ON ROBUST RENDEZVOUS FOR MOBILE AUTONOMOUS AGENTS

Sonia Martínez * **Jorge Cortés** ** **Francesco Bullo** *

* *Mechanical and Environmental Engineering, University of California, Santa Barbara, California 93106, USA*
{smartine,bullo}@engineering.ucsb.edu
** *Applied Mathematics and Statistics, University of California, Santa Cruz, California 95064, USA*
jcortes@ucsc.edu

Abstract: This paper presents coordination algorithms for networks of mobile autonomous agents. The objective of the proposed algorithms is to achieve rendezvous, that is, agreement over the location of the agents in the network. We provide analysis and design results for multi-agent networks in arbitrary dimensions under weak requirements on the switching and failing communication topology. The correctness proof relies on proximity graphs and their properties and on a LaSalle Invariance Principle for nondeterministic discrete-time systems.

Keywords: Networked control systems, rendezvous, proximity graphs, non-deterministic discrete-time dynamical systems

1. INTRODUCTION

This work is a contribution to the emerging discipline of motion coordination for ad-hoc networks of mobile autonomous agents. With this terminology we refer to groups of robotic agents with limited mobility and communication capabilities. In the future, groups of coordinated devices will perform a variety of tasks including search and recovery operations, surveillance, exploration and environmental monitoring. The potential advantages of employing arrays of agents have recently motivated vast interest in this topic. From a control viewpoint, a group of agents inherently provides robustness to failures of single agents or communication links.

The motion coordination problem for groups of autonomous agents is a control problem in the presence of communication constraints. Typically, each agent makes decisions based on partial information about the state of the entire network - obtained via communication with its immediate neighbors. An important difficulty is that the topology of the communication network depends on the agents' locations and, therefore, changes with the evolution of the network.

The "multi-agent rendezvous" problem and a first "circumcenter algorithm" have been introduced in (Ando *et al.*, 1999). This algorithm has been extended to various asynchronous strategies in (Lin *et al.*, 2004*b*; Lin *et al.*, 2004*a*; Flocchini *et al.*, 2001). A related algorithm, in which connectivity constraints are not imposed, is proposed in (Lin *et al.*, 2004*c*). These schemes are memoryless (static feedback), anonymous (all agents are indistinguishable), and spatially distributed (only local information is required). An incomplete list of recent works on motion coordination algorithms includes (Suzuki and Yamashita, 1999; Justh and Krishnaprasad, 2004) on pattern formation, (Klavins *et al.*, 2004) on self-assembly, (Liu and Passino, 2004) on foraging, and (Cortés *et al.*, 2004*b*) on deployment.

In this paper we provide novel analysis and design results on a class of rendezvous algorithms. First, we define and analyze a class of "circumcenter algorithms" defined over switching communication topologies. We classify communication topologies for our algorithms via the notion of "proximity graph," see (Jaromczyk and Toussaint, 1992) and (Cortés *et al.*, 2004*b*). Admissible topologies are proximity graphs being "spatially distributed" over the disk graph (i.e., they can be computed with the local information encoded in the disk graph) and such that their connected components have the same vertices as the disk graph (cf. Section 2.1). This is a more general class of communication topologies than those adopted in most works on coordination including (Ando *et al.*, 1999; Lin *et al.*, 2004*b*; Lin *et al.*, 2004*c*). The ability to rely on general topologies is advantageous in the design of wireless communication strategies and is referred to as "topology control", see (Li, 2003) and references therein.

Second, we consider networks of agents whose state space is $\mathbb{R}^d$, where $d \in \mathbb{N}$. We prove that our proposed class of circumcenter algorithms is correct in arbitrary dimensions and include simulations in two and three dimensions. As a natural outcome, we prove that the original circumcenter algorithm in (Ando *et al.*, 1999) can be adapted to work in higher dimensions, and that it is guaranteed to converge in finite time.

Third, we establish a general theorem on the robustness of the proposed class of circumcenter algorithms with respect to communication link failures. Rendezvous is guaranteed even if each agent experiences link failures, provided the resulting directed communication graph is strongly connected at least once every finite number of time instants. Our results provide the first contribution to the theoretical explanation of the robustness properties of the circumcenter algorithm observed in computer simulations in (Ando *et al.*, 1999). Because of length constraints, we refer the interested reader to (Cortés *et al.*, 2004*a*) for all the proofs. We only highlight that the (novel) method of proof is based on a recently-developed LaSalle Invariance Principle for nondeterministic discrete-time systems, see (Cortés *et al.*, 2004*b*).

2. PRELIMINARY DEVELOPMENTS

We review some notation for standard geometric objects; for additional information we refer to (de Berg *et al.*, 1997) and references therein. For a bounded set $S \subset \mathbb{R}^d$, $d \in \mathbb{N}$, we let $\mathrm{co}(S)$ denote the convex hull of S. For $p, q \in \mathbb{R}^d$, we let $]p, q[= \{\lambda p + (1-\lambda) q \mid \lambda \in]0,1[\}$ and $[p, q] = \mathrm{co}(\{p, q\})$ denote the *open* and *closed segment* with extreme points p and q, respectively. For a bounded set $S \subset \mathbb{R}^d$, we let $\mathrm{CC}(S)$ and $\mathrm{CR}(S)$ denote the *circumcenter* and *circumradius* of S, respectively, that is, the center and radius of the smallest-radius d-sphere enclosing S. For $p \in \mathbb{R}^d$, we let $B(p, r)$ and $\overline{B}(p, r)$ denote the *open* and *closed ball* of radius $r \in \mathbb{R}_+$ centered at p, respectively. Here, we let $\mathbb{R}_+$ and $\overline{\mathbb{R}}_+$ denote the positive and the nonnegative real numbers, respectively. A *polytope* is the convex hull of a finite point set. We let $\mathrm{Ve}(Q)$ denote the set of vertices of a polytope Q, and we emphasize that any vertex of Q is strictly convex.

Proposition 1. Let $S \subset \mathbb{R}^d$ be a polygon. Then

- (i) $\mathrm{CC}(S) \in \mathrm{co}(S) \setminus \mathrm{Ve}(\mathrm{co}(S))$;
- (ii) if $p \in S \setminus \mathrm{CC}(S)$ and $r \in \mathbb{R}_+$ satisfy $S \subset \overline{B}(p, r)$, then $]p, \mathrm{CC}(S)[$ has nonempty intersection with $\overline{B}(\frac{p+q}{2}, \frac{r}{2})$ for all $q \in S$.

2.1 Proximity graphs and their properties

We introduce some concepts regarding proximity graphs for point sets in $\mathbb{R}^d$. We assume the reader familiar with the standard notions of graph theory as defined in (Diestel, 2000, Chapter 1). Given $\mathbb{V}$, let $\mathbb{F}(\mathbb{V})$ be the collection of finite subsets of $\mathbb{V}$. We denote an element of $\mathbb{F}(\mathbb{R}^d)$ by $\mathcal{P} = \{p_1, \dots, p_n\} \subset \mathbb{R}^d$, where $p_1, \dots, p_n$ are distinct points in $\mathbb{R}^d$. Let $\mathbb{G}(\mathbb{R}^d)$ be the set of undirected graphs whose vertex set is an element of $\mathbb{F}(\mathbb{R}^d)$. A *proximity graph function* $\mathcal{G} : \mathbb{F}(\mathbb{R}^d) \to \mathbb{G}(\mathbb{R}^d)$ associates to a point set $\mathcal{P}$ an undirected graph with vertex set $\mathcal{P}$ and edge set $\mathcal{E}_{\mathcal{G}}(\mathcal{P})$, where $\mathcal{E}_{\mathcal{G}} : \mathbb{F}(\mathbb{R}^d) \to \mathbb{F}(\mathbb{R}^d \times \mathbb{R}^d)$ is such that $\mathcal{E}_{\mathcal{G}}(\mathcal{P}) \subseteq \mathcal{P} \times \mathcal{P} \setminus \mathrm{diag}(\mathcal{P} \times \mathcal{P})$ for any $\mathcal{P}$. Here, $\mathrm{diag}(\mathcal{P} \times \mathcal{P}) = \{(p, p) \in \mathcal{P} \times \mathcal{P} \mid p \in \mathcal{P}\}$. In other words, the edge set depends on the location of the vertices. Examples include the complete graph and the Euclidean Minimum Spanning Tree $\mathcal{G}_{\mathrm{EMST}}$. The following examples are defined in (de Berg *et al.*, 1997; Jaromczyk and Toussaint, 1992; Cortés *et al.*, 2004*b*):

- (i) the r-disk graph $\mathcal{G}_{\mathrm{disk}}(r)$, for $r \in \mathbb{R}_+$, with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{disk}}(r)}(\mathcal{P})$ if $\|p_i - p_j\| \leq r$;
- (ii) the Delaunay graph $\mathcal{G}_{\mathrm{D}}$, with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{D}}}(\mathcal{P})$ if the Voronoi regions of p_i and p_j have non-empty intersection;
- (iii) the Relative Neighborhood graph $\mathcal{G}_{\mathrm{RN}}$, with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{RN}}}(\mathcal{P})$ if, for all $p_k \in \mathcal{P} \setminus \{p_i, p_j\}$, $p_k \notin B(p_i, \|p_i - p_j\|) \cap B(p_j, \|p_i - p_j\|)$;
- (iv) the Gabriel graph $\mathcal{G}_{\mathrm{G}}$, with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{G}}}(\mathcal{P})$ if, for all $p_k \in \mathcal{P} \setminus \{p_i, p_j\}$, $p_k \notin B(\frac{p_i+p_j}{2}, \frac{\|p_i-p_j\|}{2})$.

If needed, we write $\mathcal{G}_{\mathrm{disk}}(\mathcal{P}, r)$ to denote $\mathcal{G}_{\mathrm{disk}}(r)$ at $\mathcal{P}$. We will also consider the proximity graphs $\mathcal{G}_{\mathrm{RN}\,\cap\mathrm{disk}}(r)$ and $\mathcal{G}_{\mathrm{G}\,\cap\mathrm{disk}}(r)$ defined by the intersection of $\mathcal{G}_{\mathrm{RN}}$ and $\mathcal{G}_{\mathrm{G}}$ with $\mathcal{G}_{\mathrm{disk}}(r)$, $r \in \mathbb{R}_+$, respectively. A different proximity graph related to, but different from, the intersection $\mathcal{G}_{\mathrm{D}\,\cap\mathrm{disk}}(r)$ of $\mathcal{G}_{\mathrm{D}}$ with $\mathcal{G}_{\mathrm{disk}}(r)$ is the r-limited Delaunay graph $\mathcal{G}_{\mathrm{LD}}(r)$ (see (Cortés *et al.*, 2004*b*)).

To each proximity graph function $\mathcal{G}$, one can associate the *set of neighbors map* $\mathcal{N}_{\mathcal{G}} : \mathbb{R}^d \times$

$\mathbb{F}(\mathbb{R}^d) \to \mathbb{F}(\mathbb{R}^d)$, defined by

$$\mathcal{N}_{\mathcal{G}}(p, \mathcal{P}) = \{q \in \mathcal{P} \mid (p, q) \in \mathcal{E}_{\mathcal{G}}(\mathcal{P} \cup \{p\})\}.$$

Given $p \in \mathbb{R}^d$, it is convenient to define $\mathcal{N}_{\mathcal{G},p} : \mathbb{F}(\mathbb{R}^d) \to \mathbb{F}(\mathbb{R}^d)$ by $\mathcal{N}_{\mathcal{G},p}(\mathcal{P}) = \mathcal{N}_{\mathcal{G}}(p, \mathcal{P})$. Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two proximity graph functions. We say that $\mathcal{G}_1$ is *spatially distributed over* $\mathcal{G}_2$ if,

$$\mathcal{N}_{\mathcal{G}_1,p}(\mathcal{P}) = \mathcal{N}_{\mathcal{G}_1,p}\big(\mathcal{N}_{\mathcal{G}_2,p}(\mathcal{P})\big) \text{ for all } p \in \mathcal{P}.$$

It is clear that if $\mathcal{G}_1$ is spatially distributed over $\mathcal{G}_2$, then $\mathcal{G}_1$ is a subgraph of $\mathcal{G}_2$, that is, $\mathcal{G}_1(\mathcal{P}) \subset \mathcal{G}_2(\mathcal{P})$ for all $\mathcal{P} \in \mathbb{F}(\mathbb{R}^d)$. The converse is in general not true (see (Cortés *et al.*, 2004*b*)). We conclude this section with some examples of proximity graphs in $\mathbb{R}^2$ and $\mathbb{R}^3$; see Figs 1 and 2.

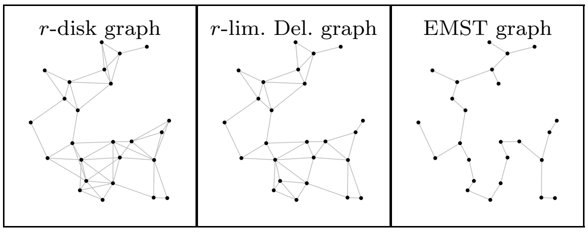

Fig. 1. From left to right, r-disk, r-limited Delaunay, and Euclidean Minimum Spanning Tree graphs in $\mathbb{R}^2$ for a configuration of 25 agents with coordinates uniformly randomly generated within the square $[-7, 7] \times [-7, 7]$. The parameter r is taken equal to 4.

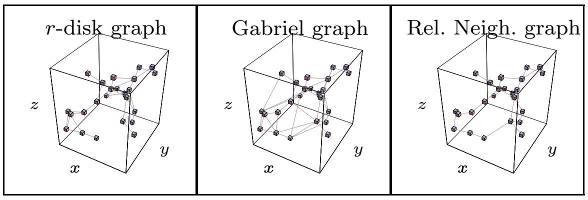

Fig. 2. From left to right, r-disk, Gabriel, and Relative Neighborhood graphs in $\mathbb{R}^3$ for a configuration of 25 agents with coordinates uniformly randomly generated within the square $[-7, 7] \times [-7, 7] \times [-7, 7]$. The parameter r is taken equal to 4.

2.2 Spatially distributed maps over proximity graphs

The notion of proximity graph is defined for sets of distinct points $\mathcal{P} = \{p_1, \dots, p_n\}$. However, we will also consider ordered sets of possibly coincident points, $P = (p_1, \dots, p_n) \in \mathbb{R}^d$. Let $i_{\mathbb{F}} : (\mathbb{R}^d)^n \to \mathbb{F}(\mathbb{R}^d)$ be the natural immersion, i.e., $i_{\mathbb{F}}(P)$ only contains only the distinct points in $P = (p_1, \dots, p_n)$. Note that $i_{\mathbb{F}}$ is invariant under permutations of its arguments and that the cardinality of $i_{\mathbb{F}}(p_1, \dots, p_n)$ is in general less than or equal to n. In what follows, $\mathcal{P} = i_{\mathbb{F}}(P)$ always denotes the point set associated to $P \in (\mathbb{R}^d)^n$.

We can now extend the notion of proximity graphs to this setting. Given a proximity graph function $\mathcal{G}$ with edge set function $\mathcal{E}_{\mathcal{G}}$, we define

$$\mathcal{G} = \mathcal{G} \circ i_{\mathbb{F}} : (\mathbb{R}^d)^n \to \mathbb{G}(\mathbb{R}^d),$$
$$\mathcal{E}_{\mathcal{G}} = \mathcal{E}_{\mathcal{G}} \circ i_{\mathbb{F}} : (\mathbb{R}^d)^n \to \mathbb{F}(\mathbb{R}^d \times \mathbb{R}^d).$$

We define the set of neighbors map $\mathcal{N}_{\mathcal{G}} : (\mathbb{R}^d)^n \to (\mathbb{F}(\mathbb{R}^d))^n$ as the function whose jth component is

$$\mathcal{N}_{\mathcal{G},j}(p_1, \dots, p_n) = \mathcal{N}_{\mathcal{G}}(p_j, i_{\mathbb{F}}(p_1, \dots, p_n)).$$

Note that coincident points in the tuple $(p_1, \dots, p_n)$ will have the same set of neighbors.

Given a set Y and a proximity graph function $\mathcal{G}$, a map $T : (\mathbb{R}^d)^n \to Y^n$ is *spatially distributed over* $\mathcal{G}$ if there exists $\tilde{T} : \mathbb{R}^d \times \mathbb{F}(\mathbb{R}^d) \to Y$, such that, for all $(p_1, \dots, p_n) \in (\mathbb{R}^d)^n$ and for all $j \in \{1, \dots, n\}$,

$$T_j(p_1, \dots, p_n) = \tilde{T}(p_j, \mathcal{N}_{\mathcal{G},j}(p_1, \dots, p_n)),$$

where T_j denotes the jth-component of T. In other words, the jth component of a spatially distributed map at $(p_1, \dots, p_n)$ can be computed with only the knowledge of the vertex p_j and the neighboring vertices in the graph $\mathcal{G}(\{p_1, \dots, p_n\})$.

3. RENDEZVOUS VIA PROXIMITY GRAPHS

3.1 Modeling a network of robotic agents

We introduce the notions of *robotic agent* and of *network of robotic agents*. Let n be the number of agents. The ith agent has a processor with the ability of allocating and operating on continuous and discrete states. It occupies a location $p_i \in \mathbb{R}^d$, $d \in \mathbb{N}$, and is capable of moving at any time $m \in \mathbb{N}$, for a unit period of time, according to

$$p_i(m + 1) = p_i(m) + u_i. \tag{1}$$

The control u_i takes values in a bounded subset of $\mathbb{R}^d$. We assume there is a maximum step size $s_{\mathrm{m}} \in \mathbb{R}_+$ common to all agents, $\|u_i\| \leq s_{\mathrm{m}}$, for all $i \in \{1, \dots, n\}$. The processor of each agent has access to its location, and transmits this information to any other agent within a closed disk of radius $r \in \mathbb{R}_+$. The communication radius is the same for all agents.

3.2 The rendezvous motion coordination problem

We now state the control design problem for the network of robotic agents. The *rendezvous objective* is to achieve agreement over the location of the agents in the network, that is, to steer each agent to a common location. This objective is to be achieved with the limited information flow described in Section 3.1. Typically, it will be impossible to solve the rendezvous problem if the agents are placed in such a way that they do not form a connected communication graph. Arguably, a good property of any algorithm for rendezvous is that of maintaining some form of connectivity between agents.

3.3 The Circumcenter Algorithm

Here is an informal description of the *Circumcenter Algorithm over a proximity graph* $\mathcal{G}$:

> Each agent performs the following tasks: (i) it detects its neighbors according to $\mathcal{G}$; (ii) it computes the circumcenter of the point set comprised of its neighbors and of itself, and (iii) it moves toward this circumcenter while maintaining connectivity with its neighbors.

This algorithm is an extension of the one introduced in (Ando *et al.*, 1999). Let us clarify two which proximity graphs are allowable and how connectivity is maintained. First, we are allowed to design algorithms that are spatially distributed over the r-disk graph $\mathcal{G}_{\text{disk}}(r)$, or more generally, over any proximity graph $\mathcal{G}$ that is spatially distributed over $\mathcal{G}_{\text{disk}}(r)$. This is a consequence of our modeling assumption that each agent can acquire the location of each other agent within distance less than or equal to r. Second, we maintain connectivity by restricting the allowable motion of each agent. If agents p_i and p_j are neighbors in $\mathcal{G}$, then their subsequent positions are required to belong to $\overline{B}(\frac{p_i+p_j}{2}, \frac{r}{2})$. If agent p_i has its neighbors at locations $\{q_1, \dots, q_l\}$, then its *constraint set* is

$$C_{p_i,r}(\{q_1,\dots,q_l\}) = \bigcap_{q\in\{q_1,\dots,q_l\}} \overline{B}\Big(\frac{p_i+q}{2}, \frac{r}{2}\Big).$$

Finally, for q_0 and q_1 in $\mathbb{R}^d$, and for a convex closed set $Q \subset \mathbb{R}^d$ with $q_0 \in Q$, let $\lambda(q_0, q_1, Q)$ denote the solution of the strictly convex problem:

$$\begin{aligned} &\text{maximize } \lambda \\ &\text{subject to } \lambda \le 1,\ (1-\lambda)q_0 + \lambda q_1 \in Q. \end{aligned} \tag{2}$$

This convex optimization problem has the following interpretation: move along the segment from q_0 to q_1 the maximum possible distance while remaining in Q. Under the stated assumptions the solution exists and is unique. We are now ready to formally describe the algorithm.

Name:	Circumcenter Algorithm over $\mathcal{G}$
Goal:	Solve the rendezvous problem
Assumes:	(i) $s_{\text{m}} \in \mathbb{R}_+$ maximum step size (ii) $r \in \mathbb{R}_+$ communication radius (iii) $\mathcal{G}$ spatially distributed proximity graph over $\mathcal{G}_{\text{disk}}(r)$

Agent $i \in \{1, \dots, n\}$ executes at each time instant in $\mathbb{N}$:

1: acquire $\{q_1, \dots, q_k\} := \mathcal{N}_{\mathcal{G}_{\text{disk}}(r),p_i}(\mathcal{P})$
2: compute $\mathcal{M}_i := \mathcal{N}_{\mathcal{G},p_i}(\{q_1,\dots,q_k\}) \cup \{p_i\}$
3: compute $Q_i := C_{p_i,r}(\mathcal{M}_i \setminus \{p_i\}) \cap \overline{B}(p_i, s_{\text{m}})$
4: compute $\lambda_i^* := \lambda(p_i, \text{CC}(\mathcal{M}_i), Q_i)$
5: set $u_i := \lambda_i^*(\text{CC}(\mathcal{M}_i) - p_i)$, i.e.,
move from p_i to $(1-\lambda_i^*)p_i + \lambda_i^*\,\text{CC}(\mathcal{M}_i)$

In what follows we refer to the Circumcenter Algorithm over $\mathcal{G}$ as $T_{\mathcal{G}} : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n$.

3.4 Correctness of the Circumcenter Algorithm

We now state the main convergence result, whose proof is provided in (Cortés *et al.*, 2004*a*).

Theorem 2. Let $p_1, \dots, p_n$ be a network of robotic agents in $\mathbb{R}^d$, for $d \in \mathbb{N}$, with maximum step size $s_{\text{m}} \in \mathbb{R}_+$ and communication radius $r \in \mathbb{R}_+$. Let the proximity graph $\mathcal{G}$ be spatially distributed over $\mathcal{G}_{\text{disk}}(r)$ and have the same connected components as $\mathcal{G}_{\text{disk}}(r)$. Any trajectory $\{P_m\}_{m\in\mathbb{N}\cup\{0\}}$ of $T_{\mathcal{G}}$ has the following properties:

(i) if the locations of two agents belong to the same connected component of $\mathcal{G}_{\text{disk}}(P_k, r)$ for some $k \in \mathbb{N}\cup\{0\}$, then they remain in the same connected component of $\mathcal{G}_{\text{disk}}(P_m, r)$ for all $m \ge k$;
(ii) there exists $P^* = (p_1^*, \dots, p_n^*) \in (\mathbb{R}^d)^n$ with the following properties: $P_m \to P^*$ as $m \to +\infty$, and $p_i^* = p_j^*$ or $\|p_i^* - p_j^*\| > r$ for each $i, j \in \{1, \dots, n\}$;
(iii) if $\mathcal{G} = \mathcal{G}_{\text{disk}}(r)$, then there exists $k \in \mathbb{N}$ such that $P_m = P^*$ for all $m \ge k$, that is, convergence is achieved in finite time.

A consequence of Theorem 2(i) and (ii) is that, if the locations of two agents belong to the same connected component of $\mathcal{G}$ at some time, then they converge to the same point in $\mathbb{R}^d$. The statements Theorem 2(i) and (ii) were originally proved in (Ando *et al.*, 1999) for the Circumcenter Algorithm over $\mathcal{G}_{\text{disk}}$ and for $d = 2$.

3.5 Robustness of the Circumcenter Algorithm

Here we characterize the robustness of the Circumcenter Algorithm with respect to link failures.

Definition 3. A *link failure in* $\mathcal{G}_{\text{disk}}(r)$ *at* $P \in (\mathbb{R}^d)^n$ is said to occur at agent p_i if (p_i, p_j) is an edge in $\mathcal{G}_{\text{disk}}(P, r)$ and the agent p_i does not detect agent p_j. For $\mathcal{P} = i_{\mathbb{F}}(P)$, we denote this link failure by the directed edge $(p_i, p_j) \in \mathcal{P} \times \mathcal{P}$.

Remark 4. Consider an application of the Circumcenter Algorithm over a proximity graph $\mathcal{G}$ as described in the steps 1-5 above. If the link failure (p_i, p_j) takes place at step 1, then the following two events will ensue:

(i) if p_j is a neighbor of p_i according to $\mathcal{G}$, then p_i looses the neighbor p_j at step 2,
(ii) if p_k is not a neighbor of p_i according to $\mathcal{G}$ because of the presence of p_j, then p_i gains the neighbor p_k at step 2.

After steps 1 and 2, the collection of neighbors has been computed inaccurately. Nevertheless the execution of steps 3 through 5 can continue. □

Definition 5. For $P \in (\mathbb{R}^d)^n$, let $\mathcal{P} = i_{\mathbb{F}}(P)$. Let $\mathcal{G}$ be a proximity graph spatially distributed over $\mathcal{G}_{\text{disk}}(r)$ and $F \subset \mathcal{P}\times\mathcal{P}$ be a set of link failures. Let

(i) $\mathcal{G}_{\text{disk}}(\mathcal{P}, r) \nleftarrow F$ be the directed graph with vertex set $\mathcal{P}$ and with edge set $\mathcal{E}_{\text{disk}}(\mathcal{P}, r) \setminus F$;
(ii) $\mathcal{G}(\mathcal{P}) \nleftarrow F$ be the directed graph with vertex set $\mathcal{P}$ and with edges determined as follows; the neighbors of $p \in \mathcal{P}$ are

$$\mathcal{N}_{\mathcal{G},p}\big(\{q \mid (p,q) \in \mathcal{E}_{\text{disk}}(\mathcal{P}, r) \setminus F\}\big),$$

that is, the edges of $\mathcal{G}(\mathcal{P}) \nleftarrow F$ arise from the computation of $\mathcal{G}(\mathcal{P})$ with the link failures F, as described in Remark 4;
(iii) $T_{\mathcal{G} \nleftarrow F}(P)$ is the configuration obtained from applying the Circumcenter Algorithm over $\mathcal{G}$ (steps 1-5) at configuration P with the link failures F at step 1.

Note that only a finite number of possible link failures can occur at any configuration. Consequently, the set of possible directed graphs arising from link failures is finite. We are now ready to state the main robust convergence result, whose proof is provided in (Cortés *et al.*, 2004*a*).

Theorem 6. Let the network $p_1, \dots, p_n$ and the proximity graph $\mathcal{G}$ have the same properties as in Theorem 2. Given $P_0 \in (\mathbb{R}^d)^n$, consider the two sequences $\{P_m\}_{m \in \mathbb{N} \cup \{0\}}$ and $\{F_m\}_{m \in \mathbb{N} \cup \{0\}}$ defined recursively by

(i) F_m is a set of link failures in $\mathcal{G}_{\text{disk}}(r)$ at P_m,
(ii) $P_{m+1} = T_{\mathcal{G} \nleftarrow F_m}(P_m)$.

If there is $\ell \in \mathbb{N}$ such that at least one graph of any ℓ consecutive elements of $\{\mathcal{G}(P_m) \nleftarrow F_m\}_{m \in \mathbb{N} \cup \{0\}}$ is strongly connected, then there exists $p^* \in \mathbb{R}^d$ such that $P_m \to (p^*, \dots, p^*)$ as $m \to +\infty$.

This theorem provides the first theoretical explanation for the robustness behavior against sensor and control errors of the Circumcenter Algorithm over $\mathcal{G}_{\text{disk}}(r)$ observed in (Ando *et al.*, 1999).

Corollary 7. With the same notation as in Theorem 6, if at each step $m \in \mathbb{N}$, the proximity graph $\mathcal{G}(P_m)$ is k_m-connected and if F_m contains at most $k_m - 1$ link failures, then there exists $p^* \in \mathbb{R}^d$ such that $P_m \to (p^*, \dots, p^*)$ as $m \to +\infty$.

Next, we analyze the performance of the Circumcenter Algorithm when each agent of the mobile network at each time step is allowed to use a different proximity graph to compute its neighbors.

Definition 8. Let $\mathcal{S}$ be a set of proximity graph functions that are spatially distributed over $\mathcal{G}_{\text{disk}}(r)$. The *Circumcenter Algorithm over* $\mathcal{S}$ is the Circumcenter Algorithm where step 2 is replaced by

2(a): choose any $\mathcal{G} \in \mathcal{S}$
2(b): compute $\mathcal{M}_i := \mathcal{N}_{\mathcal{G},p_i}(\{q_1, \dots, q_k\}) \cup \{p_i\}$.

The selection algorithm for each agent at each execution of step 2(a) is left unspecified.

Corollary 9. Let the network $p_1, \dots, p_n$ be as in Theorem 2. Let $\mathcal{S}$ be a set of proximity graph functions that are spatially distributed over $\mathcal{G}_{\text{disk}}(r)$. Assume there exists a proximity graph $\mathcal{F}$ with the same connected components as $\mathcal{G}_{\text{disk}}(r)$ such that $\mathcal{F} \subset \mathcal{G}$, for all $\mathcal{G} \in \mathcal{S}$. Then any trajectory $\{P_m\}_{m \in \mathbb{N} \cup \{0\}}$ of the Circumcenter Algorithm over $\mathcal{S}$ has properties (i) and (ii) in Theorem 2.

For $r \in \mathbb{R}_+$, $\mathcal{G}_{\text{RN} \cap \text{disk}}(r)$, $\mathcal{G}_{\text{G} \cap \text{disk}}(r)$ and $\mathcal{G}_{\text{LD}}(r)$ are spatially distributed over $\mathcal{G}_{\text{disk}}(r)$ and contain $\mathcal{G}_{\text{EMST} \cap \text{disk}}(r)$, which has the same connected components as $\mathcal{G}_{\text{disk}}(r)$ (cf. (Cortés *et al.*, 2004*a*)). As a consequence, any subset of $\{\mathcal{G}_{\text{RN} \cap \text{disk}}(r), \mathcal{G}_{\text{G} \cap \text{disk}}(r), \mathcal{G}_{\text{LD}}(r)\}$ satisfies the hypothesis of Corollary 9.

4. SIMULATIONS

In order to illustrate the performance of our rendezvous algorithms, we developed a library of basic geometric routines. The resulting Mathematica® packages `PlanGeom.m` (containing the 2-dimensional routines) and `SpatialGeom.m` (containing the 3-dimensional routines) are freely available at `http://www.soe.ucsc.edu/~jcortes/software`.

The simulation run for the Circumcenter Algorithm in the plane, $d = 2$, over $\mathcal{G}_{\text{LD}}(r)$ with link failures is illustrated in Figure 3. The 25 vehicles have a maximum step size $s_{\text{m}} = .15$, and a communication radius $r = 4$. At each time step, a

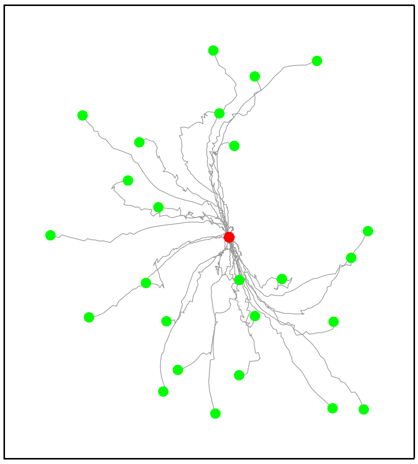

Fig. 3. Evolution (in light gray) of the Circumcenter Algorithm over the r-limited Delaunay graph $\mathcal{G}_{\text{LD}}(r)$ with link failures. The initial configuration of the network is as in Figure 1.

set consisting of 18 numbers between 1 and 25 is randomly selected, corresponding to the identities of the agents where link failures occur. For each of them, a randomly selected link failure in $\mathcal{G}_{\text{disk}}(r)$ is chosen. Since the identity of an agent might appear more than once in the random set, more than one link failure may occur at the same agent. However, rendezvous is asymptotically achieved according to Theorem 6 (usually after 80 steps).

The simulation run for the Circumcenter Algorithm in space, $d = 3$, over the set $\{\mathcal{G}_{\mathrm{disk}}(r), \mathcal{G}_{\mathrm{G}}(r) \cap \mathcal{G}_{\mathrm{disk}}(r), \mathcal{G}_{\mathrm{RN}}(r) \cap \mathcal{G}_{\mathrm{disk}}(r)\}$ is illustrated in Figure 4. The 25 vehicles have, as before, a maximum step size $s_{\mathrm{m}} = .15$, and a communication radius $r = 4$. At each time step, each agent randomly selects one of the proximity graphs in $\{\mathcal{G}_{\mathrm{disk}}(r), \mathcal{G}_{\mathrm{RN}\cap\mathrm{disk}}(r), \mathcal{G}_{\mathrm{G}\cap\mathrm{disk}}(r)\}$ and computes its corresponding set of neighbors according to it. Then, it executes steps 3 through 5 of the Circumcenter Algorithm. Rendezvous is achieved, according to Corollary 9 (in this case, in a finite number of steps- usually 100).

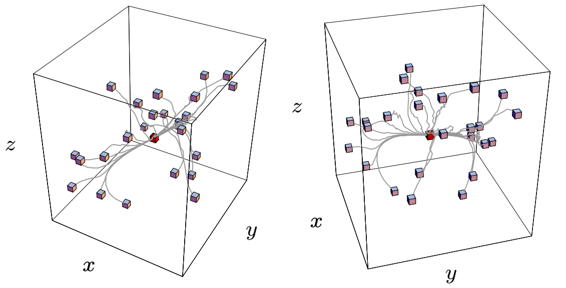

Fig. 4. Evolution (in light gray) of the Circumcenter Algorithm over $\{\mathcal{G}_{\mathrm{disk}}(r), \mathcal{G}_{\mathrm{G}}(r) \cap \mathcal{G}_{\mathrm{disk}}(r), \mathcal{G}_{\mathrm{RN}}(r) \cap \mathcal{G}_{\mathrm{disk}}(r)\}$. The initial configuration of the network is as in Figure 2. The right figure is a rotated view of the left figure by 45 degrees.

5. CONCLUSIONS

We have designed and analyzed a class of circumcenter algorithms over proximity graphs for multi-agent rendezvous. Also, we have provided a set of novel tools that we believe are important in the design and analysis of general motion coordination algorithms. Future directions of research include the study of increasingly realistic communication settings (asynchronicity, quantization, media access and power control issues), the analysis of the performance and complexity of the algorithms, and the formal design of other spatially distributed coordination primitives.

REFERENCES

Ando, H., Y. Oasa, I. Suzuki and M. Yamashita (1999). Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transactions on Robotics and Automation* **15**(5), 818–828.

Cortés, J., S. Martínez and F. Bullo (2004*a*). Robust rendezvous for mobile autonomous agents via proximity graphs in d dimensions. *IEEE Transactions on Automatic Control*. Submitted. Electronic version available at `http://www.me.ucsb.edu/bullo`.

Cortés, J., S. Martínez and F. Bullo (2004*b*). Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM. Control, Optimisation & Calculus of Variations*. Electronic version available at `http://www.me.ucsb.edu/bullo`. To appear.

de Berg, M., M. van Kreveld and M. Overmars (1997). *Computational Geometry: Algorithms and Applications*. Springer Verlag. New York.

Diestel, R. (2000). *Graph Theory*. Vol. 173 of *Graduate Texts in Mathematics*. second ed.. Springer Verlag. New York.

Flocchini, P., G. Prencipe, N. Santoro and P. Widmayer (2001). Gathering of asynchronous oblivious robots with limited visibility. In: *STACS 2001, 18th Annual Symposium on Theoretical Aspects of Computer Science (Dresden, Germany)* (A. Ferreira and H. Reichel, Eds.). Vol. 2010 of *Lecture Notes in Computer Science*. pp. 247–258. Springer Verlag. New York.

Jaromczyk, J. W. and G. T. Toussaint (1992). Relative neighborhood graphs and their relatives. *Proceedings of the IEEE* **80**(9), 1502–1517.

Justh, E. W. and P. S. Krishnaprasad (2004). Equilibria and steering laws for planar formations. *Systems & Control Letters* **52**(1), 25–38.

Klavins, E., R. Ghrist and D. Lipsky (2004). A grammatical approach to self-organizing robotic systems. *IEEE Transactions on Automatic Control*. Submitted.

Li, X.-Y. (2003). Algorithmic, geometric and graphs issues in wireless networks. *Wireless Communications and Mobile Computing* **3**(2), 119–140.

Lin, J., A. S. Morse and B. D. O. Anderson (2004*a*). The multi-agent rendezvous problem - the asynchronous case. In: *IEEE Conf. on Decision and Control*. Paradise Island, Bahamas. pp. 1926–1931.

Lin, J., A. S. Morse and B. D. O. Anderson (2004*b*). The multi-agent rendezvous problem: an extended summary. In: *Proceedings of the 2003 Block Island Workshop on Cooperative Control* (V. Kumar, N. E. Leonard and A. S. Morse, Eds.). Vol. 309 of *Lecture Notes in Control and Information Sciences*. pp. 257–282. Springer Verlag. New York.

Lin, Z., M. Broucke and B. Francis (2004*c*). Local control strategies for groups of mobile autonomous agents. *IEEE Transactions on Automatic Control* **49**(4), 622–629.

Liu, Y. and K. M. Passino (2004). Stable social foraging swarms in a noisy environment. *IEEE Transactions on Automatic Control* **49**(1), 30–44.

Suzuki, I. and M. Yamashita (1999). Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal on Computing* **28**(4), 1347–1363.