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

Joel M. Esposito

Abstract— **The swarm paradigm of multi-robot cooperation relies on a distributed architecture, where each robot makes its own decisions based on locally available knowledge. Occasionally the swarm members may need to share information about their environment 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/receiever 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. An potential field-based algorithm is provided and its operation is demonstrated through both simulation and experimentation on a group of small robots.**

I. 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 [31], coordinated map making [27], and search [12]. Manipulating and transporting large objects is an application which can benefit from spatially distributed actuation [7], [5], [9], [11]. In contrast to centralized control methods, the swarm paradigm of multi-robot 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 [14], [1].

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 electro-magnetic 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 difficulty of reliably transmitting wireless messages through large dense obstacles. Together we term these two spatial constraints *Communication Constraints*. If a pair of robots meets these constraints, they can establish a *wireless link*. 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 pre-specified 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 III. We consider existence of solutions in Section IV and discuss a key attribute of any acceptable solution. In Section V we introduce potential fields for Goal, Range and Line of Sight objectives. Section VI 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 VII followed by concluding remarks in Section VIII.

II. 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.

J.M. Esposito is with the Department of Systems Engineering, United States Naval Academy, Annapolis, MD, 21401 USA e-mail: esposito@usna.edu.

Manuscript received Jan.1, 2011.

A. Notions of connectivity

Most discussions on connectivity within robot swarms employ a neighbor or proximity graph modeling paradigm [2], 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 [25], [21]; as an extension some works [24], [30] consider multi-hop connectivity, still using inter-robot 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.

B. Role of Connectivity in Consensus and Stability

All work on flocking and formation control relies on some underlying notion of swarm connectivity to prove stability and convergence. In general the primary objective of *flocking* can be though of as establishing a consensus on individual robot velocity vectors [22]. The work by Reynolds [19] 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. 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 [25], [21], [2]. 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 inter-robot ranges within a certain tolerance, similar to [18].

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 [4]. 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 [10], [26], [17], [3], [23]. In both formation and flocking works, while there is a dynamic relationship between consensus/stability and connectivity, the primary objective is the former and the later is a necessary condition.

C. Connectivity as a Primary Control Objective

There are several works that explicitly treat connectivity as the primary control objective of the swarm. They can be loosely divided into *optimization based* (or open loop) approaches and *feedback* approaches. In [15] 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 Semi-Definite Programming formulation. The work in [24] 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 to compute an open loop path.

An example of a feedback-based method is [13]. 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 [30]. A potential field based controller is used in [29]. A potential function is defined 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 [24] or [6], this approach does not specify individual edges to be maintained.

Like the works discussed above, the work presented here, and in our earlier work [6], explicitly treats connectivity as a primary control objective. Our notion of connectivity is also described using a neighbor graph framework, while a feedback control law attempts to maintain specific edges in the graph for all time. The principle difference between the work presented here and all the works presented above, is that we modify the local connection / edge formation criteria to consider line-of-site constraints as well as the widely used single-hop, range constraint.

III. 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

$$\dot{q}_i = u_i \tag{III.1}$$

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

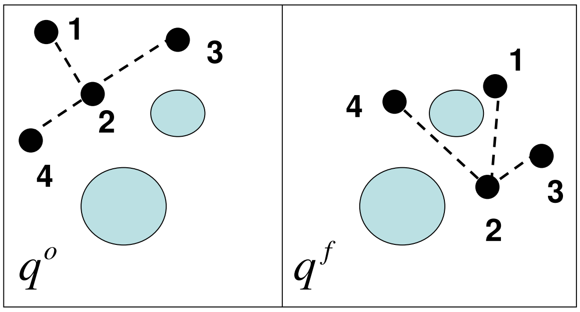

Fig. 1. The basic problem considered in this paper. Design a control law to guide the swarm from the initial position, q^o, to the desired final position, q^f, while maintaining desired communication links (Range plus Line-of-Sight).

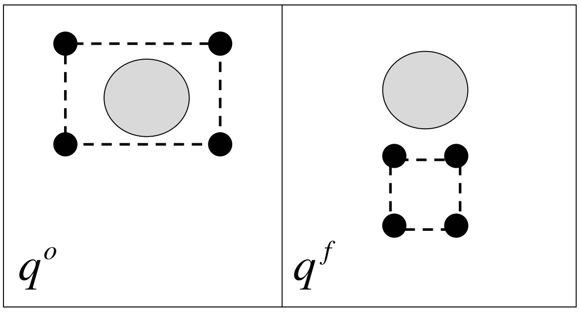

Fig. 2. Frames depict an example of a combination of G^*, q^o and q^f which are not feasible. Loops in left and right frames are not in the same homotopic equivalence class.

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; and
2) *Line-of-Sight*: $\exists x(\alpha) \in C_{free}$, $\forall \alpha \in [0, 1]$, such that $x(\alpha) = \alpha q_i + (1 - \alpha)q_j$,

note that d indicates distance as measured by the Euclidian metric. Both constraints model the power limitations of small wireless transmitters discussed in Sect. I. 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 3.1: 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 $\mathcal{U} : [t^o, t^f] \to U$ such that

1) $q(t^f) = q^f$ (*i.e., Goal* directed motion);
2) $G(q(t)) \supseteq G^*$, $\forall t \in [t^o, t^f]$ (*Line-of-Sight* and *Range*).

IV. 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 [16] is intimately related to these existence questions.

A. Homotopy Definitions

If $\tau_1, \tau_2 : [t^o, t^f] \to C_{free}$ are continuous maps (paths), we say that τ_1 and τ_2 are *homotopic* if there exists a continuous map $T : [t^o, t^f] \times [0, 1] \to C_{free}$ such that

$$T(t, 0) = \tau_1(t) \quad \text{(IV.1)}$$

$$T(t, 1) = \tau_2(t)\ , \forall t. \quad \text{(IV.2)}$$

If such a function exists, we say T is a homotopy. Note that we do not require the endpoints to remain fixed. This homotopy defines an equivalence relation on paths.

If $\tau(t^o) = \tau(t^f)$ the path is considered a loop λ. We can apply the homoptopy equivalence relation to loops as well. The trivial loop is the constant loop $\lambda(t) = \lambda(t^o)$, $\forall t$.

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

$$T(t, s) = (1 - s)\tau_1(t) + s\tau_2(t), \quad \text{(IV.3)}$$

due to its obvious connection to the line of sight constraint. If two paths τ_1, τ_2 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.

B. Intrinsic lack of solution

First note, that, in general, 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. 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 loop are not in the same homotopic equivalence class, it implies 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 4.1: To ensure the existence of solutions in this paper we only consider path-connected free spaces; and

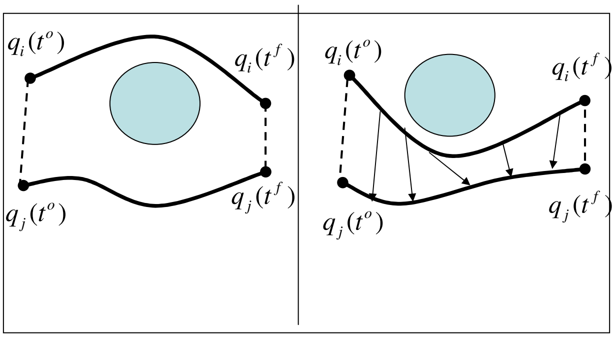

Fig. 3. The left frame illustrates a situation where Line-of-Sight is not maintained (no straight line homotopy between paths; loop is not homotopic to constant loop). The right frame shows two paths that maintain Line of Sight (straight line homotopy between paths; loop is homotopic to the constant loop).

q^0, q^f such that loops corresponding to any cycles of G^* are homotopic to the constant loop.

C. Attribute of Complete Solution Algorithms

As remarked earlier, in order to maintain an edge e_{ij}, $\forall t \in [t^o, t^f]$, 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– *i.e.* the swarm cannot "split".

Remark 4.2: An interesting interpretation of this is 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 bi-direction messaging must be used to reach a dynamic consensus on path-class selection.

This notion is difficult to formalize however. The traditional path-homotopy equivalence relation does not apply because the end points 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 (see Figure 3) consider the loop resulting from $\lambda_{ij} = [q_i(t)] \cdot [q_i(t^f) \to q_j(t^f)] \cdot [q_j(t)]^{-1} \cdot [q_j(t^o) \to q_i(t^o)]$. Requiring the paths $q_i(t)$ and $q_j(t)$ to be on the same "side" of the obstacle, means this loop is homotopic to (in the same equivalence class as) to the constant loop.

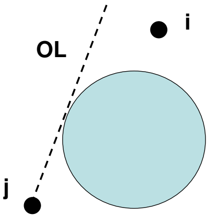

Fig. 4. Geometry of the line of sight problem, required to compute ϕ_i^{los}.

V. Potential functions

A. Range

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

$$\phi_{ij}^{range}(q_i, q_j) = \begin{cases} 0 & d(q_i, q_j) < \rho_{\max} \\ d^2(q_i, q_j) - \rho_{\max}^2 & d(q_i, q_j) \geq \rho_{\max}. \end{cases}$$

Note the potential only posses minima at configurations where the range constraint is satisfied, but is not strictly a navigation function.

B. Line-of-Sight

If two robots q_i, q_j such that $e_{ij} \in E$, are in danger of loosing line-of-sight, it means one of them is occluded from the other's view by an obstacle as seen in Fig. 4. The line connecting them at the last time when line of sight was satisfied is referred to as the *occlusion line*, OL (see Fig. 4). The line of sight constraint is enforced by a potential:

$$\phi_{ij}^{los}(q_i, q_j) = \begin{cases} 0 & if\ \ L.O.S. \\ d^2(q_i, OL) & else \end{cases} \tag{V.1}$$

where $d(q_j, OL)$ denotes the distance from q_i to the occlusion line defined in the usual way. Note the potential only possesses minima at configurations where the line-of-sight constraint is satisfied, but is not a proper Navigation Function.

C. Goal

In this paper we use Navigation Functions as the basis for ensuring the goal completion portion of the problem ($q \to q^f$). *Navigation Functions* are artificial potential fields that simultaneously provide obstacle avoidance and almost everywhere convergence to a goal configuration [20].

Definition 5.1: Navigation Function: For robot i, a scalar map $\phi_i^{goal} : C_{free} \to [0, 1]$ 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:

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

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. The selection of k makes it difficult to compute navigation function for dynamic work spaces.

Remark 5.2: It is known that workplaces with M obstacles will inevitably possess M saddle points. Note that 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 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 line of sight constraint between i and j may not be maintained.

VI. Parallel composition

At any time, q_i must select a direction to move, which goes toward its goal position (ideally, $\dot{q}_i = -\frac{\partial}{\partial q_i}\phi_i^{goal}(q_i)$), and for each corresponding edge e_{ij} in G^*, it must maintain range (ideally $\dot{q}_i = -\frac{\partial}{\partial q_i}\phi_{ij}^{range}(q_i, q_j)$) and line of sight (ideally $\dot{q}_i = -\frac{\partial}{\partial q_i}\phi_{ij}^{los}(q_i, q_j)$). However, it is not strictly necessary to use these traditional choices of motion directions.

A. Theory

Theorem 6.1: Any control law u (even a discontinuous one) which satisfies

$$\frac{\partial}{\partial q}\phi(q,t) \cdot u \le -\frac{\partial \phi(q,t)}{\partial t} \qquad \text{(VI.1)}$$

will stabilize the system to the minima of ϕ. The proof is straightforward and appears in [8].

A potential function, ϕ is essentially a Lyapunov function. Provided $\dot{\phi}(q(t)) \le 0$, $\forall t$ we can show q approaches the minima of the potential function, since a selection of u in accordance with the theorem produces:

$$\dot{\phi} = \frac{\partial}{\partial q}\phi(q,t) \cdot u + \frac{\partial \phi(q,t)}{\partial t} \le 0. \qquad \text{(VI.2)}$$

Said another way, all control laws that satisfy eq. VI.1 create vector fields which possess a Common Lyapunov Function, ϕ.

In the case of $\phi_i^{goal}(q_i)$, there is no explicit time dependence so the partial with respect to time in eq. VI.2 vanishes. In addition, it is uniformly maximal on the boundary of C_{free}, so its obstacle avoidance properties are preserved by all such u [8]. For functions such as $\phi^{los}(q_i, q_j)$ or $\phi^{range}(q_i, q_j)$ there are two ways to apply the theorem. First one could consider ϕ to have no explicit time dependence, but robots i and j are actively working in concert to achieve ϕ by each selecting u_i and u_j to decrease ϕ. In that case select

$$\frac{\partial}{\partial q_i}\phi(q_i, q_j) \cdot u_i \le 0 \qquad \text{(VI.3)}$$

$$\frac{\partial}{\partial q_j}\phi(q_i, q_j) \cdot u_j \le 0. \qquad \text{(VI.4)}$$

Alternatively, if only one of the robots is actively working to reduce ϕ then the second robot's influence can be viewed as an explicit time dependence

$$\dot{\phi} = \frac{\partial}{\partial q_i}\phi(q_i,t) \cdot u_i + \frac{\partial \phi(q_i,t)}{\partial t} \le 0. \qquad \text{(VI.5)}$$

Where the partial with respect to t can be determined if q_j's velocity is known. This information can be sent to robot i via the wireless link and ϕ can still be satisfied.

If Robot i desires to *compose in parallel* (*i.e.*, simultaneously satisfy) multiple objectives encoded by $\phi_1, \ldots, \phi_P$, u_i must satisfy

$$\frac{\partial}{\partial q_i}\phi_l(q_i,t) \cdot u_i \le -\frac{\partial \phi_l(q_i,t)}{\partial t} \qquad \forall l \in [1, P]$$

Such a u_i is called a *feasible direction*. It is possible to have multiple or no feasible directions.

Remark 6.2: In the case of two objectives $\phi_1(q)$ and $\phi_2(q)$, with no explicit time dependence, there always exists a feasible direction.

Remark 6.3: The concept of a feasible direction bears some relation to, and is named after, a numerical optimization method [28]; it is also loosely related to so-called null-space control methods [1].

B. Computation

We present an algorithm to compute a feasible direction in $\mathbb{R}^2$ if one exists and guarantee that it terminates in fixed number of steps. The algorithm is illustrated in Algorithm 1 and in Figure 5. Each input vector, $v_1, \ldots, v_P$, corresponds to the negative gradients of any relevant potentials robot i must consider. Typically for the problems considered here each robot will have one potential for the goal, and up to two additional potentials (Range and Line-of-Sight) per desired communication link (incident edge in G^*). Note that α is a arbitrary weighting factor that can be selected off-line. We use $\alpha = 0.5$ here. The notation *Rot* means to rotate a vector about an axis by an angle.

There are $P(P-1)/2$ possible cross products to compute. In $\mathbb{R}^2$ each requires 3 floating point operations. Therefore, in the worst case $3P(P-1)/2$ operations are required to test feasibility. If the problem is feasible an additional 16 floating point operations result in an answer for u_i. If the set of vectors has no feasible solution then some directions must be dropped

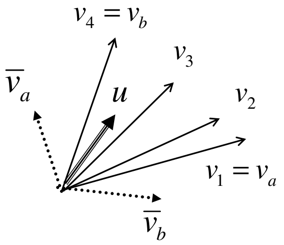

Fig. 5. Given a series of possible input directions $v_1, \ldots, v_P$, the algorithm synthesizes a movement direction u which is in the same half plane as all input vectors.

Algorithm 1 Algorithm for generating a feasible motion direction in $\mathbb{R}^2$

```
Function $u_i$ = Feasible Direction$(v_1, v_2, \ldots, v_P)$
if $\exists v_a | v_a \times v_j \cdot \hat{k} \geq 0, \forall j \neq a$ then
    Find $v_b | v_b \times v_l \cdot \hat{k} \leq 0, \forall l \neq b$
    $\bar{v}_a = Rot(\hat{k}, 90^o) \times v_a$
    $\bar{v}_b = Rot(\hat{k}, -90^o) \times v_b$
    $u_i = \alpha \bar{v}_a + (1 - \alpha)\bar{v}_b$, for some $\alpha \in [0, 1]$
else
    Infeasible; END
end if
```

until a feasible solution exists. A prioritization scheme can be used for this purpose, however in our experience this is generally not necessary for the class of problems we consider, as discussed in the following section.

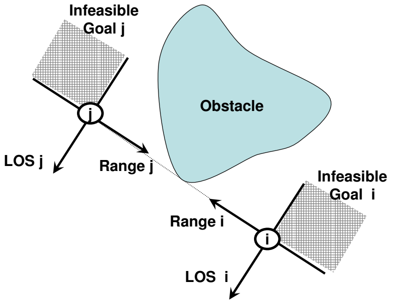

Fig. 6. A generic pair of robots, with active Line-of-Sight and Range constraints. Figure indicates possible directions of $-\frac{\partial}{\partial q}\phi^{goal}$ that could result in no feasible direction for simultaneously satisfying three objectives.

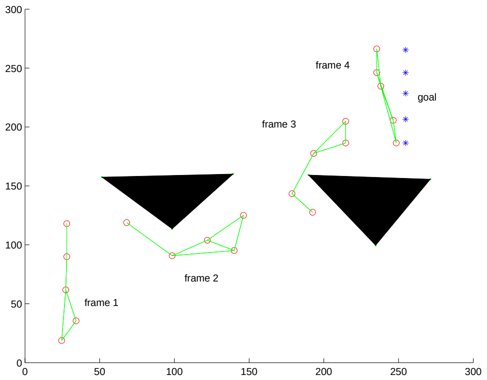

Fig. 7. A swarm of 5 robots maintaining a specified communication graph.

C. Problem Structure and completeness

Given an edge $e_{ij} \in G^*$, and ϕ_i^{goal} and ϕ_j^{goal}, a feasible direction always exits as long as the Range and Line-of-Sight constraints are not simultaneously active. This follows from the fact that given any two directions, a feasible direction always exists (Remark 5.2).

Guaranteeing the existence of a feasible direction when both Range and Line-of-Sight constraints are active is more difficult. The problem considered here has several special features worth noting.

Remark 6.4: The problem is planar.

Remark 6.5: If $\frac{\partial}{\partial q_i}\phi_i^{range}$ and $\frac{\partial}{\partial q_i}\phi_i^{los}$ are both nonzero, they are perpendicular to one another so their composition always has a feasible direction.

Remark 6.6: For two robots, i and j corresponding to the edge e_{ij}, $\frac{\partial}{\partial q_i}\phi_{ij}^{range} = -\frac{\partial}{\partial q_j}\phi_{ji}^{range}$ (anti-symmetric) and $\frac{\partial}{\partial q_i}\phi_{ij}^{los} = \frac{\partial}{\partial q_j}\phi_{ji}^{los}$ (symmetric).

In light of these observations and the discussion in Sect. VI-A, consider Figure 6. Given an edge $e_{ij} \in G^*$, provided $-\nabla\phi_i^{goal}$ *and* $-\nabla\phi_j^{goal}$ do not *both* lie in the region labeled infeasible, there always exists u_i and/or u_j which can simultaneously satisfy each robot's goal objectives as well as the Range and Line-of-Sight constraint for the pair. This can be verified graphically since for any goal direction outside the infeasible region, the goal, line-of-sight and range directions for a given robot all lie in the same half plane. And as long as one robot has a feasible direction, and knows the other robot's velocity the edge can be preserved. Further, given any $e_{ij} \in G^*$, provided: the lines connecting q_i^o and q_j^o, or q_i^f and q_j^f do not intersect a stable manifold (discussed in Remark 5.2); and the obstacles are convex, a feasible direction will always exists. This can be explained as follows. If both robots are selecting infeasible directions, it means they tend to separated by more than $\rho_{\max}$. Since the goal positions are separated by $\rho_{\max}$, the separation is not attributed to the difference in the fields. The separation behavior implies the points q_i^o and q_j^o intersects a stable manifold.

VII. Experiments

MATLAB simulations were used to demonstrate the algorithms operation. Figure 7 shows a swarm of 5 robots (red circles) maintaining the five initial edges shown in frame 1 as they move toward the goal configuration. Green lines indicate a wireless link. Figure 8 depicts the operation of the algorithm with 12 robots and 15 desired edges.

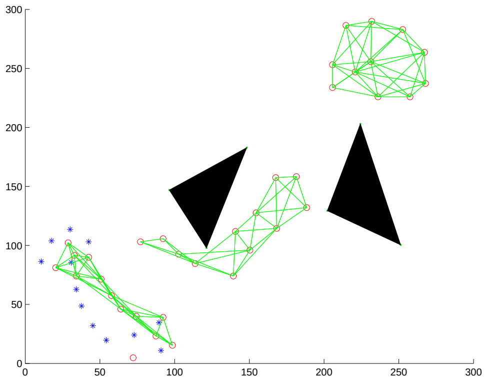

Fig. 8. A swarm of 12 robots maintaining a specified communication graph.

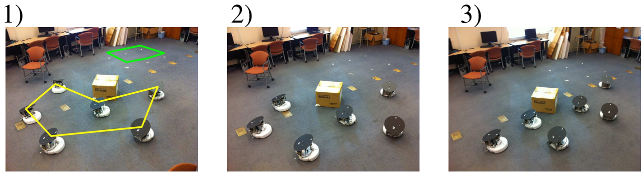

Fig. 9. Three sequential snapshots of the 6 iRobot Creates executing the motion planner presented in this paper. The first frame show the connectivity ring and the goal positions.

The above algorithm was implemented on 6 iRobot Creates, controlled using the Matlab Tool Box for the iRobot Create (MTIC) [?]. The robots were tagged with retro-reflective fiducials and obtained their position estimated from a 6 camera Vicon Motion Capture System. Each robot is also equipped with an Xbee radio modem to allow wireless communication. Figure 9-10 shows six snap shots of a scenario where the range and line of sight constraints are active.

VIII. 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 between them 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 on-line 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.

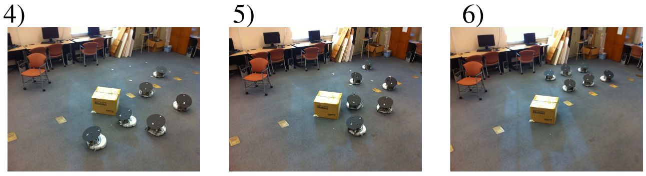

Fig. 10. Three sequential snapshots of the 6 iRobot Creates executing the motion planner presented in this paper. Frame 6 is the goal position

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] B.E. Bishop. Control of platoons of nonholonomic vehicles using redundant manipulator analogs. In *IEEE Conference on Robotics and Automation*, pages 4606–4611, 2005.

[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*, 51(8):1289 – 1298, 2006.

[3] 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*, 18(5):813–825, 2002.

[4] J.P. Desai, J.P. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic robots. *IEEE Transactions on Robotics and Automation*, 17(6):905–908, 2001.

[5] J.M. Esposito. Distributed grasp synthesis for swarm manipulation with applications to autonomous tugboats. In *IEEE Conference on Robotics and Automation*, 2008.

[6] J.M. Esposito and T.W. Dunbar. Maintaining wireless connectivity constraints for swarms in the presence of obstacles. In *IEEE Conference on Robotics and Automation*, pages 946–956, 2006.

[7] J.M. Esposito, M.G. Feemster, and E.T. Smith. Cooperative manipulation on the water using a swarm of autonomous tugboats. In *IEEE Conference on Robotics and Automation*, 2008.

[8] J.M. Esposito and V.Kumar. A method for modifying closed-loop motion plans to satisfy unpredictable dynamic constraints at runtime. In *IEEE Conference on Robotics and Automation*, pages 1691–1696, 2002.

[9] J. Fink, M.A. Hsieh, and V. Kumar. Multi-robot manipulation via caging in environments with obstacles. In *IEEE Conference on Robotics and Automation*, 2008.

[10] V. Gazi and K. M. Passino. Stability analysis of swarms. *IEEE Transactions on Automatic Control*, 48(4):692–696, 2003.

[11] M. Horowitz and T. Murphey. Adaptive cooperative manipulation with intermittent contact. In *IEEE Conference on Robotics and Automation*, 2008.

[12] B. Jung and G. Sukhatme. Tracking targets using multiple mobile robots: the effect of environment occlusion. *Autonomous Robots*, 13(2):191–205, 2002.

[13] Y. Kim and M. Mesbahi. On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian. *IEEE Transactions on Automatic Control*, 51(1):116–120, 2006.

[14] M. Kloetzer and C. Belta. Temporal logic planning and control of robotic swarms by hierarchical abstractions. *IEEE Transactions on Robotics and Automation*, 23(2):320–331, 2007.

[15] M. Mesbahi. On state-dependent dynamic graphs and their controllability properties. *IEEE Transactions on Automatic Control*, 50(3):387–392, 2005.

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

[17] M.Egerstedt P. Ogren and X. Hu.

[18] J.H. Reif and H. Wang. Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems*, 27:171 –194.

[19] C. Reynolds. Flocks, birds and schools: a distributed behavioral model. *Computer Graphics*, 21:25 – 34.

[20] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, 1992.

[21] R.Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

[22] R.Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[23] R. Sepulchre, D. Paley, and N.E. Leonard. Stabilization of planar collective motion with limited communication. *IEEE Transactions on Automatic Control*, 53(3):706–719, 2008.

[24] D. Spanos and R. Murray. Motion planning with wireless network constraints. In *American Control Conference*, 2005.

[25] H.G. Tanner, A. Jadbabaie, and G.J. Pappas. Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control*, 52(5):863–868, 2007.

[26] H.G. Tanner, G.J. Pappas, and V. Kumar. Leader-to-formation stability. *IEEE Transactions on Robotics and Automation*, 20(3).

[27] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363, 2001.

[28] G. Vanderplaats. *Numerical Optimization Techniques for Engineering Design*. McGraw–Hill, 1984.

[29] M.M. Zavlanos and G.J. Pappas. Potential fields for maintaining connectivity of mobile networks. *IEEE Transactions on Robotics*, 23(4):812.

[30] M.M. Zavlanos and G.J. Pappas. Controlling connectivity of dynamic graphs. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 6388 – 6393, 2005.

[31] F. Zhang, D.M. Fratanoni, D. Paley, J. Lund, and N.E. Leonard. Control of coordinated patterns for ocean sampling. *International Journal Of Control*, 80(7):1186 –1199, 2007.