t_{B_{k+1}})$ assume that $A_i(t_{s+1}) \rightarrow \big(A(t_{s+1}) \vee Q(t_s)\big)$. Note that this relation also implies that $A_i^{(1)}(t_{s+1}) \rightarrow \big(A(t_{s+1}) \vee Q(t_s)\big)$. Hence, for $t_{s+1} \in [t_{B_k}, t_{B_{k+1}})$, using the induction hypothesis and Proposition 5.9 we have,

$$\begin{aligned}
A_i(t_{s+2}) &\leftrightarrow \big((\neg(A_i(t_{s+1}) \wedge V_i(t_{s+1})))\big) \wedge (A_i(t_{s+1}) \vee V_i(t_{s+1}))\big) \\
&\rightarrow \big(A_i(t_{s+1}) \vee V_i(t_{s+1})\big) \\
&\rightarrow \big(A_i(t_{s+1}) \vee A_i^{(1)}(t_{s+1}) \vee (E_i \wedge V_i^c(t_{s+1}))\big) \\
&\rightarrow \big(A(t_{s+1}) \vee Q(t_s) \vee (E_i \wedge V_i^c(t_{s+1}))\big) \qquad \text{induction hyp.} \\
&\rightarrow \big(A(t_{s+2}) \vee Q(t_{s+1}) \vee Q(t_s) \vee (E_i \wedge V_i^c(t_{s+1}))\big) \\
&\rightarrow \big(A(t_{s+2}) \vee Q(t_{s+1}) \vee (E_i \wedge V_i^c(t_{s+1}))\big) \\
&\rightarrow \big(A(t_{s+2}) \vee Q(t_{s+1})\big)
\end{aligned}$$

since, $Q(t_s) \rightarrow Q(t_{s+1})$ for all time $t_s \in [t_{B_k}, t_{F_k})$ and $(E_i \wedge V_i^c(t_{s+1})) \rightarrow (E_i \wedge A_i(t_{s+2})) \rightarrow A(t_{s+2})$ by the definition of $V_i^c(t_s)$ and equation (15).

Suppose now, that $t_s \in [t_{B_k}, t_{B_{k+1}})$ is such that $v_{C_i}^s = D_{z_k}$. Then, at time $t = t_{s+1}$ the automaton C_i transitions to mode $v_{C_i}^{s+1} = S_{z_{k+1}}$ and $V_i^d(t_s) \leftrightarrow Q(t_s)$ which implies that $\big(A_i(t_{s+1}) \wedge Q(t_s)\big) \leftrightarrow \mathbf{0}_{n\times n}$, since the edge $w(t_s)$ is deleted from $A_i(t_s)$, by Proposition 5.7. Hence, $A_i(t_{s+1}) \rightarrow A(t_{s+1})$ by Lemma 1.5(i) in the Appendix, which completes the proof. ■

Proposition 5.10 implies that $\mathcal{E}_i(t_s) \subseteq \mathcal{E}(t_s)$ for any automaton C_i and any time $t_s \in [t_{B_k}, t_{B_{k+1}})$ such that $v_{C_i}^s = S_{z_k}$. In other words, when agent i is about to select a link $i \sim j$ to delete from $\mathcal{E}_i(t_s)$, $\mathcal{G}_i(t_s)$ is a spanning subgraph of $\mathcal{G}(t_s)$. Hence, we can state our main result.

Theorem 5.11: If $\mathcal{G}(t_0)$ is connected, then the product system S is guaranteed to maintain connectivity of the mobile network $\mathcal{G}(t)$ for all time $t \geq t_0$.

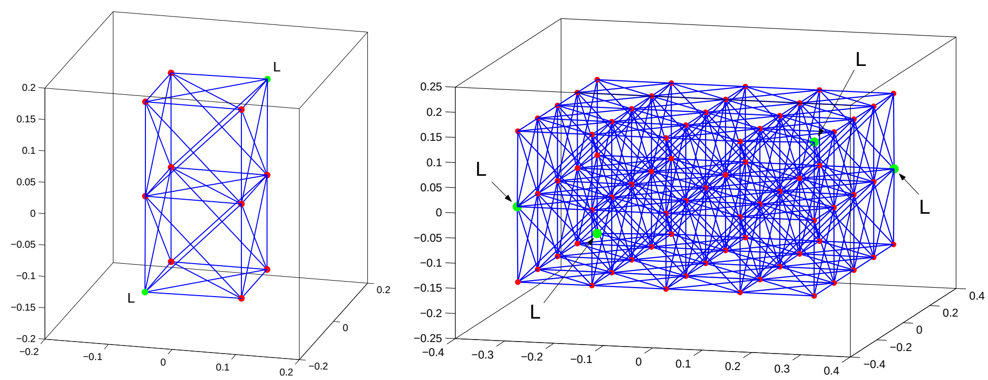
Proof: By Theorem 3.1, between transitions of the product system S, where the neighbor sets $\mathcal{N}_i$ are fixed for all automata C_i, the network is guaranteed to remain connected. Hence, we only need to show that connectivity is not violated at the transition time instants of the system S, which are due to creation or deletion of links in the network $\mathcal{G}(t)$. However, since addition of links does not endanger connectivity of the network, we only need to show connectivity is not violated when a link is deleted from $\mathcal{G}(t)$.

Note first that, by Propositions 5.4 and 5.6, the proposed market-based coordination framework in the product system S is sound, which implies that the outcome of every auction is at most one link which is common for all automata C_i. Moreover, Proposition 5.7 implies that this link is always deleted from the edge sets $\mathcal{E}_i(t)$. Hence, we need to show that the selection of this link is safe with respect to connectivity.

Consider any automaton C_i such that $v_{C_i}^s = S_{z_k}$ for $t = t_s$. Then, at $t = t_{s+1}$ the transition $v_{C_i}^s \rightarrow v_{C_i}^{s+1}$ results in $v_{C_i}^{s+1} = B_{z_k}$ and the corresponding reset becomes $\mathcal{S}_i(t_{s+1}) := reset((v_{C_i}^s, v_{C_i}^{s+1})) = \{j \in$

$\mathcal{N}_i(t_s) \backslash \mathcal{N}_i^{\epsilon_1}(t_s) \mid \lambda_2(L(\mathcal{E}_i(t_s) \backslash \{i \sim j\})) > k-1\}$.[18] But, $\mathcal{E}_i(t_s) \subseteq \mathcal{E}(t_s)$, by Proposition 5.10, and updating of the edge set $\mathcal{E}_i(t)$ is sound. Hence, if $\mathcal{S}_i(t_{s+1}) \neq \emptyset$ and any of the links $i \sim j$ with $j \in \mathcal{S}_i(t_{s+1})$ is deleted, the graph $\mathcal{G}_i(t_{s+1}) = (\mathcal{V}, \mathcal{E}_i(t_{s+1}))$ is still connected and hence, so is the overall graph $\mathcal{G}(t_{s+1}) = (\mathcal{V}, \mathcal{E}(t_{s+1}))$. ■

VI. CONNECTIVITY TASKS



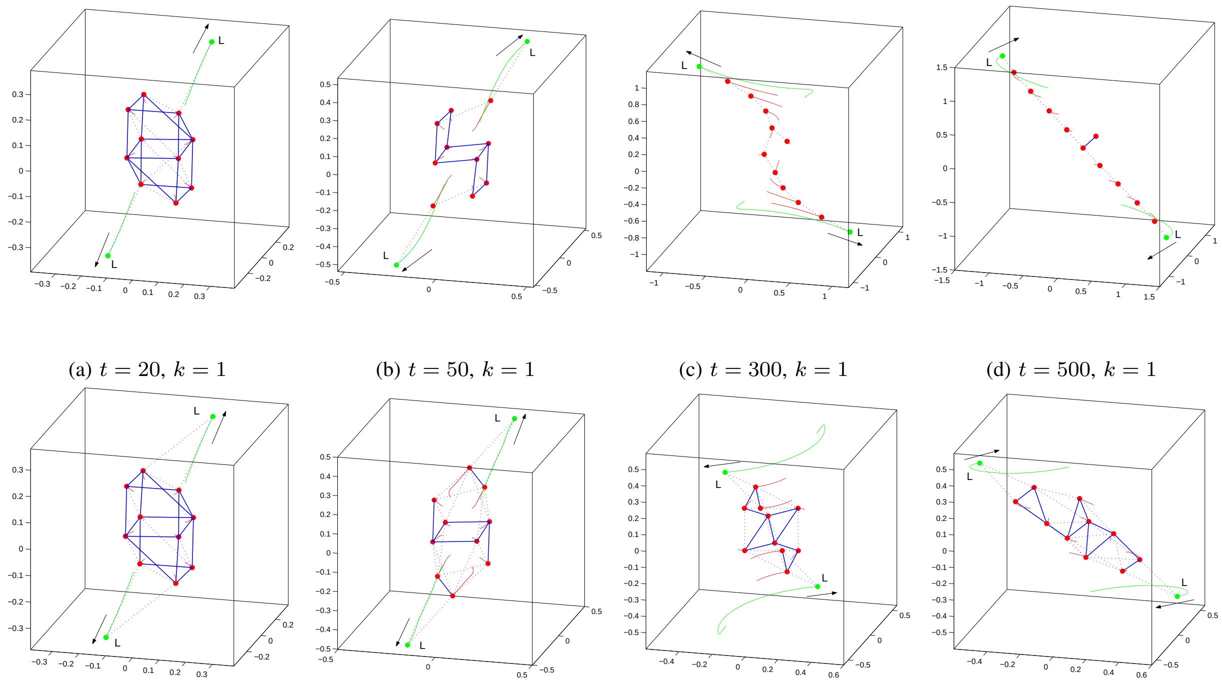
(a) $n = 12$ agents with 2 leaders. (b) $n = 75$ agents with 4 leaders.

Fig. 9. Initial Configurations.

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 $\mathcal{L} \subseteq \{1, \dots, n\}$ of *leaders* having a non-trivial secondary objective and a set of *followers* $\{1, \dots, n\} \backslash \mathcal{L}$ having no secondary objective. In particular, for all leaders $i \in \mathcal{L}$ we assume secondary objectives as in Figure 1(b), where we also add unit angular velocities, slightly abusing the secondary objective specifications in equation (1). 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.

In our first connectivity scenario, we consider $n = 12$ agents and compare k-connectivity control for $k = 1, 2$ and for the same initial configuration of the agents (Figure 9(a)). Our scenario involves 2 leaders with secondary objective parameters chosen to *stretch* the network and see whether it is able to reconfigure while maintaining connectivity. The link ranges in Definition 2.1 are $r = .25$ and $R = .4$. Figure 10 compares 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. Note also that in the absence of strong links in the network our algorithm generates a minimally connected network, i.e., a tree structure.

[18] See Remark 2.4 for a definition of k-connectivity.



(a) $t = 20$, $k = 1$ (b) $t = 50$, $k = 1$ (c) $t = 300$, $k = 1$ (d) $t = 500$, $k = 1$

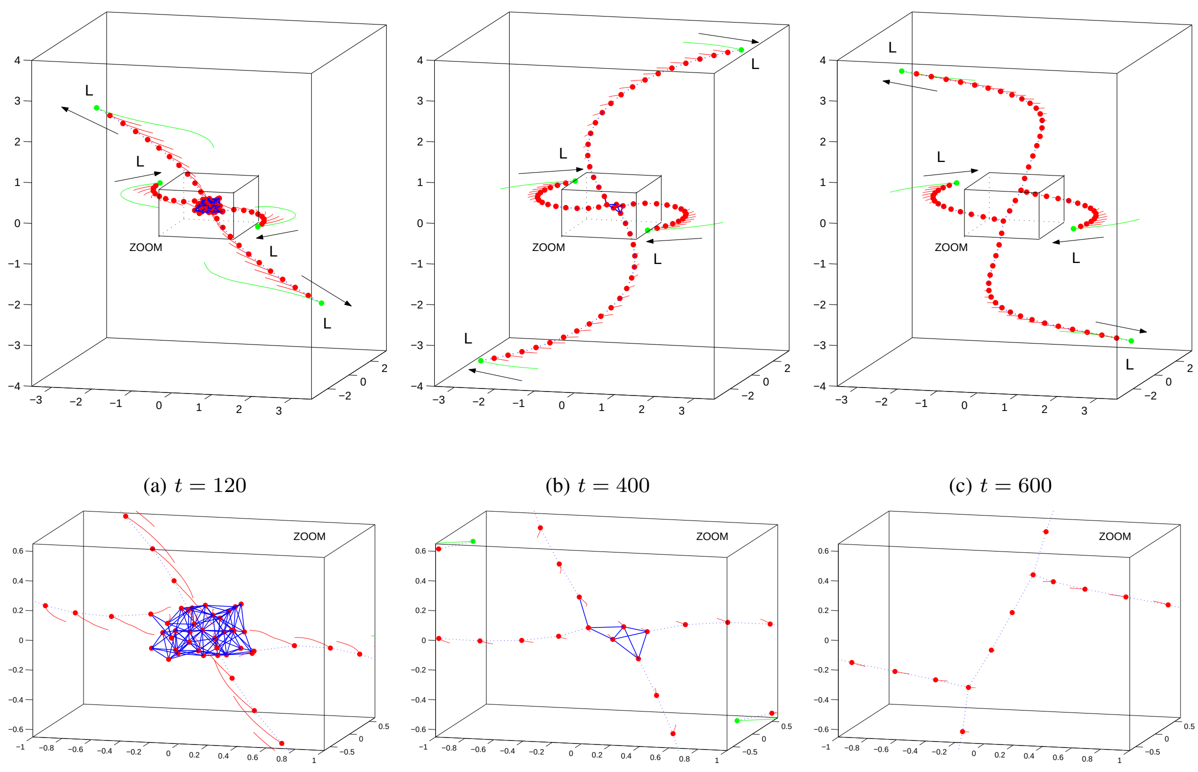
(e) $t = 20$, $k = 2$ (f) $t = 50$, $k = 2$ (g) $t = 300$, $k = 2$ (h) $t = 500$, $k = 2$

Fig. 10. k-Connectivity control for $n = 12$ agents with 2 leaders. Compare figures (a,e), (b,f), (c,g) and (d,h).

To illustrate scalability of our approach, we consider a second connectivity scenario involving a group of $n = 75$ agents with 4 leaders initialized as in Figure 9(b). The link ranges are $r = .25$ and $R = .4$, and the secondary objectives for the leaders are according to the potential in Figure 1(b) with the addition of a unit angular velocity, as before. Figure 11 shows the evolution of the system for three consecutive time instants. Again, we see that the overall network remains connected while reconfiguring according to the leaders' motion.

VII. 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 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.



(a) $t = 120$ (b) $t = 400$ (c) $t = 600$

(d) $t = 120$ (Zoom) (e) $t = 400$ (Zoom) (f) $t = 600$ (Zoom)

Fig. 11. Connectivity control for $n = 75$ agents with 4 leaders.

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, April 2005. To Appear.

[12] 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.

[13] R. Sepulchre, D. Paley, N. E. Leonard. *Stabilization of Planar Collective Motion: All-to-All Communication*, IEEE Transactions on Automatic Control, 2006. To Appear.

[14] 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.

[15] T. Balch and R.C. Arkin. *Behavior-based Formation Control for Multirobot Teams*, IEEE Transactions on Robotics and Automation, vol. 14(6), pp. 926-939, Dec. 1998.

[16] J. Lin, A. S. Morse and B. D. O. Anderson. *The Multi-Agent Rendezvous Problem*, Proceedings of the 42nd IEEE Conference on Decision and Control, pp. 1508-1513, Maui, Hawaii, Dec. 2003.

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

[18] 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.

[19] M. M. Zavlanos and G. J. Pappas. *Dynamic Assignmnet in Distributed Motion Planning with Local Coordination*, IEEE Transactions on Robotics, November 2006. Submitted.

[20] 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.

[21] M. M. Zavlanos and G. J. Pappas. *Potential Fields for Maintaining Connectivity of Mobile Networks*, IEEE Transactions on Robotics, May 2006, Accepted.

[22] C. Godsil and G. Royle. *Algebraic Graph Theory*, Springer-Verlag, New York, 2001.

[23] T. A. Henzinger. *The Theory of Hybrid Automata*, Proceedings of the 11th Annual Symposium on Logic in Computer Science (LICS), pp. 278-292, IEEE Computer Society Press, 1996.

[24] N. A. Lynch, R. Segala and F. W. Vaandrager. *Hybrid I/O Automata*, Information and Computation, vol. 185(1), pp. 105-157, Aug. 2003.

APPENDIX I
BOOLEAN OPERATIONS

In this section we define some basic boolean operations we use throughout this paper and state some of their properties. In particular, we have the following definition,

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 III, where the symbols $\neg$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$ stand for *not*, *and*, *or*, *if, then* and *if and only if*, respectively.

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. Using combinations of boolean operations, as in Definition 1.2, we can generate

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

any boolean expression,

$$Q : \{0,1\}^{n\times n} \times \cdots \times \{0,1\}^{n\times n} \rightarrow \{0,1\}^{n\times n}$$

on a set of boolean matrices $X_1, \ldots, X_k \in \{0,1\}^{n\times n}$. Let $Q = (q_{ij})$ be any such boolean expression and define the function $\sigma : \{0,1\}^{n\times n} \rightarrow \{0,1\}$ such that,

$$\sigma(Q) = \wedge_{i,j} q_{ij}$$

Clearly, $\sigma(Q) = 1$ if and only if $q_{ij} = 1$ for all $i, j = 1, \ldots, n$. Thus, we can define a *tautology* as follows,

Definition 1.3 (Tautology): We say that a boolean matrix expression Q is a *tautology* if $\sigma(Q) = 1$ for all $Q \in \{0,1\}^{n\times n}$.

The following lemma summarizes some useful tautologies of boolean operations on matrices.

Lemma 1.4 (Tautologies): Let $X, Y, Z \in \{0,1\}^{n\times n}$ be boolean matrices. We, then, have the following tautologies,

(a) $\big(X \wedge (Y \vee Z)\big) \leftrightarrow \big((X \wedge Y) \vee (Y \wedge Z)\big)$

(b) $\big(X \vee (Y \wedge Z)\big) \leftrightarrow \big((X \vee Y) \wedge (Y \vee Z)\big)$

(c) $\big(\neg(X \vee Y)\big) \leftrightarrow \big((\neg X) \wedge (\neg Y)\big)$

(d) $\big(\neg(X \wedge Y)\big) \leftrightarrow \big((\neg X) \vee (\neg Y)\big)$

(e) $\big(X \vee (X \wedge Y)\big) \leftrightarrow X$

(f) $\big((X \wedge Y) \leftrightarrow X\big) \leftrightarrow (X \rightarrow Y)$

(g) $\big(\neg(X \leftrightarrow Y)\big) \leftrightarrow \big(\big(\neg(X \wedge Y)\big) \wedge (X \vee Y)\big)$

(h) $(X \wedge Y) \rightarrow X$

(i) $\big(\big(X \rightarrow (Y \vee Z)\big) \wedge \big(\neg(X \wedge Z)\big)\big) \rightarrow (X \rightarrow Y)$

Proof: All tautologies can be easily verified by constructing the corresponding 0-1 table. ■

Note that tautologies (a) and (b) are the well known *Distributive laws*, while tautologies (c) and (d) consist *De Morgan's laws*. Using Lemma 1.4 we can show some useful properties of the boolean operators. In particular, we have the following lemma.

Lemma 1.5 (Operator Properties): Let $X_i, Y_i \in \{0,1\}^{n\times n}$ for $i = 1, \ldots, k$ and assume that $X_i \rightarrow Y_i$ for all i. Then,

(a) $(\wedge_i X_i) \rightarrow (\wedge_i Y_i)$

(b) $(\vee_i X_i) \rightarrow (\vee_i Y_i)$

Proof: Observe that $\big((\wedge_i X_i) \wedge (\wedge_j Y_j)\big) \leftrightarrow \big(\wedge_i (X_i \wedge Y_i)\big) \leftrightarrow (\wedge_i X_i)$ and part (a) follows by Lemma 1.4(f). To show part (b), apply the distributive laws and Lemma 1.4(e) to the expression,

$$\begin{aligned}\big((\vee_i X_i) \wedge (\vee_j Y_j)\big) &\leftrightarrow \big(\vee_i \big(X_i \wedge (\vee_j Y_j)\big)\big) \leftrightarrow \big(\vee_i \big((X_i \wedge Y_i) \vee \big(\vee_{j \neq i} (X_i \wedge Y_j)\big)\big)\big) \\ &\leftrightarrow \big(\vee_i \big(X_i \vee \big(\vee_{j \neq i} (X_i \wedge Y_j)\big)\big)\big) \leftrightarrow (\vee_i X_i)\end{aligned}$$

and the result follows by by Lemma 1.4(f). ■