Hindawi Publishing Corporation
Journal of Robotics
Volume 2011, Article ID 571485, 12 pages
doi:10.1155/2011/571485

Research Article

Maintaining Wireless Connectivity Constraints for Robot Swarms in the Presence of Obstacles

Joel M. Esposito

Department of Systems Engineering, United States Naval Academy, Annapolis, MD 21401, USA

Correspondence should be addressed to Joel M. Esposito, esposito@usna.edu

Received 31 December 2010; Revised 24 May 2011; Accepted 17 June 2011

Academic Editor: John T. Y. Wen

The swarm paradigm of multirobot cooperation relies on a distributed architecture, where each robot makes its own decisions based on locally available knowledge. But occasionally the swarm members may need to share information about their environment or actions through some type of ad hoc communication channel, such as a radio modem, infrared communication, or an optical connection. In all of these cases robust operation is best attained when the transmitter/receiver robot pair is (1) separated by less than some maximum distance (range constraint); and (2) not obstructed by large dense objects (line-of-sight constraint). Therefore to maintain a wireless link between two robots, it is desirable to simultaneously comply with these two spatial constraints. Given a swarm of point robots with specified initial and final configurations and a set of desired communication links consistent with the above criteria, we explore the problem of designing inputs to achieve the final configuration while preserving the desired links for the duration of the motion. Some interesting conclusions about the feasibility of the problem are offered. A potential field-based optimization algorithm is provided, along with a novel composition scheme, and its operation is demonstrated through both simulation and experimentation on a group of small robots.

1. Introduction

Large teams of mobile robots, referred to as *swarms* can be more effective in accomplishing certain tasks, as compared with a single, possibly more sophisticated, robot. The advantages of the swarm are especially apparent in applications that benefit from spatially distributed sensing, such as environmental sampling [1], coordinated map making [2], and search [3]. Manipulating and transporting large objects is an application which can benefit from spatially distributed actuation [4–7]. In contrast to centralized control methods, the swarm paradigm of multirobot cooperation relies on a distributed architecture, where each robot makes its own decisions based on locally available knowledge. Advantages of such an approach include improved scalability with respect to swarm size, robustness with respect to the failure of a single swarm member, and the possibility of a human operator controlling swarm wide behavior through a low-dimensional set of input parameters [8, 9].

However, in many applications, it is impossible or inefficient to employ truly independent control algorithms on each agent. In order to complete their task the swarm members may need to share information about their intentions or their environment. Indeed, many proposed control laws in the literature require that each member of the swarm is connected to the group through some type of ad hoc, low-power, wireless communication channel, such as a radio or optical links. Power limitations and phenomena such as secondary reflections and shadow effects create a variety constraints on the relative positions of the transmitter and receiver. We abstract these more complex electromagnetic phenomena and work with a simpler two-component communication constraint. The first is a limitation on the maximum distance between the transmitter and receiver, referred to here, and in other works, as the *Range Constraint*. Considerably less attention has been given to *Line-of-Sight Constraints*—necessitated by the difficulty of reliably transmitting wireless messages through large dense obstacles. Together we term these two spatial constraints *Communication Constraints*. If a pair of robots meet these constraints, we assume they can establish a *wireless link*. While this treatment is idealized, it is a significant improvement over the “range only” constraints considered in most of the connectivity control literature. Of course in addition to these constraints the robots may have some overall motion objectives (either individually or as a group), such as moving toward a goal and avoiding obstacles.

In this paper we address the problem of navigating the swarm, in the plane, from an initial configuration to a specified final configuration, while maintaining a prespecified list of wireless links between certain robots (range plus line-of-sight). After a review of related work below, we provide a formal problem statement in Section 3. We consider existence of solutions in Section 4 and necessary properties of solution trajectories. In Section 5 we introduce potential fields for the goal/obstacle avoidance, range and line-of-sight objectives. Section 6 discusses how to compose these sometimes disparate objectives and provides a computational algorithm for assigning motion directions. Simulation and hardware-based experimental demonstrations of the algorithm's operation are included in Section 7 followed by concluding remarks in Section 8.

2. Related Work

In this section we review some common notions of robot swarm connectivity and their role in flocking behavior and formation control. We then narrow the scope of the discussion to related work on explicit control of swarm connectivity, and the relationship to the approach presented here. We end with a discussion of the line-of-sight constraint.

2.1. Notions of Connectivity. Most discussions on connectivity within robot swarms employ a neighbor or proximity graph modeling paradigm [10], where each vertex in the graph represents a robot and each graph edge represents a wireless communication link. The criteria to establish a link is almost always based on the physical distance (i.e., range) between the robots [11, 12]. We adopt both the graph theoretic modeling paradigm and the range constraint in this paper. As an extension, some works [13, 14] consider multihop connectivity, still using interrobot range as the underlying criteria. If an edge joins two vertices, the corresponding robots are considered locally connected. The swarm as a whole is considered globally connected if an edge-path of finite length exists between any two vertices in the graph. Global connectivity can be verified algebraically by determining if the second smallest Eigenvalue of the graph's LaPlacian matrix is greater than zero. Larger values indicate a more strongly connected configuration.

2.2. Role of Connectivity in Consensus and Stability. All works on flocking and formation control rely on some underlying notion of swarm connectivity to prove stability and convergence. In general the primary objective of *flocking* can be thought of as establishing a consensus on individual robot velocity vectors [15]. The work by Reynolds [16] is considered to be the inspiration for many works on flocking. While that work did not formally describe the notion of global connectivity in terms of graphs, the control law stipulated certain informational dependencies—namely, that each robot should know the position and velocity of the other agents in its neighborhood. These assumption are now considered standard, and we make similar ones in this paper. Efforts to formally prove that the swarm reaches a consensus regarding their velocity vectors ultimately employed a neighbor graph to reflect these dependencies, showed that global connectivity is a necessary condition for consensus, and proved that the convergence rate is determined by the second smallest Eigenvalue of the graph's LaPlacian matrix [10–12]. In these works the relative pose of the robots is not specified and the connection topology is entirely range based, and therefore dynamic, so the control laws must include a potential function that maintains interrobot ranges within a certain tolerance, similar to [17].

Formation control involves moving a group of robots while maintaining a fixed relative pose between them. In these works some type of graph theoretic framework is also used [18]. Typically the connection topology is static, where certain robots are designated as leaders, and others follow maintaining specific edges in the graph. Here the connectivity properties of the graph can be used to prove stability [19–23]. In both formation and flocking works, while there is a acknowledged relationship between stable flocking and connectivity, the primary objective is the former and the later is a necessary condition.

2.3. Connectivity as a Primary Control Objective. There are several works that explicitly treat connectivity as the primary control objective. They can be loosely divided into *optimization-based* (or open loop) approaches and *feedback* approaches. In [24] the authors synthesize configurations and paths that maximize swarm's global connectivity, as measured by the second smallest Eigenvalue of the graph's LaPlacian matrix using a Semidefinite Programming formulation. The work in [13] more closely resembles the problem considered here. A desired graph is specified and must be maintained for all time. Two hop, range-based edges are considered, and they show the set of all such connected configurations is a star-shaped set. As a result, the initial path supplied to the receding horizon optimizer consists of contracting the swarm to a point, following a straight line trajectory to the goal, and expanding the swarm again. An optimization criteria called *connectivity robustness* is used to refine the result. In both cases numerical techniques are used off-line to compute an open loop path. In our work, numerical methods are employed in real time as part of a closed loop feedback control law.

An example of a feedback-based method is in [25]. The problem of maintaining specific edges in the neighbor graph is phrased as controlling the dynamics of the adjacency matrix through a set of inequalities [14]. A potential field-based controller is used in [26]. A potential function is de-fined on the Cartesian product of swarm member poses, and loss of global connectivity is modeled as a virtual obstacle in that space. Locally optimal configurations are achieved. Unlike [13] or [27], this approach does not specify individual edges to be maintained.

Like the works discussed above, the work presented here and our earlier work [27] explicitly treat connectivity as a control objective.

2.4. Line of Sight. Several authors in the wireless networking community have studied the adverse impact of transmitter/receiver occlusions from an electromagnetic field strength perspective and the modeling implications of this for mobile robotics have been discussed in detail [28]. Practitioners have long observed the effect. Applications in multirobot coverage [29] and map making [30], for example, assume that information exchanges only occur when line of sight is established. Also, the pervasive use of computer vision for robotics tasks such as cooperative localization [31], or search and pursuit [32] has also necessitated the consideration of line-of-sight constraints.

In some of these cases, loss of line of sight is considered an unfortunate "fact of life" [30]; in other cases it is exploited to determine the connectivity properties of the environment [29, 31]. But none of these works explicitly includes a motion controller that actively prevents such losses from occurring.

The principle difference between the connectivity work discussed in the previous section and the work presented in this paper, is that we modify the local connection/edge formation criteria to explicitly maintain line-of-sight constraints as well as the widely used single-hop, range constraint. To our knowledge this issue has only been considered in a few previous works. A behavior-based approach is described in [33]; while the performance cannot be proven it was illustrated through simulation. In [34] a planetary rover application was considered and six heuristics for maintaining line of sight were introduced. The approach was largely successful but could not solve all the presented test scenarios. The authors speculated that some of the unsolved scenarios were simply infeasible.

3. Problem Statement

Given n point robots, let $q_i \in \mathbb{R}^2$ be the state vector of robot i. The robots operate in a subset of the plane $C \subset \mathbb{R}^2$ which is populated with obstacles defined by compact sets O_j, $j = 1, \ldots, m$. Motion is generated according to the dynamics

$$\dot{q}_i = u_i, \tag{1}$$

where the velocity input is $u_i \in U \subset \mathbb{R}^2$; q_i is only permitted to evolve in the free space $C_{\text{free}} = C - \bigcup_{j=1}^{m} O_j$. Occasionally we will use $q \in \mathbb{R}^{2n}$ to denote the swarm state—the concatenation of states $q_1 \cdots q_n$; u to represent the concatenation of the input vectors; $\dot{q} = u$ to represent the collective swarm dynamics.

Any given swarm state q induces a communication graph $G(q) = (V, E)$. Each vertex in the graph, $v_i \in V$ represents a robot and each edge $e_{ij} \in E$ represents a wireless communication link between robots i and j. The edge e_{ij} is added to the graph if both of the following conditions are met.

(1) *Range*: $d(q_i, q_j) \leq \rho_{\max}$ where $\rho_{\max}$ is some positive constant indicating the maximum effective range of the transmitter.

(2) *Line of Sight*: $\exists x(s) \in C_{\text{free}}$, for all $s \in [0, 1]$, such that $x(s) = sq_i + (1 - s)q_j$.

Note that d indicates distance as measured by the Euclidian metric. Both constraints model the power limitations of small wireless transmitters discussed in Section 1. A configuration q is said to be *connected* if the induced communication graph G is connected (i.e., if for any node pair i, j there exists an edge path of arbitrary length between them).

We are concerned with the following problem (see Figure 1), which requires the entire swarm to move to a desired position while maintaining certain communication links, G^*.

Problem 1. Given an initial connected configuration $q^o = q(t^o)$, a desired final connected configuration q^f and a graph G^* such that $G(q^o) \supseteq G^*$ and $G(q^f) \supseteq G^*$, determine a function $\mathcal{U} : [t^o, t^f] \rightarrow U$ such that

(1) $q(t^f) = q^f$ (i.e., *Goal*-directed motion);

(2) $G(q(t)) \supseteq G^*$, for all $t \in [t^o, t^f]$ (*Line of Sight* and *Range*).

4. Existence of Solutions

Clearly there are certain combinations of the free space C_{free} and the desired connectivity graph G^* for which the problem may not have a solution. Furthermore, even when a solution exists, there are certain classes of algorithms incapable of solving the problem. The concept of homotopy [35] is intimately related to these existence questions.

4.1. Homotopy Definitions. If $q_1, q_2 : [t^o, t^f] \rightarrow C_{\text{free}}$ are continuous maps (paths), we say that $q_1(t)$ and $q_2(t)$ are *homotopic* if there exists a continuous map $T : [t^o, t^f] \times [0, 1] \rightarrow C_{\text{free}}$ such that

$$\begin{aligned} T(t, 0) &= q_1(t), \\ T(t, 1) &= q_2(t), \quad \forall t. \end{aligned} \tag{2}$$

If such a function exists, we say T is a homotopy. This homotopy defines an equivalence relation on paths. Note that unlike *path homotopy* [35], the endpoints of q_1 and q_2 do not coincide.

Of particular interest in this paper is the *straight-line homotopy*, illustrated in Figure 3 (right),

$$T(t, s) = (1 - s)q_1(t) + sq_2(t), \tag{3}$$

due to its obvious connection to the line-of-sight constraint. If two paths $q_1(t)$, $q_2(t)$ have a straight line homotopy then the line-of-sight constraint is preserved for all t. If the range constraint is violated at any point on the trajectory, the straight-line homotopy can be used to correct the condition.

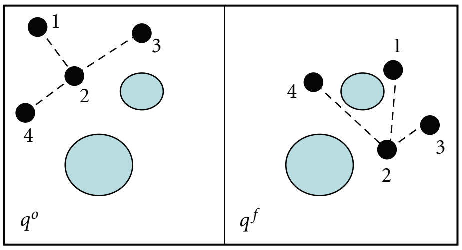


Figure 1: The basic problem considered in this paper. Design a control law to guide the swarm from the initial position, q^o, to the desired final position, q^f, while maintaining desired communication links (range plus line of sight).

4.2. Intrinsic Lack of Solution. Obviously, when C_{free} is not path connected, and q^0 and q^f lie in different connected components, there is no solution to the motion planning problem.

Furthermore, if C_{free} is multiply connected and G^* contains cycles, solutions do not exist for all choices of q^o or q^f. To capture this we apply the concept of *loop homotopy*.

A path q, is considered a loop, λ, if $q(t^o) = q(t^f)$. Note that the trivial loop is the constant loop $\lambda(t) = \lambda(t^o)$, for all t. We can apply the homotopy equivalence relation to loops as well. Similar to general homotopy, if one loop can be continuously deformed into a second loop, the two loops are *loop-homotopic equivalent*.

For a given cycle $G^c \subseteq G^*$, one can connect the points q_i^o corresponding to the vertices in the cycle to form a loop λ^o using straight-line segments; likewise a corresponding loop λ^f, using q^f, can be constructed using the same vertices. See Figure 2 for an example. If these two loops are not in the same homotopic equivalence class, it implies that the loops wrap around the obstacles in such a way that is impossible to go from q^0 to q^f without disconnecting some edges, then no solution to the problem exists.

Remark 1. To ensure the existence of solutions in this paper we only consider path-connected free spaces; and q^0, q^f such that loops corresponding to any cycles of G^* are homotopic to the constant loop.

4.3. Attribute of Complete Solution Algorithms. As remarked earlier, in order to maintain an edge e_{ij}, for all $t \in [t^o, t^f]$, there must exist a straight-line homotopy between $q_i(t)$ and $q_j(t)$. Intuitively a necessary (not sufficient) condition for such a solution is that paths $q_i(t)$ and $q_j(t)$ must pass around the same "side" of an obstacle. See Figure 3. Therefore, if G^* is connected, *all* robots, in some loose sense, must collectively pass around the same "side" of every obstacle—that is, the swarm cannot "split".

This notion is difficult to formalize however. The traditional path-homotopy equivalence relation does not apply because the endpoints of $q_i(t)$ and $q_j(t)$ do not coincide. Also, general homotopy does not preserve the line-of-sight constraint; and the straight-line homotopy does not induce an equivalence relation (it lacks transitivity). Instead we introduce the following definition which formalizes the concept of the paths "passing around the same side of the obstacle".

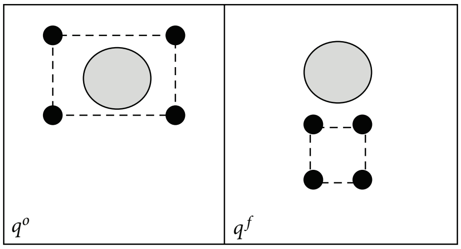


Figure 2: Geometry of the line-of-sight problem, required to compute ϕ_{ij}^{los}.

Definition 2 (Path-Loop Homotopic Equivalence). Given two paths $q_i(t)$ and $q_j(t)$, consider the loop resulting from the concatenation $\lambda_{ij} = [q_i(t)] \cdot [q_i(t^f) \to q_j(t^f)] \cdot [q_j(t)]^{-1} \cdot [q_j(t^o) \to q_i(t^o)]$ (see Figure 3). We call $q_i(t)$ and $q_j(t)$ path loop homotopic if and only if λ_{ij} is homotopic to the constant loop.

Remark 3. Later we will explore some connections between the definitions in this section and our control laws. Namely,

(1) our range potential produces motions equivilent to the straight line homotopy;

(2) the line-of-sight potential preserves path-loop homotopic equivalence;

(3) the pathloop homotopic equivalence condition suggests that a truly distributed controller is incapable of solving the problem for arbitrary initial conditions—either some "lead" robot must select a path class for the entire swarm or some type of bidirectional messaging must be used to reach a dynamic consensus on path-class selection.

5. Potential Functions

5.1. Range. The range constraint dictates that if $e_{ij} \in G^*$ then $d(q_i, q_j) \le \rho_{\max}$. This is enforced by a potential

$$\phi_{ij}^{\text{range}}\left(q_i, q_j\right) = \begin{cases} 0, & d\left(q_i, q_j\right) < \rho_{\max}, \\ \left(d\left(q_i, q_j\right) - \rho_{\max}\right)^2, & d\left(q_i, q_j\right) \ge \rho_{\max}. \end{cases} \tag{4}$$

We point out several observations.

(1) The potential only possesses minima at configurations where the range constraint is satisfied.

(2) The partial derivatives are antisymmetric: $\partial\phi_{ij}^{\text{range}}/\partial q_i = -\partial\phi_{ij}^{\text{range}}/\partial q_j$.

(3) ϕ^{range} is smooth and the partial derivatives are defined everywhere.

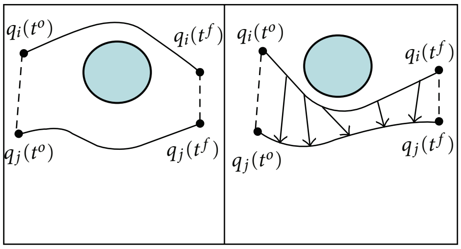


Figure 3: Frames depict an example of a combination of G^*, q^o, and q^f which are not feasible. Loops in left and right frames are not in the same homotopic equivalence class.

(4) The motion induced by $\dot{q}_i = -\partial\phi_{ij}^{\text{range}}/\partial q_i$ is essentially the straight-line homotopy and therefore preserves line of sight.

5.2. Line of Sight. If two robots q_i, q_j such that $e_{ij} \in E$, are in danger of losing line of sight, it means one of them is occluded from the other's view by an obstacle as seen in Figure 4. Consider the straight line connecting them. We call the closest parallel that does not intersect the obstacle, the *occlusion line*, OL. The line-of-sight constraint is enforced by a potential:

$$\phi_{ij}^{\text{los}}\left(q_i, q_j\right) = \begin{cases} 0 & \text{if L.O.S.} \\ d^2\left(q_i, \text{OL}\right) & \text{else,} \end{cases} \tag{5}$$

where $d(q_j, OL)$ denotes the distance from q_i to the occlusion line defined in the usual way.

We point out several observations.

(1) The potential only possesses minima at configurations where the line-of-sight constraint is satisfied.

(2) The partial derivatives are symmetric $\partial\phi_{ij}^{\text{los}}/\partial q_i = \partial\phi_{ij}^{\text{los}}/\partial q_j$.

(3) The motion induced by $u_i = u_j = -\partial\phi_{ij}^{\text{los}}/\partial q_i$ preserves path-loop homotopic equivalence.

(4) ϕ^{los} is continuous and its partial derivatives are defined almost everywhere (see Figure 4).

(5) The set of measure zero, along which ϕ^{los} is nonsmooth, is not in the basin of attraction of the induced motion.

To see why the third property holds, consider two paths that possess path-loop equivalence, such as in Figure 3 (right). Deforming the path-loop in such a way as to make it no longer equivalent to the constant loop requires forcing one of the lines connecting q_i and q_j to cross through the center of the obstacle (Figure 4) which requires *increasing* ϕ^{los}.

5.3. Goal Attainment and Obstacle Avoidance. In this paper we use Navigation Functions as the basis for ensuring the goal completion portion of the problem ($q \rightarrow q^f$). *Navigation Functions* are artificial potential fields that simultaneously provide obstacle avoidance and almost everywhere convergence to a goal configuration [36].

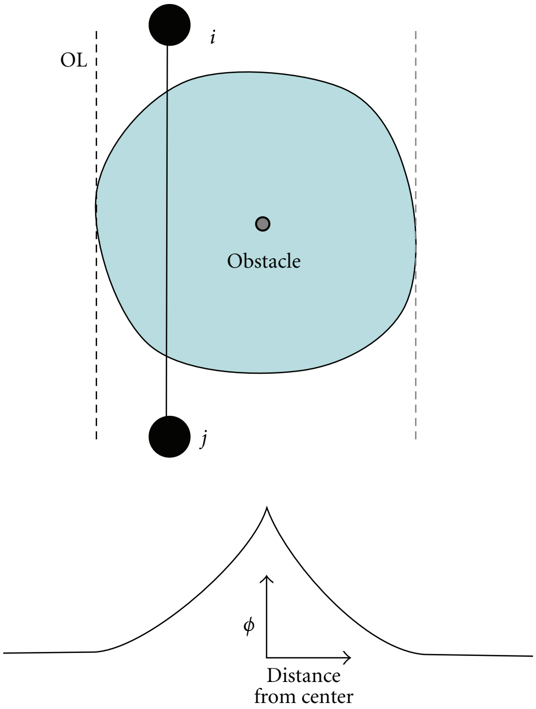


Figure 4: The left frame illustrates a situation where line of sight is not maintained (no straight-line homotopy between paths; loop is not homotopic to constant loop). The right frame shows two paths that maintain line of sight (straight-line homotopy between paths; loop is homotopic to the constant loop).

Definition 4 (Navigation Function). For robot i, a scalar map $\phi_i^{\text{goal}} : C_{\text{free}} \rightarrow [0, 1]$ is a *Navigation Function* if it is

(1) polar at q_i^f (i.e., has a unique minimum on the path connected component of C_{free} containing q_i^f);

(2) admissible on C_{free} (i.e., it is uniformly maximal on the boundary of C_{free});

(3) a Morse function (i.e., its Hessian is nonsingular at the critical points);

(4) smooth on C_{free} (i.e., at least C^2).

As an example consider that in the simplest case of circular obstacles in a circular workspace, a navigation function for robot i can be defined as

$$\phi_i^{\text{goal}}(q_i) = \frac{d^2\left(q_i, q_i^f\right)}{\left[d^k\left(q_i, q_i^f\right) + \prod_{j=0}^{M} d\left(q_i, O_j\right)\right]^{1/k}}, \tag{6}$$

where O_j is obstacle j, O_0 is the boundary of the workspace, and the parameter k must be selected high enough that all local minima, except at q_i^f, disappear.

Remark 5. A fundamental result from topology (the Morse Index Theorem [35]) states that it is impossible to derive a smooth potential function that has only one critical point (at the goal) when the workspace is multiply connected.

Therefore, any potential defined on a workspace with M obstacles will inevitably possess M saddle points. Emerging from each saddle point is a stable manifold connecting the saddle to other extrema. Initial positions on different sides of these manifolds evolve in different path loop homotopic equivalence classes around the obstacle associated with the saddle point. Therefore, if $e_{ij} \in G(q^0)$ but the line segment connecting $q^i(t^0)$ and $q^j(t^0)$ crosses the stable manifold, the necessary condition in Section 4 will not hold.

6. Parallel Composition

At any time, robot i must select a direction to move, which goes toward its goal position and avoid obstacles (ideally, $\dot{q}_i = -\partial\phi_i^{\text{goal}}/\partial q_i$), and for each corresponding edge e_{ij} in G^*, it must maintain range (ideally $\dot{q}_i = -\partial\phi_{ij}^{\text{range}}/\partial q_i$) and line of sight (ideally $\dot{q}_i = -\partial\phi_{ij}^{\text{los}}/\partial q_i$). Simply adding the three underlying potential functions does *not* guarantee that their underlying behavior is retained and may introduce local minima. In this section we explain how this disparate objectives are composed in parallel.

6.1. Theory. Consider a single robot i and a single associated scalar navigation function ϕ_i^{goal}. Typically, goal attainment and obstacle collision avoidance is achieved by the unique choice of input $u = -\nabla\partial\phi_i^{\text{goal}}$. However, we will see that there are an uncountable set of possible inputs that meet these objectives.

Theorem 6. *Given a Navigation function ϕ_i^{goal}, define a vector field $[\nabla\phi_i^{\text{goal}}]^{\perp}$ such that $\nabla\phi_i \cdot [\nabla\phi_i^{\text{goal}}]^{\perp} = 0$, for all q, with $[\nabla\phi_i^{\text{goal}}]^{\perp} = 0$ where $\nabla\phi_i^{\text{goal}} = 0$. Then the control law*

$$u_i^{\alpha} = -\alpha_1(t)\nabla\phi_i^{\text{goal}} + \alpha_2(t)\left[\nabla\phi_i^{\text{goal}}\right]^{\perp}, \tag{7}$$

also results in goal attainment and obstacle avoidance, for any function $\alpha : \mathbb{R}^+ \to \mathbb{R}^+ \times \mathbb{R}$.

The proof [37] proceeds in two parts. First we address goal attainment.

Proof. Observe that ϕ_i^{goal} satisfies all the criteria of a candidate Lyapunov function since it is continuous and positive definite. Next note that any given constant value of $\overline{\alpha}$ can be used to define a dynamic system that for which ϕ_i^{goal} is a Lyapunov function and q^f is an equilibrium, since it is strictly decreasing along system trajectories except at q^f

$$\begin{aligned}\dot{\phi}_i^{\text{goal}} &= \nabla\phi_i^{\text{goal}} \cdot u_i^{\alpha} \\ &= \nabla\phi_i^{\text{goal}} \cdot \left(-\alpha_1(t)\nabla\phi_i^{\text{goal}} + \alpha_2(t)\left[\nabla\phi_i^{\text{goal}}\right]^{\perp}\right) \\ &= \nabla\phi_i^{\text{goal}} \cdot \left(-\alpha_1\nabla\phi_i^{\text{goal}}\right) \le 0.\end{aligned} \tag{8}$$

Therefore, the equilibrium point, q^f must be asymptotically stable. This family of differential equations, parameterized by $\overline{\alpha}$, shares ϕ_i^{goal} as a *common Lyapunov Function*. It is known (see [38] or [39]) that a dynamic system whose derivative switches between such a family will be stabilized to the common equilibrium, regardless of the nature of the switching sequence. □

Next we show the obstacle avoidance property is preserved.

Proof. For any point on the boundary of the free space $q \in \partial O$ let $\hat{n}(q)$ be the unit normal pointing toward the interior of the free space. Then proving that the robot does not hit the obstacles is equivalent to proving $\dot{q}_i \cdot \hat{n}(q_i) \ge 0$, for all $q_i \in \partial O$. Using $\dot{q}_i = u_i^{\alpha}$ we have

$$\left(-\alpha_1\nabla\phi_i^{\text{goal}} + \alpha_2\left[\nabla\phi_i^{\text{goal}}\right]^{\perp}\right) \cdot \hat{n}. \tag{9}$$

Recall that navigation functions are uniformly maximal on the boundary of the free space, so $-\nabla\phi_i^{\text{goal}}(q)$ is *parallel* to $\hat{n}(q)$ for all $q \in \partial O$, so $-\nabla\phi_i^{\text{goal}}(q) \cdot \hat{n}(q) > 0$ and $[\nabla\phi_i^{\text{goal}}(q)]^{\perp} \cdot \hat{n}(q) = 0$. Therefore (9) reduces to

$$\left(-\alpha_1\nabla\phi^{\text{goal}} \cdot \hat{n}\right) \ge 0. \tag{10}$$

□

This proof suggests that we are able to select values of α online to satisfy other objectives, in a possibly discontinuous fashion, without destroying the goal attainment or obstacle avoidance behavior of the navigation function.

Remark 7. This fact that many vector fields can decrease the value of the potential was observed in [40, 41]; the set of all input vectors which decrease some cost-to-go function is termed the "cone of progress". However in both of these contexts the fact is used passively to address sensor uncertainty—rather than to explicitly construct a motion control law.

We now turn our attention to ϕ_{ij}^{range} and ϕ_{ij}^{los}.

Theorem 8. *Assume at time $t = 0$, that $\phi_{ij}^{range} = 0$ and $\phi_{ij}^{los} = 0$. Any control law u_i which satisfies both*

$$\frac{\partial}{\partial q_i}\phi_{ij}^{range} \cdot \left(u_i - u_j\right) \le 0, \tag{11}$$

$$\frac{\partial}{\partial q_i}\phi_{ij}^{los} \cdot \left(u_i + u_j\right) \le 0, \tag{12}$$

preserves the range and line of sight constraints for all t.

Proof. Thus

$$\dot{\phi}_{ij}^{\text{range}} = \frac{\partial}{\partial q_i}\phi_{ij}^{\text{range}} \cdot \dot{q}_i + \frac{\partial}{\partial q_j}\phi_{ij}^{\text{range}} \cdot \dot{q}_j. \tag{13}$$

However, $\dot{q}_i = u_i$ and the partial derivatives are antisymmetric, leading to the result above. The derivation of the second equation is analogous—except that the partial derivatives of the line-of-sight constraint are symmetric. □

6.2. Algorithm

Problem 2. Assume robot i knows $\partial\phi_i^{\text{goal}}/\partial q_i$, $\partial\phi_{ij}^{\text{los}}/\partial q_i$, $\partial\phi_{ij}^{\text{range}}/\partial q_i$, and u_j. Compute α to

$$\min\left(-(\alpha_1)^2 + (\alpha_2)^2\right), \tag{14}$$

such that

$$\begin{aligned} \frac{\partial}{\partial q_i}\phi_{ij}^{\text{range}} \cdot \left(u_i^{\alpha} - u_j\right) \leq 0, \\ \frac{\partial}{\partial q_i}\phi_{ij}^{\text{los}} \cdot \left(u_i^{\alpha} + u_j\right) \leq 0, \end{aligned} \tag{15}$$

with $0 \leq \alpha_1 \leq \alpha_1^{\max}$ and $-\alpha_2^{\max} \leq \alpha_2 \leq \alpha_2^{\max}$.

We call such a u_i^{α}, if it exists, a *feasible direction*. The concept of a feasible direction bears some relation to, and is named after, a numerical optimization method [42]; it is also loosely related to so-called null-space control methods [9].

Consider Figure 5. Geometric insight into the problem can be gained from recognizing that the range and line-of-sight constraints each define a cone in the velocity space. While the set of all u_i^{α} defines a half plane of possible velocity vectors.

If the intersection of the two cones and half plane is empty, no solution exists; and one of the constraints must be relaxed. If the intersection is not empty the formulation of the objective function reflects a preference toward rapid goal attainment (large α_1 and small α_2).

Computationally, this is a essentially a semidefinite programming problem, which can be solved in polynomial-time [43]. We use the *CVX* package, implemented in Matlab [44].

One attractive feature of the semidefinite programming approach used in *CVX* is that if the set of feasible direction, depicted in Figure 5 is not empty, the algorithm is guaranteed to find a solution in polynomial time. Because the problem is very small (2 dimensions and 2 constraints) numerical solution can be computed extremely rapidly.

In cases where the set of feasible directions is empty, simultaneously satisfying the three objectives is impossible. *CVX* will recognize this and report failure.

This approach is summarized in Algorithm 1.

6.3. Problem Structure and Completeness. So far we have show that all u^{α} satisfy the goal objective; and at each time step, if there is exists a value of α that meets the constraints, the numerical optimization algorithm is guaranteed to find it. Still the question remains: are there situations in which there is no value of α that will solve the problem? The answer is yes. As discussed in Remark 5, given any $e_{ij} \in G^*$ if the line connecting q_i^o and q_j^o, or q_i^f and q_j^f intersects a stable manifold of the Navigation Function's saddle points the problem is infeasible. Figure 6 graphically illustrates one such scenario, both robots are selecting infeasible directions as the Navigation function tries to force them into different path classes around the obstacle. When this occurs we drop one of the constraints until a feasible solution exists. This is discussed more in Section 7.

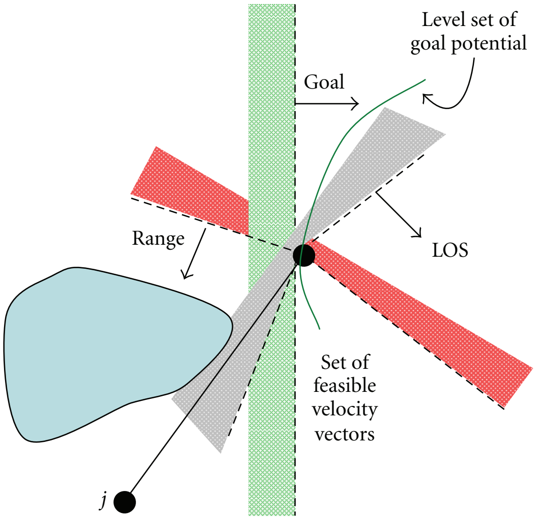


FIGURE 5: An illustration of the velocity selection problem for robot i. u^{α} is the set of all vectors in the same half plane as $-\partial\phi_i^{\text{goal}}/\partial q_i$ (labeled as *Goal*). Similarly, the range and line of sight constraints define cones in the tangent space. The set of feasible directions lies at the intersection of these sets.

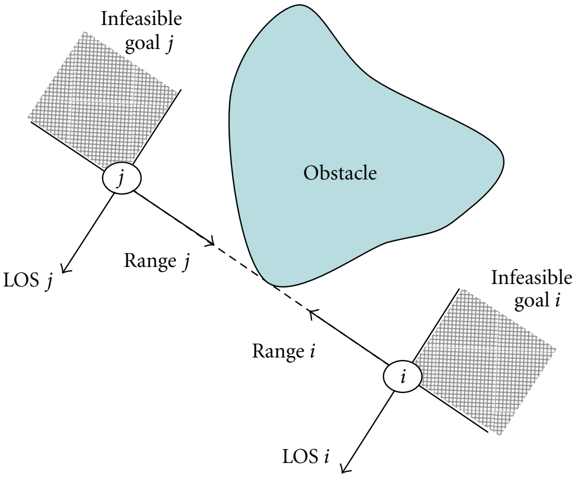


FIGURE 6: A generic pair of robots, with active line of sight and range constraints. Figure indicates possible directions of $-(\partial/\partial q)\phi^{\text{goal}}$ that could result in no feasible direction for simultaneously satisfying three objectives.

7. Experiments

The method is tested both in simulation and on a group of mobile robots.

7.1. Simulation. First, the method was simulated using Matlab (version 2010A) on a desktop PC. We created a 300 by 300 unit workspace, populated with random triangular obstacles, as seen in Figure 8. Red circles represent robots;

```
Let $t = 0$.
while $q_i \neq q_i^f$ do
    for $j$ such that there exists $e_{ij} \in G^*$ do
        Determine $q_j$ and $u_j$
        Compute $\partial\phi_{ij}^{\text{los}}/\partial q_i$ and $\partial\phi_{ij}^{\text{range}}/\partial q_i$
    end for
    Compute $\partial\phi_i^{\text{goal}}/\partial q_i$
    Compute $u^\alpha$ via numerical optimization
    if Feasible then
        Move. $t = t + \Delta t$
    else
        Infeasible. Drop a constraint or terminate.
    end if
    end while
```

Algorithm 1: Algorithm for generating a feasible motion direction for Robot i, using time steps Δt.

$n =$	Scenarios	Constrained?	Violations?	Infeassible?
5	50	36	2	4
12	50	41	4	3
20	50	47	6	9

Figure 7: Simulation results for connectivity algorithm.

green lines indicate wireless links; and blue stars are goal configurations.

For the first set of experiments 5 robots are required to maintain a "P-" shaped connectivity graph. For example see the configuration labeled "frame 1" in the lower left quadrant of Figure 8. The P shape was chosen because it includes both leaf vertices as well as a cycle. The maximum acceptable range $\rho_{\max}$ was chosen to be 50 units.

The robots are placed in randomly generated initial scenarios using a uniform distribution. In order to generate acceptable scenarios, each initial condition was tested for two criteria.

(1) Is the desired P-shaped connectivity graph a subset of the connectivity graph induced by the range and line of sight conditions?

(2) Are all of the robots inside the free space?

If the answer to either of these questions was "no", the condition was rejected and a new set of random positions was generated. This methodology is identical to the one used to generate samples for Monte-Carlo integration over non-rectangular domains.

Even though the simulation is done on a single PC, we try to replicate the typical information dependencies that would arise on a distributed multirobot system. Specifically, robot i was assumed to know:

(1) its own position q_i;

(2) the geometry of the obstacles and the workspace;

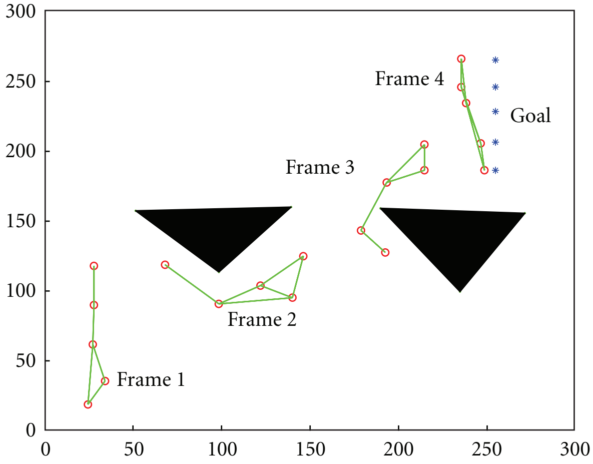


Figure 8: A swarm of 5 robots maintaining a specified communication graph, beginning in the lower left corner and moving toward the upper right corner.

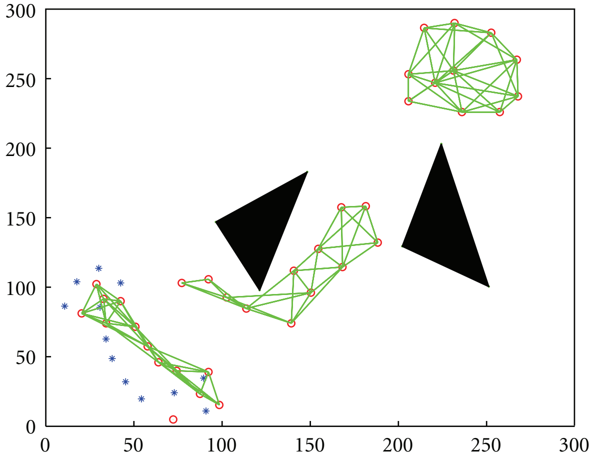


Figure 9: A swarm of 12 robots maintaining a specified communication graph. Beginning in the upper right corner and moving toward the lower left corner.

(3) the position and velocity, q_j and u_j, of any robots for which the range and line-of-sight criteria are met.

Regarding the first two assumptions, they are difficult to guarantee for physical implementation, but are none the less standard simulation assumptions in the literature. Regarding the third, recall that the motivation for designing this motion controller is to maintain communication links. Therefore, it is reasonable that those links could be used for peer-to-peer communication that is helpful for that purpose. Finally, we point out that the robots do *not* require all-to-all information exchanges of their position and velocity—the only global variable is time.

In the context of the simulation, the robots exchange the required information as permitted and recompute their velocities via numerical optimization every time step (0.1 second). Their top speed is 10 units/sec. Because there are no higher order dynamics, we use a forward Euler numerical integration method. To numerically solve the velocity selection problem at each time step, we used *CVX* Version 1.21, with the default *SeDuMi* solver and the precision option set to "low" (machine precision$^{1/4}$). The computation time for the optimization method was on average 0.032 sec per robot per time step, with a standard deviation of 0.008 sec. The computer we used ran Windows XP (32-bit), 1 GB RAM, and an AMD Athlon64 processor running at 2.2 GHz.

We selected one typical example scenario for illustration shown in Figure 8. As the robots move from frame 1 to frame 2, the leaf vertex's range to its neighbor is $\rho_{\max}$ and the range constraint is active. There is also an extra link that forms dynamically as two of the robots come with range. Between frames 2 and 3 and again from 3-4 the leaf node's line of sight constraint becomes active to prevent it from being occluded by the obstacle vertices. After frame 4, the robots proceed straight to the goal configuration at $t \approx 39$ sec. The goal configuration was selected to comply with the range and line-of-sight constraints dictated by the desired connectivity graph.

A second scenario illustrates this with a larger group of 12 robots and 15 desired links and is shown in Figure 9. Beginning in the upper right corner, the swarm constricts to fit through the gap between the two obstacles. The configuration snapshot in the middle of the figure shows an example of the many line-of-sight constraints that were activated as the robots negotiated the lower vertex of the left obstacle.

The table in Figure 7 summarizes the experiments. For each swarm size (n = 5, 12, and 20) we randomly generated 50 initial scenarios. The third column lists the number of scenarios that had active range or line-of-sight constraints. Many random scenarios do not exercise the constraints. For example, initial conditions that are very close to the goal may not require negotiating any obstacles. In other cases, the natural behavior of the navigation function generates motions that respect the range constraint.

The next column lists how many scenarios resulted in a minor constraint violation. We define a minor violation as a short duration (2 or fewer time steps), small magnitude $<(0.1\,\text{sec} \times 10\,\text{m/s})^2$ violations of (12) or (11). An illustration of a minor violation is included in Figure 10. The leaf vertex briefly violated the line-of-sight constraint. The primary reason this occurs is that velocity sharing effectively induces a delay, meaning that robot i uses robot j's velocity from the *previous* time step to compute its new velocity. This can lead to small violations of the constraints. As the time step gets smaller, or the maximum velocity decreases, the magnitude of these violations decreases. In practice, such a small constraint violation is unlikely to cause a wireless link to fail. However, the effect can be mitigated by including a small constant "buffer" in (12) or (11).

The final column lists the number of scenarios deemed infeasible by *CVX*. An illustration of an infeasible situation is included in Figure 11. One can see that robots 1, 4, and 5 start on the opposite side of the saddle point emanating from the left obstacle (depicted as the dotted red line) as compared with robots 2 and 3. Such scenarios, eventually lead to a situation in which the optimization routine determines the problem is infeasible. In such cases, we drop the offending constraints until a solution exists. They are later added back once the problem becomes feasible again and the robots are no longer separated by a saddle point. This is an unfortunate limitation of the approach proposed here.

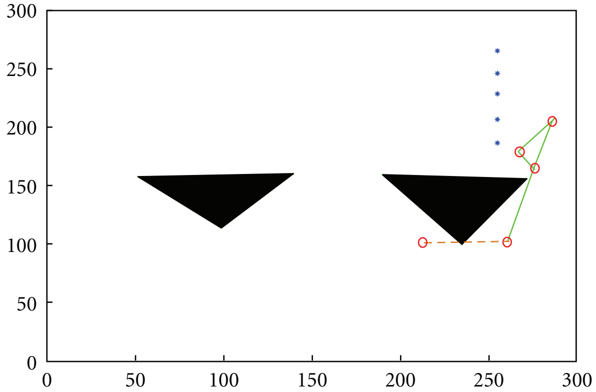


Figure 10: An illustration of a minor constraint violation.

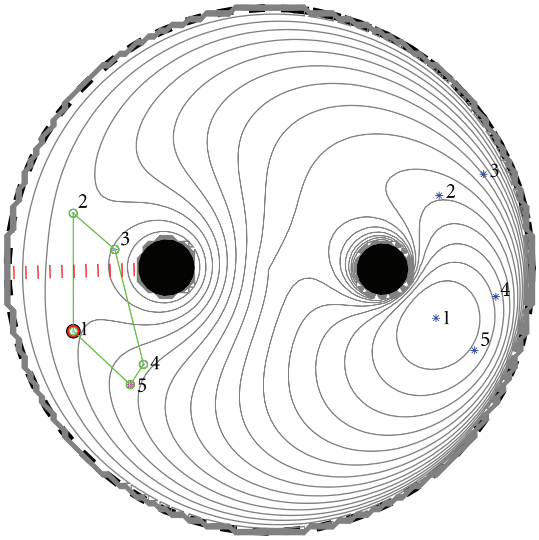
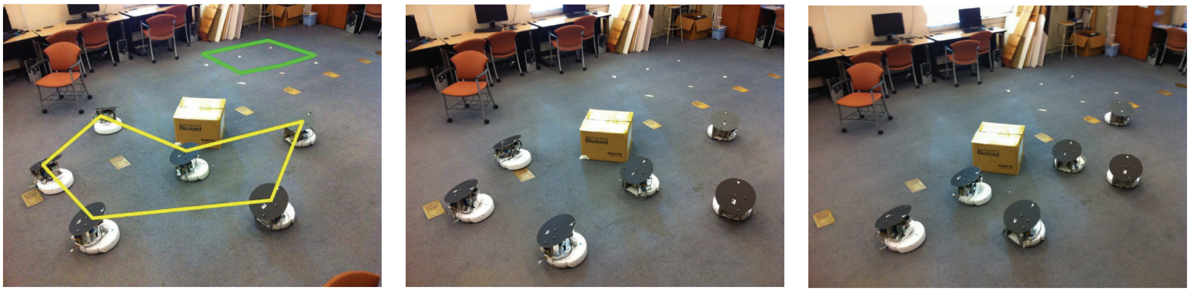


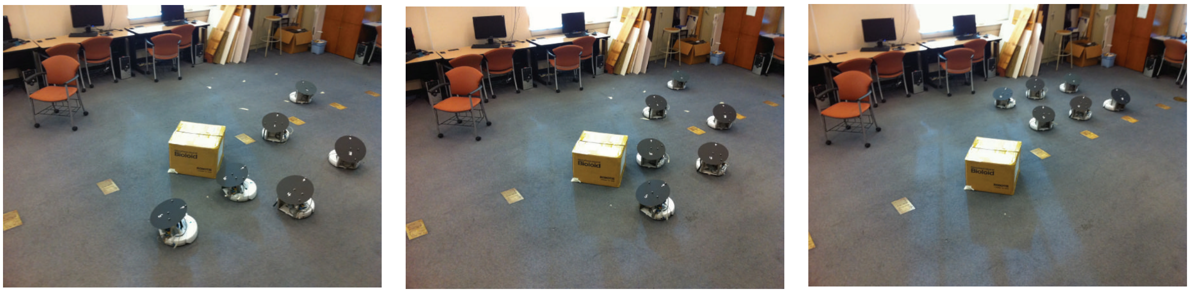
Figure 11: An illustration of an infeasible problem.

The algorithm was implemented on 6 iRobot Creates, shown in Figure 12. Each robot is controlled with an onboard netbook, using the Matlab Tool Box for the iRobot Create (MTIC) [45]. Their forward speed is limited to 0.2 meter/sec and use a time step of 0.5 seconds. They are given a manually made metric map of the environment, which is a 3 by 5 meter carpeted region. The obstacles are a 0.38 by 0.39 meter chair and a 0.6 by 0.6 meter cardboard box. We exploit their zero turn radius and use a backstepping controller (see our previous work [37] for details) to simulate the holonomic motions produced by the algorithm. The Minkowski sum of the robot's footprint and the obstacles is used to account for the size of the robots [36]. The robots were tagged with retroreflective fiducials and each robot obtained their positions from a 6 camera Vicon Motion Capture System, set up to broadcast simulated GPS NMEA messages. Each robot is also



(1) (2) (3)

Figure 12: Three sequential snapshots of the 6 iRobot Creates executing the motion planner presented in this paper. The first frame show the connectivity ring and the goal positions.



(4) (5) (6)

Figure 13: Three sequential snapshots of the 6 iRobot Creates executing the motion planner presented in this paper. Frame 6 is the goal position.

equipped with a Xbee radio modem to allow it to receive simulated GPS information as well as to exchange position and velocity with other robots. Due to the high power of the Xbee modules, and the small spatial scale of the experiment, signal strength tends to be high regardless of range or line of sight. Therefore, each robot's communication program is set to simulate these constraints, by only passing the exchanged information to the main program if they are met. Figures 12 and 13 show six snap shots of a feasible scenario where the range and line-of-sight constraints are active.

8. Conclusion

Motivated by the use of wireless communication among swarm members, in this paper we consider the problem of steering n robots to n goals, while maintaining some range and line-of-sight constraints in the presence of obstacles. Range and line of sight are two conditions which improve the reliability of wireless transmission. To the author's knowledge this is the first work to consider the effect of line-of-sight constraints for swarms. After establishing some basic conditions on the existence of solutions, we show that one rather profound condition is that all robots must pass on the same side of an obstacle (same path-class) for the swarm to remain connected. An implication of this is that, in order to remain connected, the swarm must either have a leader or some online method for achieving consensus on the path class. A further consequence of this is that navigation functions do not offer a global solution to this problem because the existence of saddle points makes it impossible to guarantee all robots select the same path class for arbitrary initial conditions. Basic potentials for Range and Line-of-Sight Constraints were introduced and a method for composing multiple potential functions into a single feasible motion direction was presented. An efficient computational algorithm to compute this direction is proposed. Simulations and hardware-based demonstrations of the algorithm's operation are provided and show promising results.

Acknowledgments

The author was supported by ONR Grant N0001405WRY20391. He wishes to thank Thomas Dunbar for help with the experimental results and numerous discussions.

References

[1] F. Zhang, D. M. Fratantoni, D. A. Paley, J. M. Lund, and N. E. Leonard, "Control of coordinated patterns for ocean sampling," *International Journal of Control*, vol. 80, no. 7, pp. 1186–1199, 2007.

[2] S. Thrun, "A probabilistic on-line mapping algorithm for teams of mobile robots," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 335–363, 2001.

[3] B. Jung and G. S. Sukhatme, "Tracking targets using multiple robots: the effect of environment occlusion," *Autonomous Robots*, vol. 13, no. 3, pp. 191–205, 2002.

[4] J. Esposito, M. Feemster, and E. Smith, "Cooperative manipulation on the water using a swarm of autonomous tugboats," in *Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA '08)*, pp. 1501–1506, Pasadena, Calif, USA, May 2008.

[5] J. M. Esposito, "Distributed grasp synthesis for swarm manipulation with applications to autonomous tugboats," in *Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA '08)*, pp. 1489–1494, Pasadena, Calif, USA, May 2008.

[6] J. Fink, M. A. Hsieh, and V. Kumar, "Multi-robot manipulation via caging in environments with obstacles," in *Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA '08)*, pp. 1471–1476, Pasadena, Calif, USA, May 2008.

[7] M. Horowitz and T. D. Murphey, "Adaptive cooperative manipulation with intermittent contact," in *Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA '08)*, pp. 1483–1488, Pasadena, Calif, USA, May 2008.

[8] M. Kloetzer and C. Belta, "Temporal logic planning and control of robotic swarms by hierarchical abstractions," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 320–330, 2007.

[9] B. E. Bishop, "Control of platoons of nonholonomic vehicles using redundant manipulator analogs," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4606–4611, Barcelona, Spain, April 2005.

[10] 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.

[11] 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. 863–868, 2007.

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

[13] D. P. Spanos and R. M. Murray, "Motion planning with wireless network constraints," in *Proceedings of the American Control Conference, (ACC '05)*, pp. 87–92, Portland, Ore, USA, June 2005.

[14] M. M. Zavlanos and G. J. Pappas, "Controlling connectivity of dynamic graphs," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 6388–6393, Orlando, Fla, USA, December 2005.

[15] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[16] C. W. Reynolds, "Flocks, birds and schools: a distributed behavioral model," *Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.

[17] J. H. Reif and H. Wang, "Social potential fields: a distributed behavioral control for autonomous robots," *Robotics and Autonomous Systems*, vol. 27, no. 3, pp. 171–194, 1999.

[18] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 905–908, 2001.

[19] V. Gazi and K. M. Passino, "Stability analysis of swarms," *IEEE Transactions on Automatic Control*, vol. 48, no. 4, pp. 692–697, 2003.

[20] H. G. Tanner, G. J. Pappas, and V. Kumar, "Leader-to-formation stability," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 443–455, 2004.

[21] P. Ogren, M. Egerstedt, and X. Hu, "A control Lyapunov function approach to multi-agent coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 847–851, 2002.

[22] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 813–825, 2002.

[23] R. Sepulchre, D. A. Paley, and N. E. Leonard, "Stabilization of planar collective motion with limited communication," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 706–719, 2008.

[24] M. Mesbahi, "On state-dependent dynamic graphs and their controllability properties," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 387–392, 2005.

[25] 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, 2006.

[26] M. M. Zavlanos and G. J. Pappas, "Potential fields for maintaining connectivity of mobile networks," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 812–816, 2007.

[27] J. M. Esposito and T. W. Dunbar, "Maintaining wireless connectivity constraints for swarms in the presence of obstacles," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 946–956, Orlando, Fla, USA, May 2006.

[28] Y. Mostofi, A. Gonzalez-Ruiz, A. Gaffarkhah, and D. Li, "Characterization and modeling of wireless channels for networked robotic and control systems—a comprehensive overview," in *Proceedings of the IEEE / RSJ International Conference on Intelligent Robots and Systems*, pp. 4849–4854, St. Louis, Mo, USA, October 2009.

[29] I. Rekleitis, V. Lee-Shue, A. Peng, and H. Choset, "Limited communication, multi-robot team based coverage," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3462–3467, New Orleans, La, USA, May 2004.

[30] R. Grabowski, P. Kholsa, and H. Choset, "Autonomous exploration via regions of interest," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2011–2016, Las Vegas, Nev, USA, October 2003.

[31] F. Dellaert, F. Alegre, and E. B. Martinson, "Intrinsic location and map making with two applications," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2344–2349, Taipei, Taiwan, September 2003.

[32] F. Belkhouche, B. Belkhouche, and P. Rastgoufard, "Line of sight robot navigation toward a moving goal," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 36, no. 2, pp. 255–267, 2006.

[33] R. C. Arkin and J. Diaz, "Line-of-sight constrained exploration for reactive multi-agent robotic teams," in *Proceedings of the 7th International Workshop on Advanced Motion Control, (AMC '02)*, Maribor, Slovenia, July 2002.

[34] S. O. Anderson, R. Simmons, and D. Golberg, "Maintaining line of sight communications networks between planetary rovers," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS '03)*, pp. 2266–2272, Las Vegas, Nev, USA, October 2003.

[35] J. R. Munkres, *Topology*, Prentice Hall, Englewood Cliffs, NJ, USA, 1975.

[36] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[37] J. M. Esposito and V. Kumar, "A method for modifying closed-loop motion plans to satisfy unpredictable dynamic

constraints at runtime," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1691–1696, Washington, DC, USA, May 2002.

[38] K. Narendra and J. Balakrishnan, "A common lyapunov function for stable LTI systems with commuting A matricies," *IEEE Transactions on Automatic Control*, vol. 39, no. 3, pp. 2469–2471, 1994.

[39] M. S. Branicky, *Studies in hybrid systems: modeling, analysis, and control*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Mass, USA, 1995.

[40] D. E. Koditschek, "The geometry of a robot programming language," in *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, vol. 3, pp. 263–268, 1994.

[41] M. Erdmann, "Understanding action and sensing by designing action-based sensors," *International Journal of Robotics Research*, vol. 14, no. 5, pp. 483–509, 1995.

[42] G. Vanderplaats, *Numerical Optimization Techniques for Engineering Design*, McGraw-Hill, New York, NY, USA, 1992.

[43] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, no. 1, pp. 49–95, 1996.

[44] M. Grant and S. Boyd, 2011, CVX: Matlab software for disci-plined convex programming, version 1.21, http://cvxr.com/cvx.

[45] J. M. Esposito and O. Barton, "Matlab toolbox for the irobot create," 2008, http://www.usna.edu/Users/weapsys/esposito/roomba.matlab/.

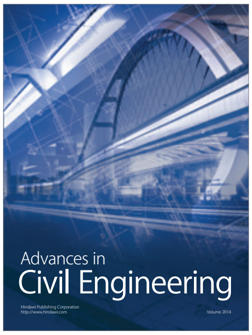
Journal of
Engineering

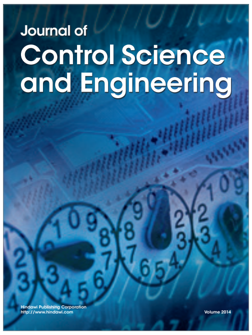
The Scientific
World Journal

International Journal of
Rotating
Machinery

Journal of
Sensors

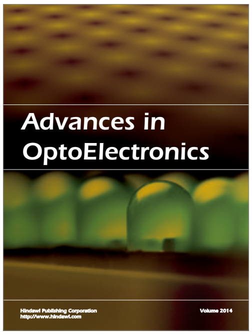
International Journal of
Distributed
Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

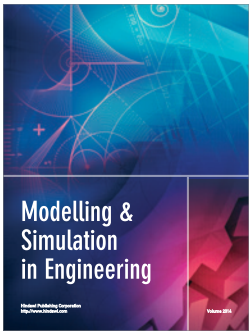
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

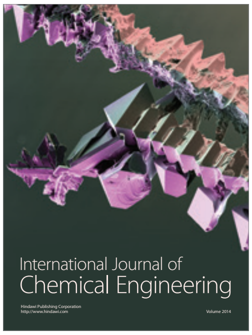

Advances in
OptoElectronics

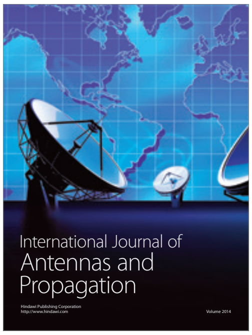
VLSI Design

International Journal of
Navigation and
Observation

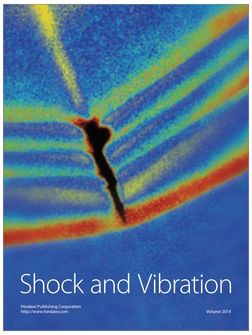

Modelling &
Simulation
in Engineering

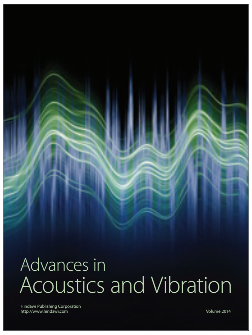
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration