

1

Maintaining Connectivity in Mobile Robot Networks

Nathan Michael, Michael M. Zavlanos, Vijay Kumar, and George J. Pappas*

GRASP Laboratory,
University of Pennsylvania,
Philadelphia, PA 19104, USA
`{nmichael, zavlanos, kumar, pappasg}@grasp.upenn.edu`

Summary. Control of robotic 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-robot systems, one great new challenge is the development of distributed motion algorithms that guarantee connectivity of the overall network. In this work we discuss the experimental validation of a distributed algorithm that preserves the connectivity of a team of robots. The algorithm requires limited local information and communication between agents to determine the addition or deletion of network links through distributed consensus and market based auctions. Non-trivial simulation and experimental results demonstrate the effectiveness of the algorithm as a means to guarantee connectivity in a team of robots.

1.1 Introduction

Wireless communication plays a vital role in distributed multi-robot algorithms. Shared information is necessary in applications such as surveillance and coverage, and in numerous algorithms in the areas of control and estimation. However, the requirement or assumption of consistent and stable wireless communication often fails due to environmental interference, fading, or robots drifting beyond the range of the wireless radios. For this reason, algorithms that address changing network topologies have recently received considerable attention. Typically these approaches rely on modeling the systems as graphs, where every robot is a node and edges correspond to communication links between pairs of robots defined according to a pre-specified communication model.

Although in most multi-robot applications the robots' primary task is detection of certain physical changes within their proximity, their communication capabilities enable them to share the individually collected data with their peers in order to achieve a global coordinated objective. Consequently, network connectivity becomes a critical requirement [1,2]. In this context, changes in radio signal strength can be used to

* This work was partially supported by NSF Grants IIS-0427313 and IIS-0427313, ARO Grants W911NF-05-1-0219 and W911NF-04-1-0148, and ONR Grant N00014-07-1-0829.

guide relative movements of robots to ensure connectivity [3]. While this approach is decentralized and no explicit communication is necessary to maintain connectivity, theoretical guarantees are difficult. In contrast, [4] suggests a distributed feedback control framework that imposes no restrictions on the network structure other than the desired connectivity specification. The proposed approach relies on local estimates of the network topology that provide every agent with a *rough* picture of the network structure, along with gossip algorithms and distributed market-based control that allow link deletions without violating connectivity. Although theoretical guarantees are provided, these guarantees are based on ideal models of robot dynamics and network performance. In this paper we verify the integrity and correctness of the asynchronous and parallel computation proposed in [4] as well as message passing with time delays through experimental analysis on a team of robots.

1.2 Related Work

Due to their frequent appearance in multi-agent systems, dynamic networks have already received considerable attention. In [5], a measure of local connectedness of a network is introduced that under certain conditions is sufficient for global connectedness. Distributed maintenance of nearest neighbor links in formation stabilization is addressed in [6], while in [7], a controllability framework for state-dependent dynamic graphs is developed. In [8], 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 [9].

In this work we leverage market-based auctions to determine communication link deletion while preserving connectivity is a decentralized manner. Similar methods of task consensus through distributed market-based auctions are surveyed in [10].

The effects of interference on wireless communication in complex environments are of pragmatic interest in experimental robotics. In [11], the authors propose a methodology for exploiting multi-path fading by controlling the robot according to radio signal strength. To overcome environment interference, the authors of [12] consider the problem of controlling a team of robots to ensure end-to-end communication. They propose two different metrics, point-to-point signal strength and data throughput, to monitor the network connectivity of the system. Even *ad-hoc* communication protocols pose difficult challenges during multi-robot experimentation, as shown by the authors of [13].

1.3 Problem Formulation

Consider a network of n agents in \mathbb{R}^p with integrated wireless communication capabilities and denote by $x_i(t) \in \mathbb{R}^p$ the position of agent i at time t . Denote, further, by (i, j) a communication link between agents i and j and assume that such links can be enabled and disabled in time due to agent mobility. This gives rise to the notion

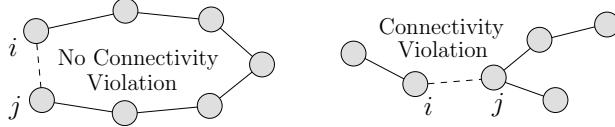


Fig. 1.1. Control challenges requiring knowledge of the network structure. Without such knowledge, deletion of a link (i, j) can either preserve (left) or violate (right) connectivity.

of a *dynamic graph* $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, where $\mathcal{V} = \{1, \dots, n\}$ is 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

- 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, $(i, j) \notin \mathcal{E}(t)$.

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 the main focus of this paper. Furthermore, assume $\mathcal{G}(t)$ 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 a dynamic network $\mathcal{G}(t)$ consisting of n fully actuated agents, assume that

$$\dot{x}_i(t) = -\nabla_{x_i} f_i(t) - \sum_{j \in \mathcal{N}_i(t)} \nabla_{x_i} V_{ij}(t), \quad (1.1)$$

where $f_i(t) \geq 0$ corresponds to a *secondary objective* (drift) modeled by a twice differentiable, radially unbounded potential and V_{ij} is the artificial potential function

$$V_{ij}(x_{ij}) \triangleq \frac{1}{\|x_{ij}\|_2^2} + \frac{1}{R - \|x_{ij}\|_2^2}, \quad (1.2)$$

used to maintain links and avoid collisions between adjacent agents, where $x_{ij} \triangleq x_i - x_j$. Hence, the problem addressed in this paper can be stated as follows.

Problem 1 (Distributed Connectivity Control). Given an initially connected network $\mathcal{G}(t_0)$ consisting of n agents as in (1.1), determine local control laws that regulate the neighbor set $\mathcal{N}_i(t)$ of every agent so that the dynamic network $\mathcal{G}(t)$ is connected for all time.

Problem 1 equivalently implies that we want the network $\mathcal{G}(t)$ to be invariant with respect to connectivity. We achieve this goal by choosing an equivalent formulation,



Fig. 1.2. 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.

using the algebraic representation of a dynamic graph. In particular, the structure of any dynamic graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ can be equivalently represented by a dynamic *laplacian matrix*

$$L(t) = \Delta(t) - A(t), \quad (1.3)$$

where $A(t) = (a_{ij}(t))$ corresponds to the *adjacency matrix* of the graph $\mathcal{G}(t)$, which is such that $a_{ij}(t) = 1$ if $(i, j) \in \mathcal{E}(t)$ and $a_{ij}(t) = 0$ otherwise and $\Delta(t) = \text{diag}(\sum_{j=1}^n a_{ij}(t))$ denotes the *valency matrix*.² The spectral properties of the laplacian matrix are closely related to graph connectivity. In particular, we have the following lemma.

Lemma 1.3.1 ([14]) *Let $\lambda_1(L(t)) \leq \lambda_2(L(t)) \leq \dots \leq \lambda_n(L(t))$ be the ordered eigenvalues of the laplacian matrix $L(t)$. Then, $\lambda_1(L(t)) = 0$ for all t , with corresponding eigenvector $\mathbf{1}$, i.e., the vector of all entries equal to 1. Moreover, $\lambda_2(L(t)) > 0$ if and only if $\mathcal{G}(t)$ is connected.*

1.4 Distributed Topology Control: Challenges and Machinery

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. As 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. 1.1). Such knowledge can be obtained through local *estimates* of the network topology (Sect. 1.4.1), which, along with a *tie breaking* mechanism obtained by means of *gossip algorithms* and distributed *market-based* control (Sect. 1.4.2), ensure connectivity even when combinations of multiple deletion requests could possibly violate it (Fig. 1.2).³

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

³ For details and theoretical results regarding the algorithms proposed in this section, we refer the reader to [4].

Table 1.1.

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

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

$$A_i(t+1) = \neg(A_i(t) \leftrightarrow V_i(t)), \quad (1.4)$$

where the control input $V_i(t) = (v_{jk}^{[i]}(t)) \in \{0, 1\}^{n \times n}$ is such that $v_{jk}^{[i]}(t) = 1$ if a control action is taken to add or delete link (j, k) (Table 1.1). It can be shown that the control input $V_i(t)$ can be decoupled into two disjoint components $V_i^a(t)$ and $V_i^d(t)$ regulating link *additions* and *deletions*, respectively. Furthermore, the local network dynamics (1.4) are essentially a *consensus* (with inputs) on the adjacency matrix estimates $A_i(t)$, providing every agent with a *rough* picture of the overall network, as desired.

1.4.2 Controlling Addition and Deletion of Links

Regarding the component $V_i^a(t) = (v_{jk}^{[i]a}(t))$ 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.,

$$v_{jk}^{[i]a}(t) \triangleq \underbrace{\left((j \neq i) \wedge (k \neq i) \right)}_{\text{add all existing links provided by neighbors}} \vee \underbrace{\left(x_k(t) \in \mathcal{B}_r(x_j(t)) \right)}_{\text{maintain current neighbors and add new neighbors}}, \quad (1.5)$$

where $\mathcal{B}_r(x_i(t)) = \{y \in \mathbb{R}^p \mid \|y - x_i(t)\|_2 < r\}$ denotes an open ball of radius $r > 0$ centered at $x_i(t) \in \mathbb{R}^p$.

Unlike link additions, deletion of nearest neighbor links is a challenging task. Although knowledge of the estimate $\mathcal{G}_i(t)$ allows every agent i to determine adjacent links that if deleted individually, preserve network connectivity (Fig. 1.1), it is not sufficient for dealing with simultaneous link deletions by multiple non-adjacent agents that may disconnect $\mathcal{G}(t)$ (Fig. 1.2). 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

Algorithm 1 Auction Mechanism for Agent i

- 1: Compute the set of *safe* links (i, j) with $j \in \mathcal{N}_i(t)$ that if deleted from $\mathcal{E}_i(t)$, then $\mathcal{G}_i(t)$ remains connected (see Lemma 1.3.1).
 - 2: Initialize a request $r_i \triangleq [i \ j \ b]^T \in \mathbb{R}^3$ consisting of the link (i, j) to be deleted and a bid $b \in \mathbb{R}$, such that $b > 0$ if $\mathcal{S}_i(t) \neq \emptyset$ and $b = 0$ otherwise, indicating how “important” this request is.
 - 3: Initialize a set of max-bids $\mathcal{M}_i(t) \triangleq \{r_i(t)\} \in 2^{\mathbb{R}^3}$ and a binary vector of tokens $T_i(t) \triangleq [0 \ \dots \ 1_i \ \dots \ 0]^T \in \{0, 1\}^n$ indicating the start of an auction.
 - 4: **while** $(\wedge_{j=1}^n T_{ij}(t)) = 0$ **do**
 - 5: Collect tokens from neighbors only, i.e.,

$$T_i(t+1) := T_i(t) \vee (\vee_{j \in \mathcal{N}_i(t)} T_j(t))$$
 - 6: Apply a max-consensus update on $\mathcal{M}_i(t)$, i.e.,

$$\mathcal{M}_i(t+1) := \{r_j \mid j = \operatorname{argmax}_{r_k \in \cup_{l \in \{\mathcal{N}_i(t), i\}} \mathcal{M}_l(t)} \{r_{k3}\}\}$$
 - 7: **end while**
 - 8: Compute the *winner* link $w_i(t)$ of the auction,
 $w_i(t) \triangleq \{(r_{j1}, r_{j2}) \mid r_j \in \mathcal{M}_i(t)\}$
-

framework consists of a sequence of auctions as in Algorithm 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 $V_i^d(t) = (v_{jk}^{[i]d}(t))$ can then be defined as

$$v_{jk}^{[i]d}(t) \triangleq (w_i(t) = (j, k)) \wedge (|w_i(t)| = 1). \quad (1.6)$$

Note that $|w_i(t)| > 1$ implies either a tie in the maximum bids, in which case (1.6) results in $V_i^d(t) = \mathbf{0}$ for all agents i so that no link is deleted from any network estimate $\mathcal{E}_i(t)$. Clearly, the existence of some notion of *synchronization* of all agents to the same auction is necessary for correctness of the proposed approach.

1.5 Simulation and Experimental Results

The distributed algorithm for preserving connectivity discussed in Sects. 1.3–1.4 was tested in simulation and experimentally on a team of nonholonomic differential drive robots. The following discussion details the experiment design and infrastructure. Additionally, we discuss some of the challenges we addressed during the transition from theory to experimentation.

1.5.1 Implementation Details

The algorithm was implemented in C++ using *Player*, an open-source robotics framework [15]. The *Player* server enables network communications between multiple robots. *Gazebo* is a three-dimensional open-source simulation environment with dynamics and collision detection that integrates with *Player*, allowing the same code base to be used in both simulation and experimentation on the real hardware.

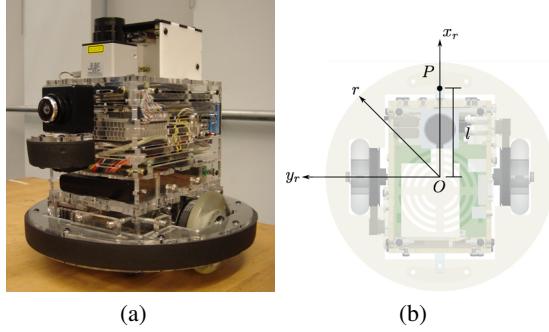


Fig. 1.3. A small form-factor experimental platform equipped with on-board computation and wireless communication (Fig. 3(a)). A top view of the robot model used in both simulation and experimentation showing the body-fixed coordinate system. P is a reference point on the robot whose position is regulated by the kinematic control (Fig 3(b)).

The algorithm was tested in simulation via *Gazebo* and on the experimental infrastructure presented in [16], which consists of a team of small differential-drive robots (one of which is shown in Fig. 3(a)), an indoor tracking system for ground-truth purposes, and a computer infrastructure to support wireless communication and data logging. Accurate models of the robots were created for use in simulation to emulate the real robots. Further, the asynchronous and distributed nature of the hardware was emulated by creating separate execution threads for each agent where all inter-agent communication was accomplished through the *Player* server.

We consider a simple model of a point robot with coordinates (x, y) in the world coordinate system. On the differential-drive robot in Fig. 3(b), these are the coordinates of a reference point P on the robot which is offset from the axle by a distance l . The velocity of the reference point P can be converted into linear and angular velocities for the robot through the equations below:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\frac{\sin \theta}{l} & \frac{\cos \theta}{l} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}.$$

It is well-known that if a robot's reference point is at the point P , and if r is the radius of a circle circumscribing the robot, all points on the robot lie within a circle of radius $l + r$ centered at P .

Each robot begins the experiment with knowledge of its own identification number and the total number of other robots. After estimating its pose using a local extended Kalman Filter that fuses information from the tracking system and the robot's odometry, the robot broadcasts its current ID, position, and estimate of the connectivity as well as any current auction information. Each robot also listens for incoming messages, from which it constructs an estimate of the network graph as well as determines the positions of its neighbors (defined by the network graph). Concurrently, the robots control to avoid neighbors but maintain any existing edges in the graph. Edge removal only occurs at the completion of a distributed auction.

As discussed in Lemma 1.3.1, the second eigenvalue, λ_2 (or the Fiedler value), determines the connectivity of a graph. If $\lambda_2 > 0$, then the graph is connected. The focus of this work is to consider the maintenance of connectivity between the team of robots. Therefore, we consider the estimated value of λ_2 by each agent as the metric of the success or failure of the algorithm to maintain connectivity, as depicted by Fig. 1.6. Note that we assume that initially the team of robots are connected in order to assure connectivity is preserved.

1.5.2 Transitioning from Theory to Experimentation

As this work considers the verification of the theory proposed in Sects. 1.3–1.4, a discussion of the transition from theory to experimentation on a distributed team of robots is merited. Environments such as MATLAB, where shared memory (resulting in “instantaneous” communication), serial and synchronous updates (versus asynchronous parallel computation), and perfect state information yield convincing, yet non-realistic simulation results. However, this work demonstrates that indeed, the theoretic guarantees proposed hold true in implementation. Further, algorithms that rely on concurrent estimation through distributed consensus algorithms are practical and relevant to experimental implementations given the current technological level. The greatest challenge in validating the theory on a team of robots is ensuring the integrity and correctness of the asynchronous and parallel computation as well as message passing with time delays. Further, implementation details such as computational capabilities, message packet size, pose estimation, robot dynamics, networking protocols, and bandwidth constraints represent some, but not all, necessary considerations. While there has been limited validation that theory, such as that proposed in this work, is realistically feasible, the experimental results verify that distributed algorithms of this nature work effectively in theory, simulation, and experimentation and offer numerous benefits to applications requiring distributed inter-agent connectivity.

1.5.3 Simulations

The effectiveness of the algorithm to preserve connectivity during non-trivial control operations that result in edge creation, maintenance, and deletion in the communication graph determines its effectiveness. Figures 4(a) and 4(c) depict two scenarios where the connectivity of the system is specified to require at least one or two neighbors, respectively, at all times. Such a definition is often referred to as k -connectivity, where in this case $k = 1$ or $k = 2$ (see [14] for details). By ensuring that $\lambda_2 > k - 1$, we are able to specify the minimal number of neighbors required for each agent. It is clear from Figs. 4(b) and 4(d) that the system remains connected ($\lambda_2 > 0$) and that the algorithm preserves the specified minimum number of neighbors throughout the simulation.

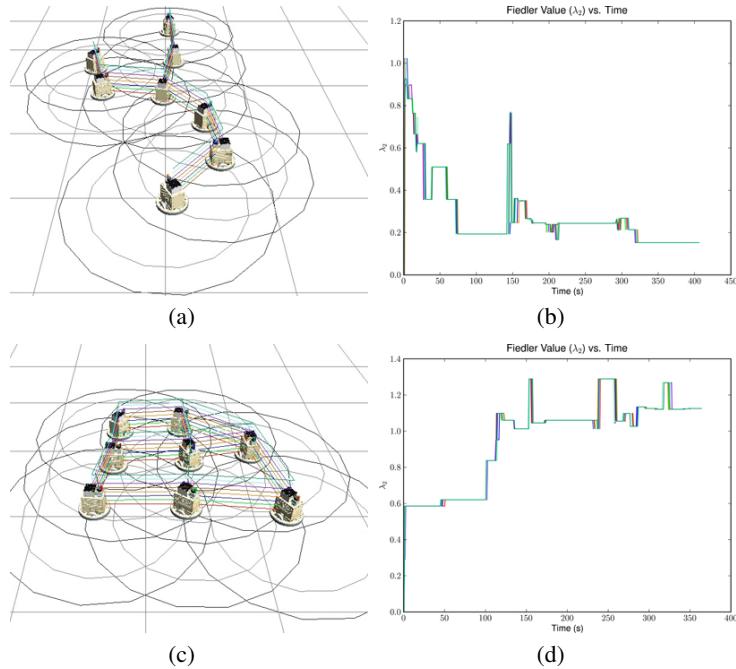


Fig. 1.4. Eight robots control to preserve connectivity in simulation. The algorithm accommodates multiple degrees of connectivity such as requiring that all robots have at least one (Fig. 4(a)) or two (Fig. 4(c)) neighbors. Note that in both cases the system remains connected ($\lambda_2 > 0$) and the value of λ_2 increases when requiring two neighbors (Fig. 4(d)).

1.5.4 Experiments

We evaluated the algorithm experimentally on a team of robots. The results of an experimental trial run are depicted in Fig. 1.5. Figure 1.6 shows the second smallest eigenvalue, λ_2 , computed by each robot from the locally estimated connectivity graph throughout the trial run. The non-negativity of λ_2 confirms the preservation of connectivity.

1.5.5 Limitations

It is worth noting some of the limitations of the algorithm and implementation that are of a pragmatic nature which we are currently addressing. The present implementation of *Player* only supports TCP communication. For this reason, the messaging is peer-to-peer. However, since each robot is communicating with each other robot, this approach is very similar to broadcast communication, but with the beneficial delivery guarantees of TCP and the negative computational and networking costs of $O(n)$ messages versus a single message in a “true” broadcast paradigm. The incorporation of UDP into the *Player* framework is currently being pursued to allow

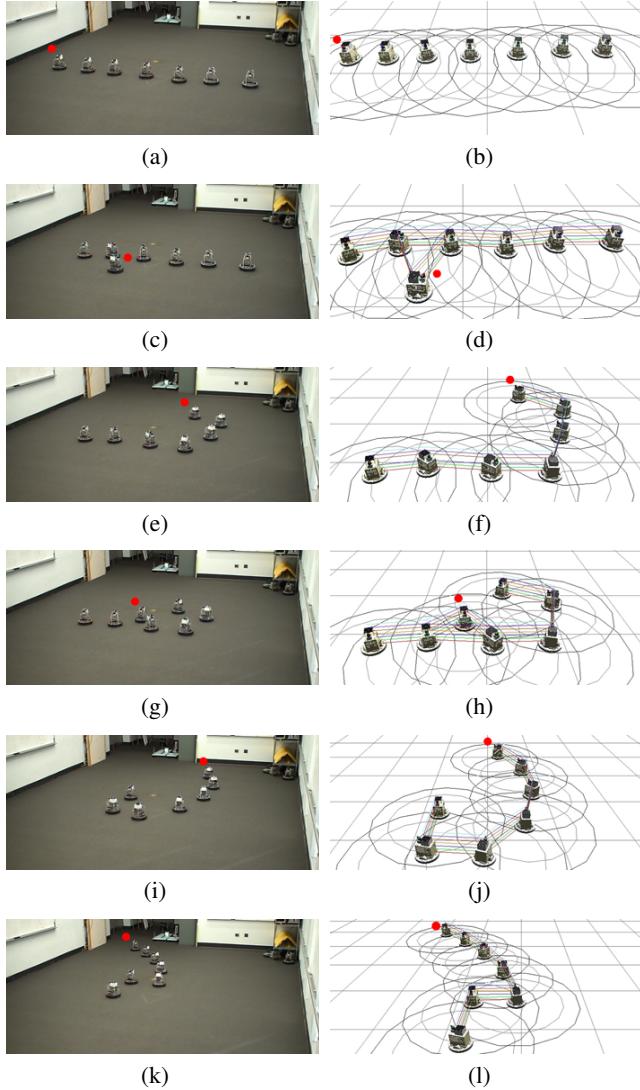


Fig. 1.5. The evolution of an experimental trial run and the corresponding algorithm visualization. The visualization was generated during the experiment using the same data available to the robots through the distributed messaging. During the trial, the team of robots control reactively to the changes in the network topology and inter-agent interactions resulting from the movement of a robot controlled via joystick input (marked by a red circle). The robots start with limited information, knowing only the total number of robots, in this case seven (Figs. 5(a)-5(b)). After computing their local pose, the robots begin broadcasting their ID, pose, and adjacency and auction information. Based on these messages, the robots build a local estimate of the connectivity graph. Each colored line in the visualization depicts the current connectivity graph estimate for a particular robot. The robots create, maintain, and delete links based on inter-agent separation distances (Figs. 5(c)-5(l)). The inner and outer black circles around each robot indicate the communication and edge creation bounds, respectively. The radius of the inner and outer circles were set to 0.75 m and 1 m to accommodate the limited experimental space.

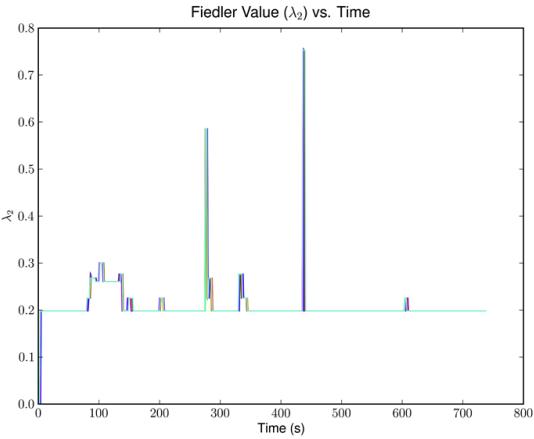


Fig. 1.6. Initially, each robot only has access to the total number of robots. Through local sensing and message passing, the robots estimate their pose as well as the connectivity of the system. For this reason, the system is initially disconnected. As the network graph of the system is estimated, the connectivity is locally computed based on λ_2 . After which point, the underlying control law and auction-based edge deletion ensure that connectivity is preserved as edges are created based on spatial proximity.

broadcast communication. Also, the algorithm does not accommodate cluttered or complex environments as it only takes into account inter-agent collision avoidance. We are currently considering the introduction of line-of-sight constraints and obstacle avoidance. Another limitation results from the limited experimental space. The communication limit, which also determines the regions permitting edge addition and deletion, was artificially set to a small value (1 m). This approach emulates a system with limited communication but does not completely capture the complexities of bounded communication systems. Additionally, the wireless communication leveraged a single access point rather than multiple access points or ad-hoc networking.

1.6 Conclusion and Future Work

In this work we discuss the experimental validation of a distributed algorithm that preserves the connectivity of a team of robots. We also review a gradient-descent control law that preserves the system connectivity by ensuring that links to neighbors are maintained. The algorithm requires limited local information and communication between agents to determine the addition or deletion of network links through distributed consensus and market based auctions. Non-trivial simulation and experimental results demonstrate the effectiveness of the algorithm as a means to guarantee connectivity in a team of robots.

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, Sept. 2004.
2. 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, Aug. 2006.
3. M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Maintaining network connectivity and performance in robot teams," *Journal of Field Robotics*, vol. 25, no. 1, pp. 111–131, 2008.
4. M. M. Zavlanos and G. J. Pappas, "Distributed connectivity control of mobile networks," in *Proc. of the IEEE Conf. on Decision and Control*, New Orleans, LA, Dec. 2007.
5. D. P. Spanos and R. M. Murray, "Robust connectivity of networked vehicles," in *Proc. of the IEEE Conf. on Decision and Control*, Bahamas, Dec. 2004, pp. 2893–2898.
6. M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, Aug. 2007.
7. M. Mesbahi, "On state-dependent dynamic graphs and their controllability properties," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 387–392, Mar. 2005.
8. Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, Jan. 2006.
9. M. C. DeGennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *Proc. of the IEEE Conf. on Decision and Control*, San Diego, CA, Mar. 2006, pp. 3628–3633.
10. M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proc. of the IEEE*, vol. 94, no. 7, pp. 1257–1270, July 2006.
11. M. Lindhe, H. Johansson, and A. Bicchi, "An experimental study of exploiting multipath fading for robot communications," in *Robotics: Science and Systems*, Atlanta, GA, June 2007.
12. M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Towards the deployment of a mobile robot network with end-to-end performance guarantees," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Orlando, FL, May 2006, pp. 2085–2090.
13. F. Zeiger, N. Kraemer, and K. Schilling, "Commanding mobile robots via wireless ad-hoc networks - a comparison of four ad-hoc routing protocol implementations," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Pasadena, CA, May 2008, pp. 590–595.
14. C. Godsil and G. Royle, *Algebraic Graph Theory*. Springer Graduate Texts in Mathematics #207, 2001.
15. B. P. Gerkey, R. T. Vaughan, and A. Howard, "The Player/Stage Project: Tools for multi-robot and distributed sensor systems," in *Proc. of the Int. Conf. on Advanced Robotics*, Coimbra, Portugal, June 2003, pp. 317–323.
16. N. Michael, J. Fink, and V. Kumar, "Experimental testbed for large multi-robot teams: Verification and validation," *IEEE Robotics and Automation Magazine*, vol. 15, no. 1, pp. 53–61, Mar. 2008.