Hybrid Control for Connectivity Preserving Flocking

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

***Abstract*— In this note, we address the combined problem of motion and network topology control in a group of mobile agents with common objective the flocking behavior of the group. Instead of assuming network connectivity, we enforce it by means of distributed topology control that decides on both deletion and creation of agent links, adapting the network to the group's spatial distribution. With this protocol ensuring network connectivity, a decentralized motion controller aligns agent velocity vectors and regulates inter-agent distances to maintain existing network links. The stability of the flocking controller is established in continuous time, by exploiting the time delay between link deletion and creation caused by the topology control protocol, which induces a dwell time between network switches.**

***Index Terms*— Multi-Agent Systems, Cooperative Control, Hybrid Systems, Algebraic Graph Theory.**

I. Introduction

Existing results on distributed consensus and flocking algorithms critically rely on maintaining a connected communication network among the agents, either for all time (as in [1], [2], [3]) or over sequences of bounded time intervals (as in [4], [5], [6]). In this paper, we relax this assumption and propose a distributed control framework that guarantees velocity alignment, cohesion, separation, *and* connectivity of the networked multi-agent system, by construction.

Inspired by the flocking and schooling phenomena observed in nature are many recent applications in control theory and robotics. Any attempt to list related references in this note is bound to be partial and incomplete, hence, we rather focus on work that emphasizes on the *connectivity* aspect of networked dynamical systems. In [7], network connectivity is maintained by means of potential fields that guarantee positive definiteness of the second smallest eigenvalue of the graph Laplacian matrix, while in [8] a measure of local connectedness of a network is introduced that under certain conditions is sufficient for global connectedness. Distributed maintenance of nearest neighbor links by means of unbounded "edge tension" functions is addressed in [9], where a control hysteresis is also introduced to avoid infinite control inputs when new links are about to be inserted to the network. Similarly, in [10] a system of interconnected unicycles is steered to a common configuration by means of nonsmooth, potential-based control inputs that turn unbounded when the distance between adjacent agents approaches a certain threshold. Invariance of the level sets of an appropriate function ensures that initially established links will be preserved along the system's trajectories.

In this paper, we address the problem of velocity synchronization in a network of n interconnected agents, while maintaining connectivity of the underlying proximity-based graph and ensuring collision avoidance among the agents. Unlike previous approaches, our proposed framework allows switching among connected network topologies that are due to both addition, as well as *deletion* of communication links between agents. As in [3] the dynamics of an agent are expressed by a double integrator,

$$\dot{x}_i(t) = v_i(t) \tag{1a}$$

$$\dot{v}_i(t) = u_i(t) \tag{1b}$$

where $x_i(t), v_i(t) \in \mathbb{R}^m$ denote the position and velocity vectors of agent i at time t, respectively, and $u_i(t) \in \mathbb{R}^m$ is a sought control vector that depends on the state variables of the agent's nearest neighbors only. Updating of the required neighbor set is then due to distributed inter-agent coordination, which integrated with the continuous agents' motion results in a multi-agent hybrid system. Under the assumption of an initially connected communication network, the overall system is shown to flock for all initial conditions.

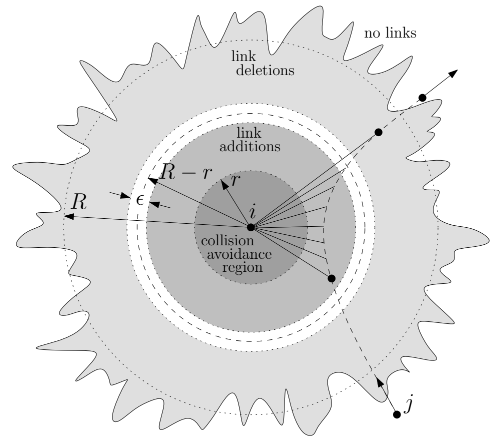


Fig. 1. Partitioning of the neighborhood of an agent according to its ability to communicate and the requirement to avoid collisions. Regions where communication links are established or lost are qualitatively associated to signal strength, ranging from strong to weak and no signal at all. The inner-most disk marks the area around robot i where collision avoidance maneuvers are initiated. Note a region of width $\epsilon > 0$ between the areas where links can be added or deleted, which introduces a dwell time $\tau > 0$ between any changes in the network topology. The dashed line shows the path of agent j, and the solid lines indicate a link between agents i and j.

II. Problem Formulation

Consider a network of n agents in $\mathbb{R}^m$ with integrated wireless communication capabilities and denote by (i, j) a communication link between agents i and j. We assume that such links can be enabled and disabled in time due to agent mobility and, as in [11], we employ proximity graphs to represent the agents' communication network. Motivation for relating the agents' ability to establish communication links to the distance between them comes from the fact that radio signal strength attenuates with distance, while the probability of a successful transmission rapidly decreases beyond a certain threshold. To capture such radio signals, we propose a rather *qualitative* model for the communication network, where new communication links can be established only between agents whose distance becomes smaller than some threshold $R > 0$. Beyond that threshold, we assume there is considerable uncertainty over the agents' ability to successfully communicate, hence communication links are lost. On the other hand, to avoid collisions, agents are not supposed to get

The work of the first and third author is partially supported by ARO MURI SWARMS Grant W911NF-05-1-0219 and the NSF ITR Grant 0324977. The work of the second author is supported by NSF under Grant No. 0447898

Michael M. Zavlanos and George J. Pappas are with the GRASP Laboratory, Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA. {zavlanos,pappasg}@grasp.upenn.edu Herbert G. Tanner is with the Department of Mechanical Engineering, University of New Mexico, Albuquerque, NM 87131, USA. tanner@unm.edu

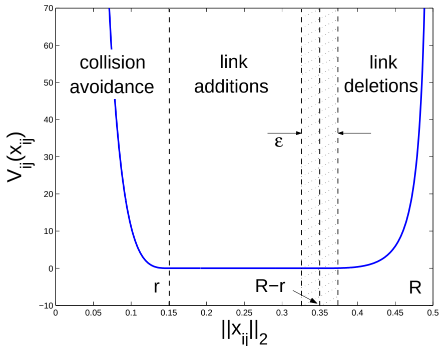


Fig. 2. The artificial potential function $V_{ij}(x_{ij})$. The function is symmetric with respect to x_i and x_j, and when bounded, it guarantees both collision avoidance for $\|x_{ij}\|_2 \to 0$ and edge preservation for $\|x_{ij}\|_2 \to R$. Here, the function is plotted for $r = 0.15$, $R = 0.5$, and $\epsilon = 0.05$.

too close to each other. Once their distance falls below a threshold $r > 0$, collision avoidance maneuvers must be initiated. The proposed dynamic network is illustrated in Fig. 1 and can be formally captured by the notion of a *dynamic proximity graph* $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, where $\mathcal{V} = \{1, \ldots, n\}$ denotes the set of vertices indexed by the set of agents and $\mathcal{E}(t) = \{(i,j) \mid i,j \in \mathcal{V}\}$ denotes the time varying set of links, such that for constants r, R, and ϵ satisfying $0 < 2r < R$ and $0 < \epsilon/2 < \min\{R - 2r, r\}$, we have

- if $\|x_i(t) - x_j(t)\|_2 \in [0, r)$ then, $(i,j) \in \mathcal{E}(t)$;
- if $(i,j) \notin \mathcal{E}(t)$ and $\|x_i(t) - x_j(t)\|_2 \in [r, R - r - \epsilon/2)$ then, (i,j) is a candidate link to be *added* to $\mathcal{E}(t)$;
- if $\|x_i(t) - x_j(t)\|_2 \in [R - r - \epsilon/2, R - r + \epsilon/2)$ then (i,j) preserves its membership status in $\mathcal{E}(t)$ (no addition or deletion);
- if $(i,j) \in \mathcal{E}(t)$ and $\|x_i(t) - x_j(t)\|_2 \in [R - r + \epsilon/2, R)$ then, (i,j) is a candidate link to be *deleted* from $\mathcal{E}(t)$;
- if $\|x_i(t) - x_j(t)\|_2 \in [R, \infty)$ then, $(i,j) \notin \mathcal{E}(t)$.

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. If, additionally, $\mathcal{G}(t)$ is such that there exists a path, i.e., a sequence of distinct vertices such that consecutive vertices are adjacent, between any two of its vertices, then we say that $\mathcal{G}(t)$ is *connected*. 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 $\mathcal{N}_i(t) = \{j \in \mathcal{V} \mid (i,j) \in \mathcal{E}(t)\}$.

Given any dynamic proximity graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ consisting of n dynamic agents described by Eqn. 1, define the set of control laws

$$u_i = -\sum_{j \in \mathcal{N}_i(t)} (v_i - v_j) - \sum_{j \in \mathcal{N}_i(t)} \nabla_{x_i} V_{ij} \tag{2}$$

where V_{ij} is the artificial potential function (Fig. 2),

$$V_{ij}(x_{ij}) \triangleq \begin{cases} \|x_{ij}\|_2^{-2} + P_1(x_{ij}), & \|x_{ij}\|_2 \in (0, r] \\ 0, & \|x_{ij}\|_2 \in (r, R - r) \\ \frac{1}{R^2 - \|x_{ij}\|_2^2} + P_2(x_{ij}), & \|x_{ij}\|_2 \in [R - r, R) \end{cases} \tag{3}$$

where $x_{ij} \triangleq x_i - x_j$ and $P_k(x_{ij}) \triangleq a_k\|x_{ij}\|_2^2 + b_k\|x_{ij}\|_2 + c_k$ with $k = 1, 2$, for appropriate constants a_k, b_k and c_k so that the derivatives of V_{ij} up to second order are continuous in $(0, R)$, i.e., for $\|x_{ij}\|_2 \in [r, R - r]$ they satisfy $V_{ij}(x_{ij}) = \frac{\partial V_{ij}}{\partial \|x_{ij}\|_2} = \frac{\partial^2 V_{ij}}{\partial \|x_{ij}\|_2^2} = 0$. Hence, the problem addressed in this paper can be stated as follows.

Problem 1 (Connectivity Preserving Flocking): Given an initially connected network $\mathcal{G}(t_0)$ of n dynamic agents described by (1)–(2), determine local controllers that regulate the neighbor sets $\mathcal{N}_i(t)$ so that the overall network $\mathcal{G}(t)$ is connected for all time, all agent velocities are aligned and collisions among the agents are avoided.

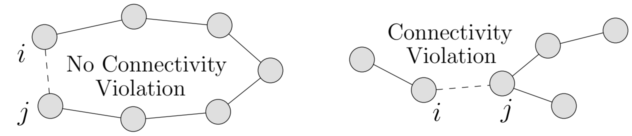


Fig. 3. Control challenges requiring knowledge of the network structure. Without such knowledge, deletion of a link (i,j) can either violate connectivity (right) or not (left).

Our approach to Problem 1 requires that the network $\mathcal{G}(t)$ remains invariant 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*

$$L(t) = \Delta(t) - A(t), \tag{4}$$

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*.[1] The spectral properties of the laplacian matrix are closely related to graph connectivity. In particular, we have the following lemma.

Lemma 2.1 ([12]): 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.

III. Distributed Connectivity Control

Consider a dynamic graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ defined by the time varying set of edges $\mathcal{E}(t)$. 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).

A. Local Estimates of the Network Topology

Let $\mathcal{G}_i(t) = (\mathcal{V}, \mathcal{E}_i(t))$ denote a local *estimate* of the global network $\mathcal{G}(t)$ that agent i can obtain using information from its

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

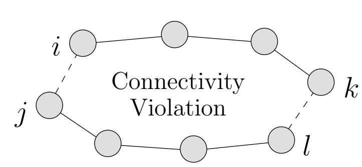


Fig. 4. Control challenges due to multiple link deletions. In the absence of an agreement protocol, simultaneous deletion of links (i,j) and (k,l) violates connectivity.

TABLE I

$a^{[i]}_{jk}(t)$	$v^{[i]}_{jk}(t)$	$a^{[i]}_{jk}(t+1)$
1	1	0
1	0	1
0	1	1
0	0	0

nearest neighbors $\mathcal{N}_i(t) = \{j \in \mathcal{V} \mid (i,j) \in \mathcal{E}(t)\}$ only. Let also $A_i(t) = \big(a^{[i]}_{jk}(t)\big)$ denote the adjacency matrix associated with the graph $\mathcal{G}_i(t)$ and define its dynamics by[2]

$$A_i(t+1) = \neg\big(A_i(t) \leftrightarrow V_i(t)\big)\big), \quad (5)$$

where the control input $V_i(t) = \big(v^{[i]}_{jk}(t)\big) \in \{0,1\}^{n\times n}$ is such that $v^{[i]}_{jk}(t) = 1$ if a control action is taken to add or delete link (j,k) (Table I). It can be shown that the control input $V_i(t)$ can be decomposed into two disjoint components $V_i^a(t)$ and $V_i^d(t)$ regulating link *additions* and *deletions*, respectively [13]. Furthermore, the local network dynamics (5) are essentially a *consensus* (with inputs) on the adjacency matrix estimates $A_i(t)$. In particular, it can be shown that if the estimates $A_i(t)$ are initialized with nearest neighbors and if no changes in the network topology are assumed (no inputs), the dynamics (5) become

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

providing every agent with a *rough* picture of the overall network, as desired [13]. Clearly, if all network estimates $\mathcal{G}_i(t)$ are spanning subgraphs of the overall network $\mathcal{G}(t)$, i.e., if $\mathcal{E}_i(t) \subseteq \mathcal{E}(t)$ for all agents i, then connectivity of $\mathcal{G}_i(t)$ for all i implies connectivity of $\mathcal{G}(t)$. In other words, the condition $\mathcal{E}_i(t) \subseteq \mathcal{E}(t)$ is a critical requirement for correctness of the proposed approach, and as shown in [13], this is guaranteed by the proposed dynamics (5).

B. Controlling Addition and Deletion of Links

Regarding the component $V_i^a(t) = (v^{[i]a}_{jk}(t))$ that regulates link additions, we require that it is such that $A_i(t)$ is updated with all existing links in the network that are provided by agent i's neighbors $\mathcal{N}_i(t)$ and that it also captures new links that agent i can create with agents $j \notin \mathcal{N}_i$, i.e.,

$$v^{[i]a}_{jk}(t) \triangleq \underbrace{\big((j\neq i)\wedge(k\neq i)\big)}_{\text{add all existing links provided by neighbors}} \vee \underbrace{\big(\|x_{jk}(t)\|_2 \in [r, R-r-\epsilon/2)\big)}_{\text{maintain current neighbors and add new neighbors}}, \quad (6)$$

Unlike link additions, deletion of nearest neighbor links is a 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 and employ *market-based* control to achieve agreement of all agents regarding the link that is to be deleted. The proposed market-based control framework consists of a sequence of auctions as in Alg. 1, each one of which results in at most one link $w_i(t)$ (corresponding to the highest bid) that is deleted from the network. The control input that regulates link deletions $V_i^d(t) = \big(v^{[i]d}_{jk}(t)\big)$ can be defined as

$$v^{[i]d}_{jk}(t) \triangleq \big(w_i(t) = (j,k)\big) \wedge (|w_i(t)| = 1). \quad (7)$$

[2]The discrete time semantics in Eqn. 5 are associated with receipt of messages from neighboring agents (Section V).

Algorithm 1 Auction mechanism for agent i

1: Compute the set of *safe* links (i,j) with $j \in \mathcal{N}_i(t)\backslash\{k \mid \|x_{ik}(t)\|_2 \in [R-r+\epsilon/2, R)\}$ that if deleted from $\mathcal{E}_i(t)$, then $\mathcal{G}_i(t)$ remains connected (see Fig. 1 and Lemma 2.1).
2: Select a single safe link (i,j) and initialize a request $r_i \triangleq [\ i\ j\ b\ ]^T \in \mathbb{R}^3$ consisting of that link 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** $(\wedge_{j=1}^n T_{ij}(t)) = 0$ **do**
5: Collect tokens from neighbors only, i.e., $T_i(t+1) := T_i(t) \vee \big(\vee_{j\in\mathcal{N}_i(t)} T_j(t)\big)$
6: Apply a max-consensus update on $\mathcal{M}_i(t)$, i.e., $\mathcal{M}_i(t+1) := \big\{r_j \mid j = \underset{r_k\in\cup_{l\in\{\mathcal{N}_i(t),i\}}\mathcal{M}_l(t)}{\operatorname{argmax}} \{r_{k3}\}\big\}$
7: **end while**
8: Compute the *winner* link $w_i(t)$ of the auction, $w_i(t) \triangleq \{(r_{j1}, r_{j2}) \mid r_j \in \mathcal{M}_i(t)\}$

Note that $|w_i(t)| > 1$ implies either a tie in the maximum bids, in which case, Eqn. 7 results in $V_i^d(t) = \mathbf{0}$ for all agents i so that no link is deleted from any network estimate $\mathcal{E}_i(t)$ (cf. Eqn.5).

Remark 3.1 (Choosing the Bids): Note that any positive real numbers can serve as bids in Alg. 1. However, letting $b \geq 0$ be a function of the distance $\|x_i(t)-x_j(t)\|_2$, $j \in \mathcal{S}_i(t)$ 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 3.2 (Convergence of max-Consensus): The condition $(\wedge_{j=1}^n T_{ij}(t)) = 1$ marking the end of the *while* loop in Alg. 1, implies convergence of the max-consensus algorithm on the sets $\mathcal{M}_i(t)$ to the maximum over all agents. By using tokens $T_i(t)$ to indicate the end of an auction, we employ the structure of the network, and achieve more efficient updating.[3] Additionally, this approach allows convergence of all agents to a common auction outcome, even in the presence of communication time delays in the system [13]. Memory and communication costs for transmitting these binary tokens is minimal.

Remark 3.3 (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 [14]. This complexity can, however, be reduced to $O(n)$ for sparse symmetric matrices [15], 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. Integration with Agent Mobility: Velocity Alignment & Collision Avoidance

With the topology control component of Section III adding and deleting edges at will, the agent control law (2) experiences discontinuities, and induces a switching nonlinear closed loop dynamical system (1)–(2). Let t_p for $p = 1, 2, \dots$ denote the *switching times* when the topology of $\mathcal{G}(t)$ changes, and define a switching signal $\mathcal{G}(t) : [t_0, \infty) \to \mathcal{G}_{\mathcal{C}}$, where $\mathcal{G}_{\mathcal{C}}$ denotes the set of all connected graphs on n vertices.[4] As discussed in Section V, communication

[3]For complete graphs, for instance, one iteration is sufficient for convergence of the max-consensus algorithm, significantly reducing the $O(n)$ convergence rate of this update rule.

[4]Note that $\mathcal{G}(t)$ is also a map from the real time-line to the set of graphs.

time delays introduce a dwell time $\tau > 0$ between transitions in the network topology $\mathcal{G}(t)$. Furthermore, the topology controller described in Section III guarantees that the sequence of proximity graphs consists of connected graphs. Using these results, we now state our main result.

Theorem 4.1 (Connectivity Preserving Flocking): For the closed loop system (1)–(2), assume that $\mathcal{G}(t_0) \in \mathcal{G}_{\mathcal{C}}$ and $t_p - t_{p-1} > \tau > 0$ for all switching times $t_p > 0$. Then, $\mathcal{G}(t) \in \mathcal{G}_{\mathcal{C}}$ for all $t > 0$, and $\|x_{ij}(t)\|_2 > 0$ for all $i, j \in \mathcal{V}$ and all $t > 0$. Moreover, $v_i \to v_j$ as $t \to \infty$, for all $i, j \in \mathcal{V}$.

Proof: Let $t_{p_1}, t_{p_2}, \dots$ denote an infinite subsequence of switching times such that the switching signal $\mathcal{G}(t)$ in each of the intervals $[t_{p_q}, t_{p_{q+1}})$ for $q = 1, 2, \dots$ is the same. Denote the union of these intervals by $\mathcal{Q}$ and for all time $t \in \mathcal{Q}$, let $\hat{\mathbf{x}} \triangleq [\dots x_{ij}^T \dots]^T \in \mathbb{R}^{mn(n-1)}$, $\mathbf{v} \triangleq [\dots v_i^T \dots]^T \in \mathbb{R}^{mn}$ and $\mathbf{u} \triangleq [\dots u_i^T \dots]^T \in \mathbb{R}^{mn}$ denote the stack vectors of the robot relative positions $x_{ij} \in \mathbb{R}^m$, velocity vectors $v_i \in \mathbb{R}^m$ and control signals vectors $u_i \in \mathbb{R}^m$, respectively. Consider the dynamical system

$$\dot{\hat{\mathbf{x}}} = (B_K \otimes I_m)\mathbf{v} \tag{8a}$$
$$\dot{\mathbf{v}} = \mathbf{u}, \tag{8b}$$

where B_K is the incidence matrix of the complete proximity graph [12], and define the function $V_{\mathcal{G}} : D_{\mathcal{G}} \times \mathbb{R}_+^{mn} \to \mathbb{R}_+$, where $D_{\mathcal{G}} = \{\mathbf{x} \in \mathbb{R}^{mn(n-1)} | \|x_{ij}\|_2 \in (0, R) \ \forall \ (i,j) \in \mathcal{E}\}$ as

$$V_{\mathcal{G}} = \frac{1}{2}\Big(\|\mathbf{v}\|_2^2 + \sum_{i=1}^{n}\sum_{j \in \mathcal{N}_i} V_{ij}\Big). \tag{9}$$

For any $c > 0$, let $\Omega_{\mathcal{G}} = \{(\mathbf{x}, \mathbf{v}) \in D_{\mathcal{G}} \times \mathbb{R}_+^{mn} | V_{\mathcal{G}} \leq c\}$ denote the level sets of $V_{\mathcal{G}}$ and observe that (cf. [3])

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j \in \mathcal{N}_i} \dot{V}_{ij} = \sum_{i=1}^{n}\sum_{j \in \mathcal{N}_i} \dot{x}_i^T \nabla_{x_i} V_{ij}. \tag{10}$$

Eqn. 10 and the Kronecker product notation ($\otimes$) simplifies the expression for $\dot{V}_{\mathcal{G}}$ to

$$\dot{V}_{\mathcal{G}} = -\mathbf{v}^T (L_{\mathcal{G}}(t) \otimes I_m)\mathbf{v}, \tag{11}$$

which is always nonpositive, since the laplacian $L_{\mathcal{G}}(t)$ is always positive semidefinite. Hence, for any signal $\mathcal{G}$, the level sets $\Omega_{\mathcal{G}}$ are positively invariant, implying that for any $(i,j) \in \mathcal{E}$, V_{ij} remains bounded. On the other hand, if for some $(i,j) \in \mathcal{E}$, $\|x_{ij}\|_2 \to R$, then $V_{ij}(x_{ij}) \to \infty$. Thus, by continuity of $V_{\mathcal{G}}$ in $D_{\mathcal{G}}$, it follows that $\|x_{ij}\|_2 < R$, for all $(i,j) \in \mathcal{E}$ and $t \in [t_{p_q}, t_{p_q+1})$. In other words, all links in $\mathcal{G}$ are maintained between switching times, and since $\mathcal{G}(t_{p_q}) \in \mathcal{G}_{\mathcal{C}}$ for $q = 1, 2, \dots$, we have that $\mathcal{G}(t) \in \mathcal{G}_{\mathcal{C}}$ for all $t \in [t_{p_q}, t_{p_q+1})$, hence, for all $t \in \mathcal{Q}$. A similar argument for the case where $\|x_{ij}\|_2 \to 0$ can be used to establish collision avoidance.

Since the addition of an edge between robots i and j can occur only in the region where $\|x_{ij}\|_2 \in [r, R - r - \epsilon/2)$, such an event temporarily has no effect on the value of $\sum_{i=1}^{n}\sum_{j \in \mathcal{N}_i} V_{ij}$ (Fig. 2). In conjunction with (11) we conclude that $V_{\mathcal{G}(t_{p_q})} \geq V_{\mathcal{G}(t_{p_q+1})}$, so if $V_{\mathcal{G}(t_{p_1})}$ is bounded, so is $V_{\mathcal{G}(t_{p_q})}$ for all $q = 1, 2, \dots$. Moreover, the level sets of $\Omega_{\mathcal{G}}$ are closed by continuity of $V_{\mathcal{G}}$ in $D_{\mathcal{G}} \times \mathbb{R}_+^{mn}$. Note now that

$$\begin{aligned} \Omega_{\mathcal{G}} \quad &\subseteq \quad \{\mathbf{v} \mid \|\mathbf{v}\|_2^2 \leq c\} \cap \big(\cap_{(i,j)\in\mathcal{E}} \{x_{ij} \mid V_{ij} \leq c\}\big) \\ &= \quad \{\mathbf{v} \mid \|\mathbf{v}\|_2^2 \leq c\} \cap \big(\cap_{(i,j)\in\mathcal{E}} V_{ij}^{-1}([0,c])\big) \triangleq \Omega. \end{aligned} \tag{12}$$

The velocity set $\{\mathbf{v} | \|\mathbf{v}\|_2^2 \leq c\}$ is closed and bounded and hence, compact. Moreover, 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 $V_{ij}^{-1}([0,c])$

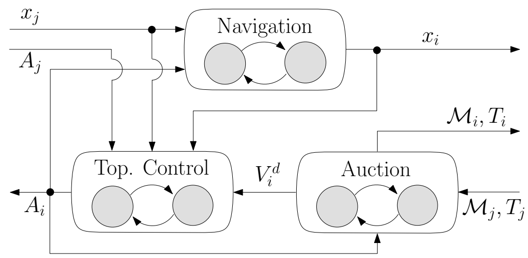


Fig. 5. The hybrid automaton $\mathbb{T}_i \times \mathbb{A}_i \times \mathbb{N}_i$ of a mobile robot i that consists of the composition of a topology control $\mathbb{T}_i$, an auction $\mathbb{A}_i$ and a navigation automaton $\mathbb{N}_i$.

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 $N \to R$, and given that $\lim_{\|x_{ij}\|_2 \to R} V_{ij} = \infty$, 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 $V_{ij}^{-1}([0,c])$ are bounded and hence, compact. Therefore, the set Ω is compact as an intersection of finite compact sets. It follows that $\Omega_{\mathcal{G}}$ is also compact, as a closed subset of a compact set.

So far we have shown that the level sets $\Omega_{\mathcal{G}}$ of $V_{\mathcal{G}}$ are both positively invariant and compact. The invariance of $\Omega_{\mathcal{G}}$ implies that no collisions between robots occur. We now use these results to show that all robot velocities become asymptotically the same. Note first that compactness and positive invariance of $\Omega_{\mathcal{G}}$ implies that $(\mathbf{x}, \mathbf{v}) \in D_{\mathcal{G}} \times \mathbb{R}_+^{mn}$ remains bounded for all $t \in \mathcal{Q}$. Moreover, since $V_{\mathcal{G}} \in C^2$ inside $D_{\mathcal{G}} \times \mathbb{R}_+^{mn}$, the right-hand-side of (8) is is locally Lipschitz, which implies that $(\dot{\mathbf{x}}, \dot{\mathbf{v}})$ is bounded. Hence, $(\mathbf{x}, \mathbf{v})$ is continuous in any bounded time interval $[t_{p_q}, t_{p_q+1})$, suggesting that the quantity $\mathbf{v}^T(L_{\mathcal{G}}(t) \otimes I_m)\mathbf{v}$ is also continuous and hence, uniformly continuous on $\mathcal{Q}$. Define the auxiliary function

$$y_{\mathcal{Q}}(t) \triangleq \begin{cases} \mathbf{v}^T(L_{\mathcal{G}}(t) \otimes I_m)\mathbf{v}, & t \in \mathcal{Q} \\ 0, & \text{otherwise} \end{cases}. \tag{13}$$

Since $V_{\mathcal{G}(t_p)} \leq V_{\mathcal{G}(t_{p-1})}$ for any consecutive signals $\mathcal{G}(t_{p-1}), \mathcal{G}(t_p) \in \mathcal{G}_{\mathcal{C}}$, characterized by simultaneous addition and deletion of links, and all switching times t_p, we have that

$$\int_0^t |y_{\mathcal{Q}}(s)| ds = \int_0^t y_{\mathcal{Q}}(s) ds \leq V_{\mathcal{G}(t_{p_1})} - V_{\mathcal{G}(t)} \leq V_{\mathcal{G}(t_{p_1})},$$

which suggests that $y_{\mathcal{Q}} \in \mathcal{L}_1$. We now proceed to showing that $y_{\mathcal{Q}}(t) \to 0$ as $t \to \infty$. Our argument is along the lines of [16]: suppose that this is not true. Then, there exists an $\varepsilon > 0$ and an infinite sequence of times $s_1, s_2, \dots$ such that the values $y_{\mathcal{Q}}(s_1), y_{\mathcal{Q}}(s_2), \dots$ are bounded away from zero by at least ε. It follows from (13) that the times $s_1, s_2, \dots$ necessarily belong to $\mathcal{Q}$. Since $y_{\mathcal{Q}}$ is uniformly continuous, we can find a $\delta > 0$ such that each s_i is contained in some interval of length δ, on which $y_{\mathcal{Q}}(t) \geq \varepsilon/2$. (Recall that the length of each such interval in $\mathcal{Q}$ is lower bounded by $\tau > 0$.) This contradicts the fact that $y_{\mathcal{Q}} \in \mathcal{L}_1$. Hence, $y_{\mathcal{Q}}(t) \to 0$ as $t \to \infty$, which suggests that $\mathbf{v} \to \mathbf{1} \otimes \zeta^T$ for some $\zeta \in \mathbb{R}^m$. This means that the corresponding components of robot velocities converge asymptotically to common values. ■

V. THE CLOSED LOOP HYBRID AGENT

The discrete topology controllers described in Section III along with the continuous motion controllers (1)–(2), give rise to a hybrid model for every agent i (Fig. 5), defined by the composition (or product) $\mathbb{T}_i \times \mathbb{A}_i \times \mathbb{N}_i$ of a *topology control* $\mathbb{T}_i$, an *auction* $\mathbb{A}_i$ and a *navigation* automaton $\mathbb{N}_i$, respectively [13]. The topology control automaton of robot 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 robot 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 robot i's neighbors (Alg. 1). To capture the continuous agent motion, the navigation automaton $\mathbb{N}_i$ 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. 1 and 2). The updated robot positions are then provided to the topology control automaton that further updates robot i's network estimate A_i and the resulting set of neighbors $\mathcal{N}_i$. Note that in the proposed hybrid system, all variables are considered *shared*, however, the only variables that are practically needed are the ones provided by every robot's neighbors, which guarantees the local nature of the proposed control framework.

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

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

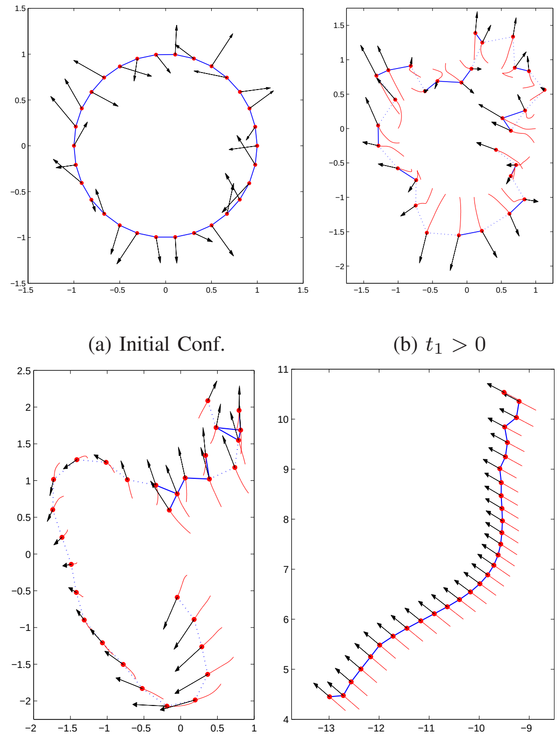
containing their network estimates A_i, max-bid sets $\mathcal{M}_i$ and tokens T_i.[5] Clearly, such messages are neither received simultaneously nor instantaneously. Instead, they are queued and are received with a time delay $\tau_i > 0$ and in an order that may vary according to the frequency of transmission of each robot.[6] To address these challenges, a notion of *synchronization* is required among the three individual automata modeling a single hybrid agent, as well as among all hybrid agents in the overall hybrid multi-agent 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 robot i's neighbors $j \in \mathcal{N}_i$. We achieve this synchronization by creating three identical copies of the auction protocol described in Alg. 1, that only differ on their labels. In other words, 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 $\{a, b, c, a, b, c, \dots\}$ and that nearest neighbor updates use information from equally labeled auctions [13].

VI. Simulation Results

We apply the proposed hybrid controller in coordination problems where connectivity of the network can not be trivially maintained. We consider $n = 30$ agents in $\mathbb{R}^2$, symmetrically distributed on the perimeter of a unit circle, with initial velocities chosen randomly within the unit square and parameters $r = 0.15$ and $R = 0.5$. The agents are denoted with dots, while the links between them are indicated by either solid or dashed lines, depending on whether the corresponding inter-agent distances are in the $[0, R-r)$ or $[R-r, R)$ region, respectively. Solid curves attached to every agent indicate the recently traveled paths, while arrows correspond to the agents' velocities (Fig. 6). Fig. 7(a)–7(c) show the evolutions with time (log-scale) of the Fiedler eigenvalue $\lambda_2(t)$, the auxiliary function $y_{\mathcal{Q}}(t)$, and the minimum distance $\min_{ij}\{\|x_{ij}(t)\|_2\}$ between agents. One can clearly see that the network always remains connected, all agent velocities are asymptotically aligned and collisions among agents are always avoided, as desired.

[5]Note that in order to define the continuous motion dynamics, we need to 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 discretizing the motion dynamics. Details can be found in [17], where the proposed algorithm is implemented on a real robotic platform.

[6]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 $\{1, 1, 2, 1, 3, 2, 2, 1, \dots\}$.



(a) Initial Conf. (b) $t_1 > 0$

(c) $t_2 > t_1$ (d) Steady State

Fig. 6. Decentralized flocking of 30 agents with topology control.

VII. Conclusion

In this note we offer a solution to the problem of producing flocking behavior in a group of mobile agents, without assuming network connectivity or forcing all initial links to be maintained over all time. By means of of a distributed topology control protocol, all possible network links are subject to creation or deletion, depending on the spatial distribution of the agents at any given time instant. The discrete network protocol ensures connectivity of the dynamic network, as well as a hysteresis between topology changes, properties which a continuous decentralized motion controller exploits to guarantee velocity synchronization and collision avoidance. The two controllers are combined into a hybrid architecture, where topology control facilitates motion coordination, and motion control preserves the topology dictated by the discrete network controller.

References

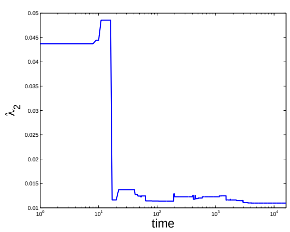
[1] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, September 2004.

[2] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, March 2006.

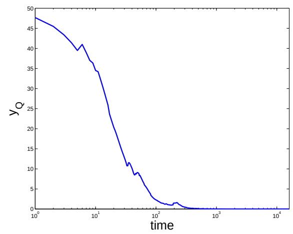
[3] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 863–867, 2007.

[4] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, July 2002.

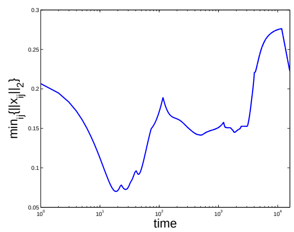
[5] W. Ren and R. Beard, "Consensus of information under dynamically changing interaction topologies," in *Proceedings of the American Control Conference*, June 2004, pp. 4939–4944.



(a) The Fiedler eigenvalue.



(b) The auxiliary function.



(c) Min. inter-agent dist.

Fig. 7. Decentralized flocking with topology control. Fig. 7(a) shows how the second smallest eigenvalue of the network's Laplacian varies in $\mathbb{R}_+$. The topology control protocol allows the network to change without being shattered. Fig. 7(b) verifies the prediction of the proof of Theorem 4.1 that the auxiliary function y_Q vanishes with time. Fig. 7(c) demonstrates the collision avoidance capabilities of (2).

[6] J. Lin, A. Morse, and B. Anderson, "The multi-agent rendezvous problem," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003, pp. 1508–1513.

[7] M. M. Zavlanos and G. J. Pappas, "Potential fields for maintaining connectivity of mobile networks," *IEEE Transactions on Robotics*, May 2006, (to appear).

[8] D. P. Spanos and R. M. Murray, "Robust connectivity of networked vehicles," in *Proceedings of 43rd IEEE Conference on Decision and Control*, 2004, pp. 2893–2898.

[9] M. Ji and M. Egerstedt, "Robust connectivity of networked vehicles," in *Proceedings of IEEE Conference on Decision and Control*, 2006, pp. 2893–2898.

[10] D. V. Dimarogonas and K. J. Kyriakopoulos, "Connectivity preserving state agreement for multiple unicycles," in *Proceedings of European Control Conference*, 2007, (to appear).

[11] 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, no. 8, pp. 1289–1298, 2006.

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

[13] M. M. Zavlanos and G. J. Pappas, "Distributed connectivity control of mobile networks," Department of Computer and Information Science, University of Pennsylvania, Tech. Rep. MS-CIS-07-08, March 2007.

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

[15] S.-T. Yau and Y. Y. Lu, "A new approach to sparse matrix eigenvalues," in *Proc. 1st IEEE Conference on Regional Aerospace Control Systems*, Westlake Village, CA, May 1993, pp. 132–137.

[16] J. P. Hespanha, D. Liberzon, D. Angeli, and E. D. Sontag, "Nonlinear norm-observability notions and stability of switched systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 154–168, 2005.

[17] N. Michael, M. M. Zavlanos, V. Kumar, and G. J. Pappas, "Maintaining connectivity in mobile robot networks," in *11th International Symposium on Experimental Robotics*, Athens, Greece, July 2008, (to appear).

APPENDIX I
BOOLEAN OPERATIONS

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

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

The operations definition can be extended to *boolean matrices*:

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.