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. While there has been significant progress in recent years in the study of estimation and control of dynamic network graphs, limited attention has been paid to the experimental validation and verification of such algorithms on distributed teams of robots. In this work we conduct an experimental study of a non-trivial distributed connectivity control algorithm on a team of seven nonholonomic robots as well as in simulation. The implementation of the algorithm is completely decentralized and asynchronous, assuming that each robot only has access to its pose and knowledge of the total number of robots. All other necessary information is determined via message passing with neighboring robots. We show that such algorithms, requiring complex inter-agent communication and coordination, are feasible as well as highly successful in enabling a network of robots to adapt to disturbances while preserving connectivity.

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 many 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]. In this context, changes in radio signal strength can be used to guide relative movements of robots to ensure connectivity [2]. While this approach is decentralized and no explicit communication is necessary to maintain connectivity, theoretical guarantees are difficult to prove. In contrast, [3] suggests a distributed feedback control framework that imposes no restrictions on the network structure other than the

* This work was partially supported by NSF Grants IIS-0427313 and IIP-0742304, ARO Grants W911NF-05-1-0219 and W911NF-04-1-0148, and ONR Grant N00014-07-1-0829.

O. Khatib et al. (Eds.): Experi. Robotics: The 11th Intern. Sympo., STAR 54, pp. 117–126.
springerlink.com

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 [3] as well as message passing with time delays through experimental analysis on a team of robots.

2 Related Work

Due to their frequent appearance in multi-agent systems, dynamic networks have already received considerable attention. In [4], 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 [5], while in [6], a controllability framework for state-dependent dynamic graphs is developed. In [7], 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 [8].

In this work we leverage market-based auctions to determine communication link deletion while preserving connectivity in a decentralized manner. Similar methods of task consensus through distributed market-based auctions are surveyed in [9].

The effects of interference on wireless communication in complex environments are of pragmatic interest in experimental robotics. In [10], 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 [11] 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 [12].

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 of a *dynamic graph* $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, where $\mathcal{V} = \{1, \ldots, n\}$ 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, $(i,j) \notin \mathcal{E}(t)$.

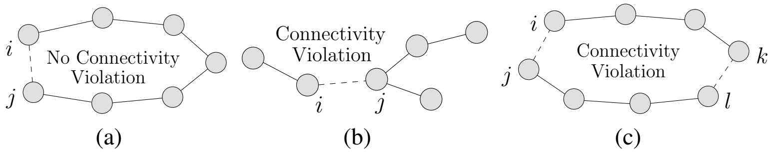


Fig. 1. Control challenges requiring knowledge of the network structure and multiple link deletions. Without such knowledge, deletion of a link (i, j) can either preserve (Fig. 1(a)) or violate (Fig. 1(b)) connectivity. In the absence of an agreement protocol, simultaneous deletion of links (i, j) and (k, l) violates connectivity (Fig. 1(c)).

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

$$\dot{x}_i(t) = u_i(t) - \sum_{j \in \mathcal{N}_i(t)} \nabla_{x_i} V_{ij}(t), \tag{1}$$

where $u_i(t)$ corresponds to an external input and V_{ij} is the artificial potential function

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

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), 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, 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*

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

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.

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

Lemma 3.1 ([13]). *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.*

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 (Figs. 1(a)–1(b)). Such knowledge can be obtained through local *estimates* of the network topology (Sect. 4.1), which, along with a *tie breaking* mechanism obtained by means of *gossip algorithms* and distributed *market-based* control (Sect. 4.2), ensure connectivity even when combinations of multiple deletion requests could possibly violate it (Fig. 1(c)).[2]

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 $A_i(t) = \big(a_{jk}^{[i]}(t)\big)$ denote the adjacency matrix associated with the graph $\mathcal{G}_i(t)$ and define its dynamics by

$$A_i(t+1) = \neg\big(A_i(t) \leftrightarrow V_i(t)\big), \tag{2}$$

where the control input $V_i(t) = \big(v_{jk}^{[i]}(t)\big) \in \{0,1\}^{n \times n}$ is such that $v_{jk}^{[i]}(t) = 1$ if a control action is taken to add or delete link (j, k). 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 (2) 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.

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.,

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

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

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(a)), it is not sufficient for dealing with simultaneous link deletions by multiple non-adjacent agents that may disconnect $\mathcal{G}(t)$ (Fig. 1(c)). 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 market-based control framework, proposed in [3], consists of a sequence of auctions, 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 $V_i^d(t) = \big(v_{jk}^{[i]d}(t)\big)$ can then be defined as

$$v_{jk}^{[i]d}(t) \triangleq \big(w_i(t) = (j,k)\big) \wedge (|w_i(t)| = 1). \qquad (3)$$

Note that $|w_i(t)| > 1$ implies a tie in the maximum bids, in which case (3) 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.

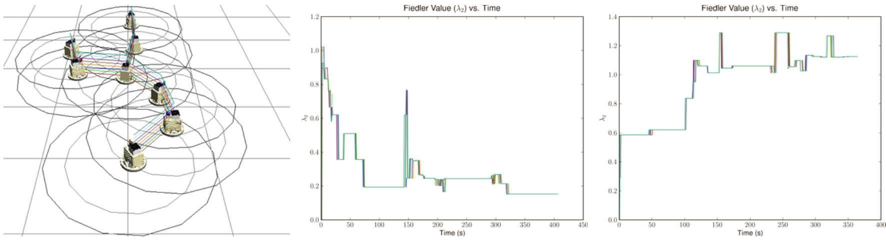
5 Simulation and Experimental Results

The distributed algorithm for preserving connectivity discussed in Sects. 3–4 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.

5.1 Implementation Details

The algorithm was implemented in C++ using *Player*, an open-source robotics framework [14]. 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.

The algorithm was tested in simulation via *Gazebo* and on the experimental infrastructure presented in [15], which consists of a team of small differential-drive robots, 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



(a) (b) (c)

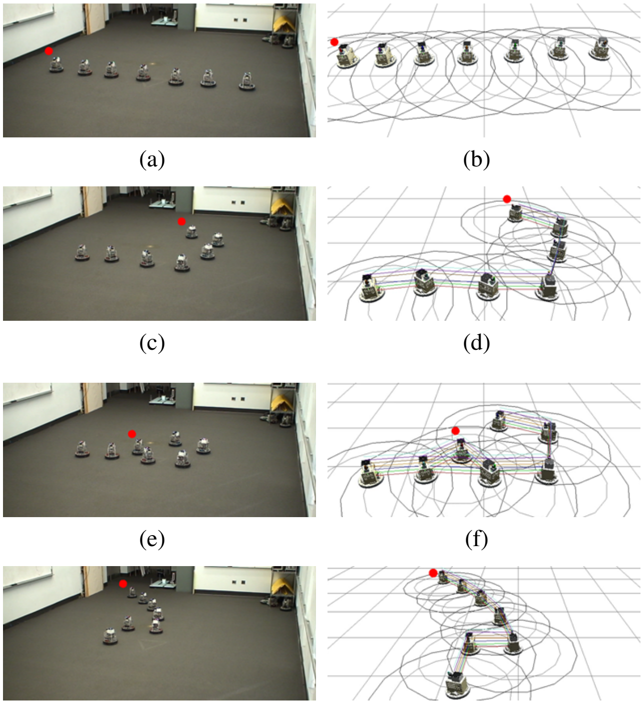
Fig. 2. Simulation of a 1-connectivity and 2-connectivity control scenario for a network of eight robots. Figure 2(a)shows an instance of the robotic network for the 1-connectivity scenario. Colored lines indicate the network estimates of the individual robots. Similarly, Figs. 2(b) and 2(c) show the evolution of the Fiedler eigenvalue λ_2, computed locally by each agent based on their network estimate. As before, different colors correspond to different robots. Since the connectivity of the local network estimates implies the connectivity of the overall network [3], Figs. 2(b) and 2(c) can be used to verify the desired connectivity specifications. Note that although the estimates are similar, the system is clearly asynchronous.

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. However, as the experimental platform is a differential drive robot, we transform the kinematic agent controls of (1) to linear and angular velocities using the technique of feedback linearization [16].

The experiment requires that every robot has knowledge of its own identification number (ID), the total number of robots, as well as its pose, which it estimates using a local extended Kalman Filter that fuses information from the tracking system and the robot's odometry. During the first few iterations of the algorithm every robot seeks to obtain an estimate of the overall communication network by broadcasting its current ID and network estimate (initialized as empty), while listening for incoming messages from its neighbors. In this way, after the first iteration of the algorithm, every robot populates its network estimate with the estimate of its neighbors. Concurrently, every robot broadcasts its position and current auction information, and plans a collision-free motion that adds or removes communication links with its neighbors, according to the distributed market-based controller. Clearly, any changes in the network structure can only happen when the network estimates of all robots have been sufficiently populated.

As discussed in Sects. 3–4, the sign of the second eigenvalue λ_2 (or the Fiedler value) of the laplacian matrix of a graph captures its connectivity. In particular, if $\lambda_2 > 0$, then the graph is connected (Lemma 3.1). Hence, λ_2 can be used as a metric of the success or failure of the algorithm to maintain connectivity of a mobile robotic network, as depicted by Fig. 4. Note that we assume that initially the team of robots are connected in order to assure that connectivity is preserved.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 3. The evolution of an experimental trial run and the corresponding algorithm visualization. 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. 3(a)-3(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. 3(c)-3(h)). The inner and outer black circles around each robot indicate the edge creation and communication bounds, respectively. The radius of the inner and outer circles were set to 0.75 m and 1 m.

5.2 Transitioning from Theory to Experimentation

As this work considers the verification of the theory proposed in Sects. 3–4, 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 inherently unrealistic simulation results. A significant challenge in validating the theory relevant to this work 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

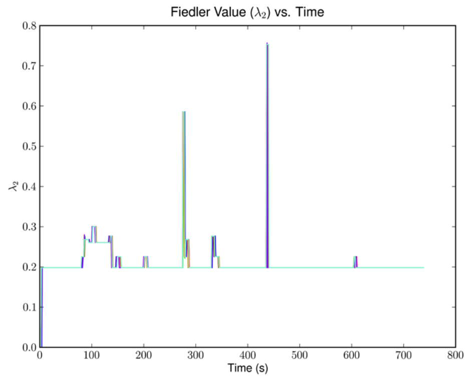


Fig. 4. The Fiedler eigenvalue, λ_2, computed locally by each robot based upon their network estimate (denoted by different colors). Initially, the robots have no information of the network topology, hence, their network estimates appear disconnected. As these estimates become populated, market-based coordination ensures connectivity of the overall network ($\lambda_2 > 0$). Fig. 4 demonstrates the correctness of the proposed implementation by the fact that the connectivity of the local network estimates implies the connectivity of the overall network [3].

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.

5.3 Simulations

In this section we test the ability of the proposed algorithm to preserve connectivity of the overall network during non-trivial control operations that result in creation, maintenance, and deletion of communication links. Figure 2(a) shows an instance of a robotic network that is required to remain k-connected for $k = 1$ (see [13] for details). Theoretically, k-connectivity is captured by the eigenvalue condition $\lambda_2 > k - 1$. This condition is also satisfied in simulation, as shown in Figs. 2(b) and 2(c).

5.4 Experiments

We evaluated the algorithm experimentally on a team of robots. The results of an experimental trial run are depicted in Fig. 3. Figure 4 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.

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 broadcast communication. Additionally, the current 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.

6 Conclusion

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. Cortes, J., Martinez, S., Bullo, F.: Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. IEEE Transactions on Automatic Control 51, 1289–1298 (2006)
2. Hsieh, M.A., Cowley, A., Kumar, V., Taylor, C.J.: Maintaining network connectivity and performance in robot teams. Journal of Field Robotics 25, 111–131 (2008)
3. Zavlanos, M.M., Pappas, G.J.: Distributed connectivity control of mobile networks. In: Proc. of the IEEE Conf. on Decision and Control, New Orleans, LA, December 2007, pp. 3591–3596 (2007)
4. Spanos, D.P., Murray, R.M.: Robust connectivity of networked vehicles. In: Proc. of the IEEE Conf. on Decision and Control, Bahamas, December 2004, pp. 2893–2898 (2004)
5. Ji, M., Egerstedt, M.: Distributed coordination control of multiagent systems while preserving connectedness. IEEE Transactions on Robotics 23, 693–703 (2007)
6. Mesbahi, M.: On state-dependent dynamic graphs and their controllability properties. IEEE Transactions on Automatic Control 50, 387–392 (2005)
7. Kim, Y., Mesbahi, M.: On maximizing the second smallest eigenvalue of a state-dependent graph laplacian. IEEE Transactions on Automatic Control 51(1), 116–120 (2006)
8. DeGennaro, M.C., Jadbabaie, A.: Decentralized control of connectivity for multi-agent systems. In: Proc. of the IEEE Conf. on Decision and Control, San Diego, CA, March 2006, pp. 3628–3633 (2006)
9. Dias, M.B., Zlot, R., Kalra, N., Stentz, A.: Market-based multirobot coordination: A survey and analysis. Proc. of the IEEE 94(7), 1257–1270 (2006)
10. Lindhe, M., Johansson, H., Bicchi, A.: An experimental study of exploiting multipath fading for robot communications. In: Robotics: Science and Systems, Atlanta, GA (June 2007)

11. Hsieh, M.A., Cowley, A., Kumar, V., Taylor, C.J.: 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 (2006)
12. Zeiger, F., Kraemer, N., Schilling, K.: 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 (2008)
13. Godsil, C., Royle, G.: Algebraic Graph Theory. Springer Graduate Texts in Mathematics, vol. 207. Springer, Heidelberg (2001)
14. Gerkey, B.P., Vaughan, R.T., Howard, A.: 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 (2003)
15. Michael, N., Fink, J., Kumar, V.: Experimental testbed for large multi-robot teams: Verification and validation. IEEE Robotics and Automation Magazine 15, 53–61 (2008)
16. Sastry, S.: Nonlinear Systems: Analysis, Stability, and Control. Interdisciplinary Applied Mathematics, vol. 10. Springer, New York (1999)