Motion Coordination with Distributed Information

The challenge of obtaining global behavior out of local interactions

Sonia Martínez Jorge Cortés Francesco Bullo

Think globally, act locally
René J. Dubos, 1972

Introduction

Motion coordination is a remarkable phenomenon in biological systems and an extremely useful tool in man-made groups of vehicles, mobile sensors and embedded robotic systems. Just like animals do, groups of mobile autonomous agents need the ability to deploy over a given region, assume a specified pattern, rendezvous at a common point, or jointly move in a synchronized manner. These coordination tasks are typically to be achieved with little available communication between the agents, and therefore, with limited information about the state of the entire system.

An important scientific motivation for the study of motion coordination is the analysis of emergent and self-organized behaviors in biological groups with distributed agent-to-agent interactions. Interesting dynamical systems arise in biological networks at multiple levels of resolution, all the way from interactions between molecules and cells, see [1], to the behavioral ecology of animal groups, see [2]. Flocks of birds and school of fish are able to travel in formation and act as one unit (see [3] and Figures 1 and 2); these swarming behaviors allow animals to defend themselves against predators and to protect areas that they claim as their own. Wildebeest and other animals exhibit complex collective behaviors when migrating (see [4, 5] and Figure 3). Certain foraging behaviors include individual animals partitioning their environment in non-overlapping zones (see [6] and Figure 4). Honey bees [7], gorillas [8], and whitefaced capuchins [9] exhibit synchronized group activities such as initiation of motion and change of travel direction. These remarkable dynamic capabilities are achieved apparently without following a group leader; see [2, 3, 5, 6, 7, 8, 9] for specific examples of animal species and [10, 11] for general studies. In other words, these complex coordinated behaviors emerge while each individual has no global knowledge of the network state and can only plan its motion by observing its closest neighbors.

At the same time, an important engineering reason to study motion coordination stems from the recent interest in man-made groups of embedded systems (such as multi-vehicles and sensor networks). Indeed, the vision is that groups of autonomous agents with computing,

Figure 1. School of fish. Groups of animals are able to act as one unit apparently without following a group leader. Photograph taken by the authors at the 50th IEEE Conference in Decision and Control at Paradise Island, Bahamas, in December 2004.

Figure 2. Flock of snow geese. Self-organized behaviors emerge in biological groups even though each individual has no global knowledge of the group state. Snow geese fly in formation during migration near the Eastern Shore of Virginia National Wildlife Refuge. Photograph taken from [12].

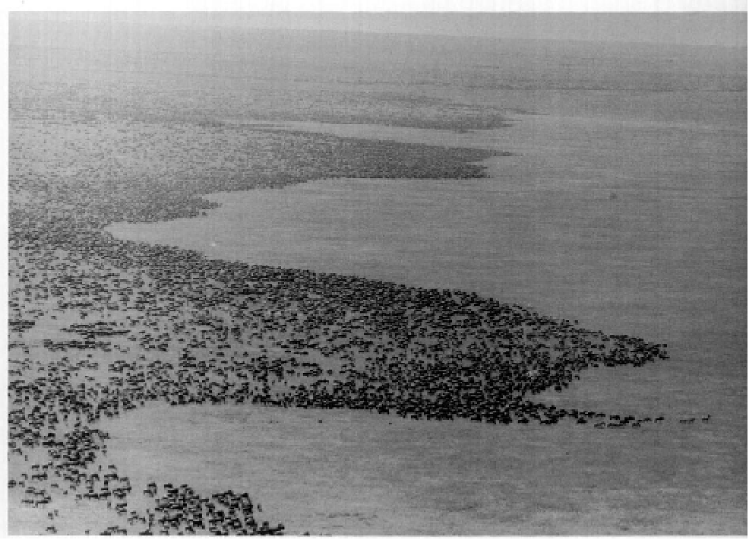


Figure 3. Herd of wildebeest in the Serengeti National Park, Tanzania. Wildebeest and other animals exhibit complex coordinated behaviors when migrating. Aerial photograph taken from [4].

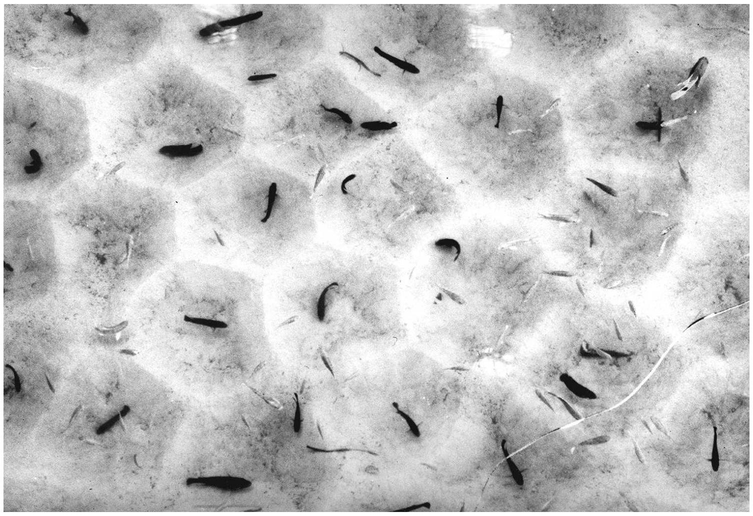


Figure 4. Territories of male Tilapia mossambica. Some species of fish exhibit territorial behavior by globally partitioning the environment into non-overlapping zones. In this top-view photograph, each territory is a pit dug in the sand by its occupant. The rims of the pits form a pattern of polygons. The breeding males are the black fish, which range in size from about 15 cm to 20 cm. The gray fish are the females, juveniles, and nonbreeding males. Top-view photograph taken from [6].

communication and mobility capabilities will soon become economically feasible and perform a variety of spatially-distributed sensing tasks such as search and rescue, surveillance, environmental monitoring, and exploration.

As a consequence of this growing interest, the research activity on cooperative control has increased tremendously over the last few years. A necessarily incomplete list of works on distributed, or leaderless, motion coordination includes [13, 14, 15] on pattern formation, [16, 17, 18] on flocking, [19] on self-assembly, [20] on swarm aggregation, [21] on gradient climbing, [22, 23, 24, 25] on deployment and task allocation, [26, 27, 28, 29] on rendezvous, [30, 31] on cyclic pursuit, [32] on vehicle routing and [33, 34, 35] on consensus. Heuristic approaches to the design of interaction rules and emergent behaviors are thoroughly investigated within the literature on behavior-based robotics; see for example [36, 37, 38, 39].

The objective of this paper is to illustrate ways in which systems theory helps us analyze emergent behaviors in animal groups and design autonomous and reliable robotic networks. We present and survey some recently-developed theoretical tools for modeling, analysis and design of motion coordination. We pay special attention to the distributed character of coordination algorithms, the rigorous characterization of their performance, and the development of design methodologies provide mobile networks with provably correct cooperative strategies.

First, we are interested in how to characterize the distributed character of cooperative strategies. To arrive at a satisfactory notion, we resort to the concept of proximity graph from computational geometry [40]. Proximity graphs of different type model agent-to-agent interactions that depend only on the agents' location in space. This is the case for example in wireless communication or in communication based on line-of-sight. Thus, the notion of proximity graph allows us to model the information flow between mobile agents.

Second, we consider the problem of mathematically expressing motion coordination tasks. This question is important if we are interested in providing analytical guarantees for the performance of coordination algorithms. We discuss various aggregate objective functions from geometric optimization for tasks such as deployment (by means of a class of multicenter functions that encode area-coverage, detection likelihood, and visibility coverage), rendezvous (by means of the diameter of convex hull function), cohesiveness, and agreement (by means of the Laplacian potential from algebraic graph theory). We also discuss their smoothness properties and identify their extreme points using nonsmooth analysis.

Third, we discuss some techniques that help assess the performance of coordination algorithms. We consider a combination of system-theoretic and linear algebraic tools that are helpful in establishing stability and convergence of motion coordination algorithms. This treatment includes methods from circulant and Toeplitz tridiagonal matrices and a recently-developed version of the invariance principle for non-deterministic discrete-time dynamical systems.

Fourth, and finally, we focus our attention on designing distributed coordination algorithms for specific tasks. We build on the tools introduced earlier and present various approaches. A first approach is based on the design of gradient flows: here we are typically given a coordination task to be performed by the network and a proximity graph as communication constraint. A second approach is based on the analysis of emergent behaviors: in this case a notion of neighboring agents and an interaction law between them is usually given. A third approach is based on the identification of meaningful local objective functions whose optimization helps the network achieve the desired global task. The last and fourth approach relies on the composition of basic behaviors. We apply these approaches to numerous examples of coordination algorithms developed in the literature.

Making sense of *distributed*

Our first goal is to provide a clear notion of spatially distributed coordination algorithms. Roughly speaking, we would characterize an algorithm as distributed, as opposed to centralized, if the algorithm relies on local information (instead of on global knowledge). The literature on automata theory and parallel computing [41, 42] presents precise notions of distributed algorithms for networks with fixed topology. Here, however, we are interested in ad-hoc networks of mobile agents, where the topology changes dynamically, and these definitions are not completely applicable. This feature motivates our current effort to arrive at a satisfactory definition of spatially distributed algorithms. In doing so, we borrow the notion of proximity graph from computational geometry. Before getting into it, let us recall some basic geometric notions.

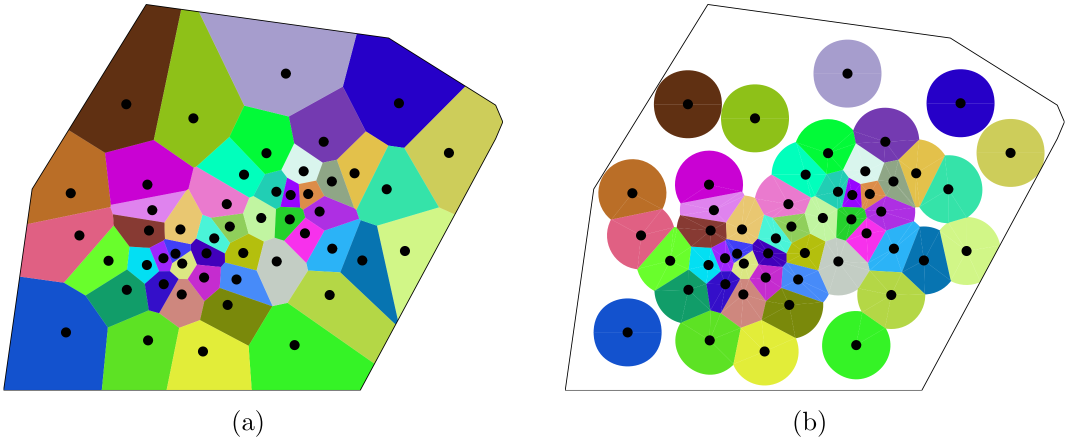
Basic geometric notions

A *partition* of a set S is a subdivision of S into components that are disjoint except for their boundary. Two examples of partitions are useful in our problems. Given $S \subset \mathbb{R}^2$ and a set of n distinct points $\mathcal{P} = \{p_1, \dots, p_n\}$ in S, the *Voronoi partition* [43, 44] of S generated by $\mathcal{P}$ is the collection of sets $\{V_i(\mathcal{P})\}_{i \in \{1,\dots,n\}}$ defined by $V_i(\mathcal{P}) = \{q \in S \mid \|q - p_i\| \leq \|q - p_j\|\,,\ \text{for all } p_j \in \mathcal{P}\}$. An example Voronoi partition is depicted in the left plot of Figure 5.

For $p \in \mathbb{R}^2$ and $r > 0$, we denote by $B(p, r)$ the closed ball in $\mathbb{R}^2$ centered at p of radius r. For a set of n distinct points $\mathcal{P} = \{p_1, \dots, p_n\}$ in S, the *r-limited Voronoi partition* inside S is the collection of sets $\{V_{i,r}(\mathcal{P}) = V_i(\mathcal{P}) \cap B(p_i, r)\}_{i \in \{1,\dots,n\}}$. This name is justified by the fact that these sets are the Voronoi partition of $\cup_i B(p_i, r) \cap S$. The right plot in Figure 5 shows an example of this geometric construction. We refer to $V_i(\mathcal{P})$ and $V_{i,r}(\mathcal{P})$ as the *Voronoi cell* and the *r-limited Voronoi cell* of p_i, respectively.

Proximity graphs and their properties

The notion of proximity graph allows us to capture the interconnection topology of a network of robotic agents. Roughly speaking, a proximity graph is a graph whose vertex set is a set of distinct points on the Euclidean space and whose edge set is a function of the relative locations of the point set. Let us clarify this notion. A *proximity graph* $\mathcal{G}$ associates to $\mathcal{P} = \{p_1, \dots, p_n\} \subset \mathbb{R}^d$ an undirected graph with vertex set $\mathcal{P}$ and whose edge set is


(a)

(b)

Figure 5. Two types of Voronoi partitions. The decomposition of the environment induced by Voronoi partitions has applications in diverse areas such as wireless communications, signal compression, facility location, and mesh optimization. Here, we explore the application of Voronoi partition to deployment problems of multi-agent networks. (a) and (b) show, respectively, the Voronoi and the r-limited Voronoi partition of a sample convex polygon. In both cases the generators are 50 randomly-selected points. The colored regions are Voronoi cells and r-limited Voronoi cells, respectively.

$\mathcal{E}_{\mathcal{G}}(\mathcal{P}) \subseteq \{(p, q) \in \mathcal{P} \times \mathcal{P} \mid p \neq q\}$. Therefore, a point cannot be its own neighbor. From this definition, we observe that the distinguishing feature of proximity graphs is that their edge sets change with the location of their vertices. A related notion is that of state-dependent graphs, see [45]. Let us provide some examples of proximity graphs (see [23, 40, 43] for further reference):

(i) the *r-disk graph* $\mathcal{G}_{\text{disk}}(r)$, for $r > 0$, where two agents are neighbors if they are located within a distance r, that is, $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{disk}}(r)}(\mathcal{P})$ if $\|p_i - p_j\| \leq r$;

(ii) the *Delaunay graph* $\mathcal{G}_{\text{D}}$, where two agents are neighbors if their corresponding Voronoi cells intersect, that is, $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{D}}}(\mathcal{P})$ if $V_i(\mathcal{P}) \cap V_j(\mathcal{P}) \neq \emptyset$;

(iii) the *r-limited Delaunay graph* $\mathcal{G}_{\text{LD}}(r)$, for $r > 0$, where two agents are neighbors if their corresponding r-limited Voronoi cells intersect, that is, $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{LD}}}(\mathcal{P})$ if $V_{i,r}(\mathcal{P}) \cap V_{j,r}(\mathcal{P}) \neq \emptyset$;

(iv) given a simple polytope Q in $\mathbb{R}^d$, the *visibility graph* $\mathcal{G}_{\text{vis,Q}}$, where two agents are neighbors if they can see each other, that is, $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{vis,Q}}}(\mathcal{P})$ if the closed segment from p_i to p_j is contained in Q.

Additionally, we introduce the *complete graph* $\mathcal{G}_{\text{complete}}$, where all agents are neighbors of each other. This notion allows us to model fully interconnected networks. Finally, it is also worth mentioning the *Euclidean Minimum Spanning Tree*, whose properties are studied in combinatorial and geometric optimization. Figure 6 shows some examples of these proximity graphs. The connectivity properties of these graphs play an important role in coordination problems and are the subject of current study, see [23].

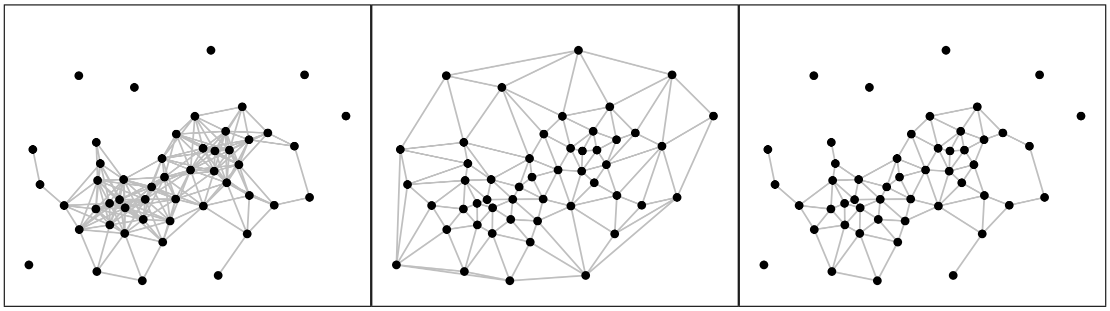


Figure 6. Proximity graphs in $\mathbb{R}^2$. Proximity graphs provide a natural way to mathematically model the interconnection topology among the agents resulting from their sensing or communication capabilities. From left to right, $2r$-disk, Delaunay, and r-limited Delaunay graphs for the point set in Figure 5.

In the case of graphs with a fixed topology, it is typical to alternatively describe the edge set by means of the sets of neighbors of the individual graph vertices. Likewise, we can associate to each proximity graph $\mathcal{G}$, each $p \in \mathbb{R}^d$ and each $\mathcal{P} = \{p_1, \dots, p_n\} \subset \mathbb{R}^d$, the *set of neighbors*

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

Finally, we consider the broad question of when does a given proximity graph provide sufficient information to compute a second proximity graph. We say that a proximity graph $\mathcal{G}_1$ is *spatially distributed* over a proximity graph $\mathcal{G}_2$, if any agent equipped with the location of its neighbors according to $\mathcal{G}_2$ can compute its set of neighbors according to $\mathcal{G}_1$. For instance, if an agent knows the position of its neighbors in the complete graph (that is, of every other agent in the network), then it is clear that the agent can compute its neighbors with respect to any proximity graph. More interestingly, the $r/2$-limited Delaunay graph $\mathcal{G}_{\mathrm{LD}}(\frac{r}{2})$ is spatially distributed over r-disk graph $\mathcal{G}_{\mathrm{disk}}(r)$. Note that, if $\mathcal{G}_1$ is spatially distributed over $\mathcal{G}_2$, then $\mathcal{G}_1(\mathcal{P}) \subset \mathcal{G}_2(\mathcal{P})$ for all $\mathcal{P}$, but the converse is in general not true. For instance, $\mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$ is a subgraph of $\mathcal{G}_{\mathrm{disk}}$, but $\mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$ is not spatially distributed over $\mathcal{G}_{\mathrm{disk}}(r)$ (see [23] for further details).

Spatially distributed maps

We are now ready to introduce the notion of a spatially distributed map. To simplify the exposition, we do not distinguish notationally between a tuple in $(p_1, \dots, p_n) \in (\mathbb{R}^d)^n$ and its associated point set $\{p_1, \dots, p_n\} \subset \mathbb{R}^d$; we denote both quantities by P. More details on this concept are in [29].

Consider a map $T : (\mathbb{R}^d)^n \to Y^n$, with Y a given set. We say that T is *spatially distributed over a proximity graph* $\mathcal{G}$ if the jth component T_j evaluated at $(p_1, \dots, p_n)$ can be computed with only the knowledge of the vertex p_j and the neighboring vertices in $\mathcal{G}(\{p_1, \dots, p_n\})$.

According to this definition, a proximity graph $\mathcal{G}_1$ is spatially distributed over a proximity graph $\mathcal{G}_2$ if and only if the set of neighbors map $\mathcal{N}_{\mathcal{G}_1}$ is spatially distributed over $\mathcal{G}_2$. Later, when discussing various coordination algorithms, we characterize them as being spatially distributed with regards to appropriate proximity graphs.

Encoding coordination tasks

Our second goal is to develop mathematically sound methods to express motion coordination tasks. In the following, we argue that aggregate behaviors of the entire mobile network can be typically quantified by means of appropriately defined objective functions. Using tools from geometric optimization, we show how to encode various network objectives into locational optimization functions. We also pay special attention to the smoothness properties of these functions and the spatially distributed character of their gradients.

Aggregate objective functions for deployment

Loosely speaking, the deployment problem consists of placing a network of mobile agents inside an environment of interest to achieve maximum coverage of it. Of course, "coverage" can be defined in many possible ways, as we illustrate in the following discussion.

Let $Q \subset \mathbb{R}^d$ be a convex polytope. A *density function* $\phi : Q \to \mathbb{R}_{\geq 0}$ is a bounded measurable function. We can regard ϕ as a function measuring the probability that some event takes place over the environment. A *performance function* $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$ is a non-increasing and piecewise differentiable function with finite jump discontinuities. This function describes the utility of placing an agent at a certain distance from a location in the environment. To illustrate this notion, consider a sensing scenario in which the agents are equipped with acoustic sensors that take measurements of sounds originating in the environment. Because of noise and loss of resolution, the ability to detect a sound originating at a point q from the ith sensor at the position p_i degrades with the distance $\|q - p_i\|$. This ability is measured by the performance function f.

Given a density function ϕ and a performance function f, we are interested in maximizing the *expected value of the coverage performance* provided by the group of agents over any point in Q. To this end, let us define the function $\mathcal{H} : Q^n \to \mathbb{R}$ by

$$\mathcal{H}(P) = \int_Q \max_{i \in \{1,\dots,n\}} f(\|q - p_i\|)\phi(q)dq. \tag{1}$$

Since $\mathcal{H}$ depends on all the locations $p_1, \dots, p_n$, $\mathcal{H}$ is an aggregate objective function. We seek to find local maximizers for $\mathcal{H}$.

Different choices of performance function give rise to different aggregate objective functions with particular features. We now examine some important cases (let us remind the reader that $\{V_i(P)\}_{i\in\{1,\dots,n\}}$ and $\{V_{i,R}(P)\}_{i\in\{1,\dots,n\}}$ denote the Voronoi partition and the R-limited Voronoi partition of Q generated by $P \in (\mathbb{R}^d)^n$, respectively):

Distortion problem: If $f(x) = -x^2$, then $\mathcal{H}$ takes the form

$$\mathcal{H}_{\mathrm{C}}(P) = -\sum_{i=1}^{n} \int_{V_i(P)} \|q - p_i\|^2 \phi(q) dq = -\sum_{i=1}^{n} J(V_i(P), p_i),$$

where $J(W, p)$ denotes the polar moment of inertia of the set $W \subset Q$ about the point p. In signal compression $-\mathcal{H}_{\mathrm{C}}$ is referred to as the distortion function and is relevant in many disciplines including facility location, numerical integration, and clustering analysis, see [46].

Area problem: For a set S, let 1_S denote the indicator function, $1_S(q) = 1$, if $q \in S$, and $1_S(q) = 0$, if $q \notin S$. If $f = 1_{[0,R]}$, then $\mathcal{H}$ corresponds to the area, measured according to ϕ, covered by the union of the n balls $B(p_1, R), \dots, B(p_n, R)$; that is,

$$\mathcal{H}_{\mathrm{area},R}(P) = \mathrm{area}_\phi(\cup_{i=1}^n B(p_i, R))\,,$$

where $\mathrm{area}_\phi(S) = \int_S \phi(q) dq$.

Aggregate objective functions for visibility-based deployment

Given a nonconvex polytope Q in $\mathbb{R}^d$ and $p \in Q$, let $S(p) = \{q \in Q \mid [q, p] \subset Q\}$ denote the *visible region in* Q from the location p (here $[q, p]$ is the closed segment from q to p). Define

$$\mathcal{H}_{\mathrm{vis}}(P) = \int_Q \max_{i\in\{1,\dots,n\}} 1_{S(p_i)}(q) dq.$$

Roughly speaking, the function $\mathcal{H}_{\mathrm{vis}}$ measures the amount of area of the nonconvex polygon Q which is visible from any of the agents located at $p_1, \dots, p_n$. Therefore, we seek to find local maximizers of $\mathcal{H}_{\mathrm{vis}}$. Note that we can also extend the definition of $\mathcal{H}_{\mathrm{vis}}$ using a density function $\phi : Q \to \mathbb{R}_{\geq 0}$, so that more importance is given to some regions of the environment being visible to the network (for instance, doors) than others.

Aggregate objective functions for consensus

In this section we consider a setup based on a fixed graph instead of a proximity graph. Let $G = (\{1, \dots, n\}, E)$ be an undirected graph with n vertices. The *Laplacian matrix* L associated with the graph G (see, for instance, [47]) is the $n \times n$ matrix defined by:

$$L_{ij} = \begin{cases} -1, & \text{if } (i,j) \in E, \\ \mathrm{degree}(i), & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases}$$

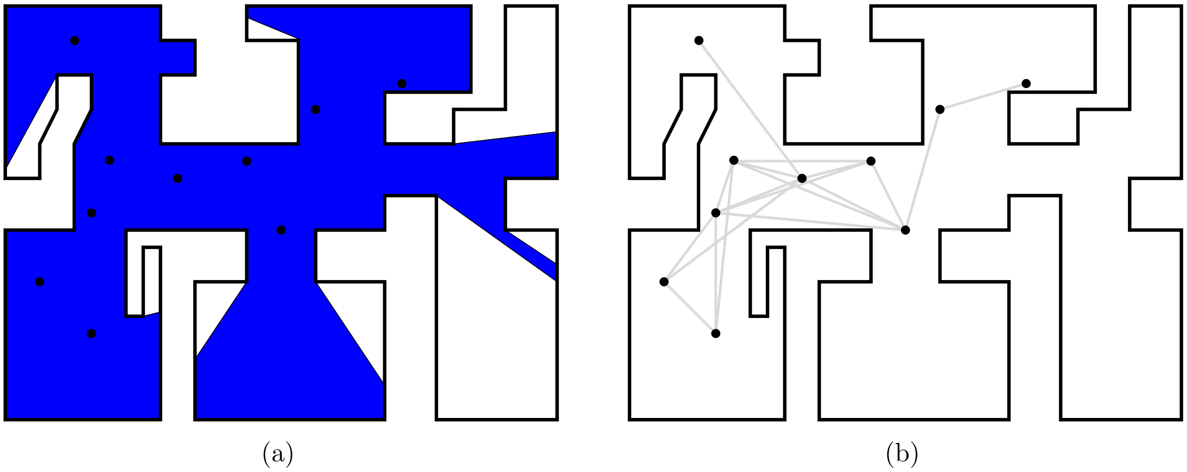


Figure 7. Visibility-based deployment. A network of agents equipped with omnidirectional cameras is able to see the blue-colored region of the nonconvex environment in (a). The underlying visibility proximity graph is depicted in (b). The overall objective is to deploy the agents in such as way as to maximize the area visible to the network.

where degree(i) is the number of neighbors of node i. The Laplacian matrix has some useful properties: L is positive semi-definite, L is not full rank, and L has rank $n-1$ if and only if the graph G is connected. Following [33], we define the *disagreement function* or *Laplacian potential* $\Phi_G : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ associated with G by

$$\Phi_G(x) = x^T L x = \frac{1}{2} \sum_{(i,j)\in E} (x_j - x_i)^2 .$$

For $i \in \{1, \dots, n\}$, the variable x_i is associated with the agent p_i. The variable x_i might represent physical quantities including heading, position, temperature, or voltage. Two agents p_i and p_j are said to *agree* if and only if $x_i = x_j$. It is clear that $\Phi_G(x) = 0$ if and only if all neighboring nodes in the graph G agree. If, in addition, the graph G is connected, then all nodes in the graph agree and a consensus is reached. Therefore, $\Phi_G(x)$ is a meaningful function that quantifies the group disagreement in a network.

Note that achieving consensus is a network coordination problem that does not necessarily refer to physical variables such as spatial coordinates or velocities. In what follows we consider two "spatial versions" of consensus, that we refer to as rendezvous and cohesiveness.

Aggregate objective function for rendezvous

Roughly speaking, rendezvous means agreement over the location of the agents in a network. An objective function that is useful for the purpose of rendezvous is $V_{\text{diam}} : (\mathbb{R}^d)^n \to \mathbb{R}_{\geq 0}$ defined by

$$V_{\text{diam}}(P) = \max\{\|p_i - p_j\| \mid i, j \in \{1, \dots, n\}\}.$$

It is clear that $V_{\text{diam}}(P) = 0$ if and only if $p_i = p_j$ for all $i, j \in \{1, \dots, n\}$. Therefore, the set of global minimizers of V_{diam} corresponds to the network configurations where the agents rendezvous. The map $V_{\text{diam}} : (\mathbb{R}^d)^n \to \mathbb{R}_{\geq 0}$ is locally Lipschitz and invariant under permutations of its arguments.

Aggregate objective functions for cohesiveness

Let us consider one final example of aggregate objective function that encodes a motion coordination task. A *repulsion/attraction function* $h : \mathbb{R}_{\geq 0} \to \mathbb{R}$ is a continuously differentiable function satisfying the following conditions: (i) $\lim_{R \to 0^+} h(R) = +\infty$, (ii) there exists $R_0 > 0$ such that h is convex on $(0, R_0)$ and concave on $(R_0, +\infty)$, (iii) h achieves its minimum at all the points in the interval $[R_*, R'_*] \subset (0, R_0)$, and (iv) there exists $R_1 \geq R_0$ such that $h(R) = c$ for all $R \geq R_1$. Figure 8 gives a particular example of a repulsion/attraction function.

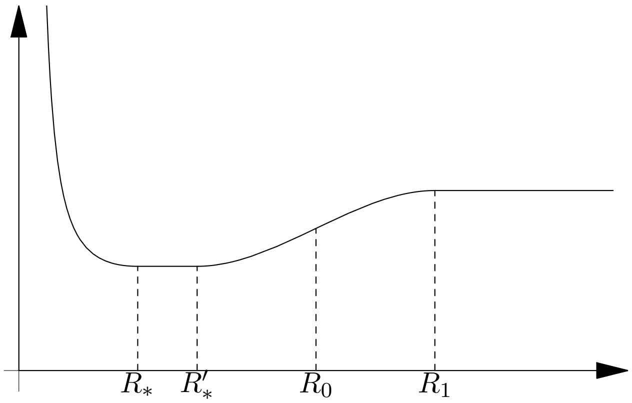


Figure 8. Sample repulsion/attraction function. Repulsion/attraction functions serve to encode desirable network configurations where all neighboring agents are located within a distance contained in $[R_*, R'_*]$. These functions play a key role in cohesiveness problems for mobile networks.

Let $\mathcal{G}$ be a proximity graph. Define the aggregate objective function

$$\mathcal{H}_{\text{cohe},\mathcal{G}}(P) = \sum_{(p_i,p_j) \in \mathcal{E}_{\mathcal{G}}(P)} h(\|p_i - p_j\|) .$$

The minimizers of $\mathcal{H}_{\text{cohe},\mathcal{G}}$ correspond to "cohesive" network configurations. Specifically, for $n \in \{2, 3\}$, configurations that achieve the minimum value of $\mathcal{H}_{\text{cohe},\mathcal{G}}$ have all neighboring agents' locations within a distance contained in the interval $[R_*, R'_*]$. This objective function and its variations are employed over different proximity graphs in a number of works in the literature ([20] and [21] over the complete graph, [17] over the r-disk graph) to guarantee collision avoidance and cohesiveness of the mobile network.

Correctness and performance analysis of coordination algorithms

In this section we discuss various techniques used in the literature to analyze cooperative control problems. Let us first describe informally the notion of coordination algorithm. Roughly speaking, such an algorithm consists of a control law for each agent of the network. Mathematically, a coordination algorithm is either a vector field or a map depending on whether the dynamical model is defined on continuous or discrete time.

Given a coordination algorithm, a first scientific concern is the investigation of its correctness. From a mathematical viewpoint, when a coordination algorithm for a networked control system is designed, or a particular interaction law modeling a biological behavior is described, a set of coupled dynamical systems arise. Of course, the couplings between the various dynamical systems change as the topology of the mobile network changes, making things intriguing and complicated at the same time. Conceptually, we can loosely understand that an algorithm behaves correctly when certain sets (encoding the desired behaviors) are invariant and attractive for the evolution of the closed-loop network. A second relevant concern regards the properties of the algorithm execution. For instance, one often faces issues such as discontinuity and non-smoothness of the vector fields modeling the evolution of the network. Other times, non-determinism arises because of asynchronism (for example, to simplify the analysis of an algorithm, the asynchronous, deterministic evolution of a mobile network may be subsumed into a larger set of synchronous, non-deterministic evolutions, see [27]), design choices when devising the coordination algorithm (at each time instant throughout the evolution, each agent may choose among multiple possible control actions, as opposed to a single one, see [23]) or communication, control and sensor errors during the execution of the coordination algorithm (see [26, 29]). It is also of interest to have estimates on how quickly a coordination algorithm completes the required task, as well as on how costly the algorithm is in terms of computations, exchanged messages and energy consumption.

Among the analysis methods used, we roughly distinguish between linear techniques (ergodic, stochastic [16] and circulant matrices [31] from matrix analysis, graph Laplacians and algebraic connectivity [16, 33] from algebraic graph theory) and nonlinear techniques (symmetries of differential equations [15], invariance principles for differential inclusions and for non-deterministic dynamical systems [24], graph grammars [19] from automata theory). For reasons of brevity, it is not possible to include here a comprehensive account of all these methods. Instead, we include here two sidebars about two insightful techniques: a class of Toeplitz matrices and the invariance principle for non-deterministic dynamical systems. The interested reader is invited to explore the references in the bibliography for more in-depth discussions of these and other methods.

Tridiagonal Toeplitz and circulant matrices

Toeplitz and circulant matrices are classic research subjects and we refer to [48, 49] for extensive treatments. For $n \geq 2$ and $a, b, c \in \mathbb{R}$, we define the $n \times n$ matrices $\mathrm{Trid}_n(a, b, c)$ and $\mathrm{Circ}_n(a, b, c)$ by

$$\mathrm{Trid}_n(a,b,c) = \begin{bmatrix} b & c & 0 & \dots & 0 \\ a & b & c & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & a & b & c \\ 0 & \dots & 0 & a & b \end{bmatrix}, \quad \mathrm{Circ}_n(a,b,c) = \mathrm{Trid}_n(a,b,c) + \begin{bmatrix} 0 & \dots & \dots & 0 & a \\ 0 & \dots & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ c & 0 & \dots & 0 & 0 \end{bmatrix}.$$

We refer to Trid_n and Circ_n as tridiagonal Toeplitz and circulant, respectively. These matrices appear when the communication network has the chain or the ring topology as, for instance, in rendezvous [50] and in cyclic pursuit [30, 31] problems. In Figure 9, we illustrate two algorithms in which the control action of each agent depends on the location of the agent's clockwise and counterclockwise neighbors.

An important feature of these matrices is that their eigenvalues and their dependency on n can be explicitly computed; see [50]. First, consider the discrete-time trajectory $x : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ satisfying

$$x(\ell+1) = \mathrm{Trid}_n(a,b,c)\, x(\ell), \qquad x(0) = x_0.$$

For the relevant case where $a = c \neq 0$ and $|b| + 2|a| = 1$, we can show not only that $\lim_{\ell \to +\infty} x(\ell) = 0$, but more importantly that the maximum time required for $\|x(\ell)\|_2 \leq \varepsilon \|x_0\|_2$ is of order $n^2 \log \varepsilon^{-1}$, for small ε. Second, consider the discrete-time trajectory $y : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ satisfying

$$y(\ell+1) = \mathrm{Circ}_n(a,b,c)\, y(\ell), \qquad y(0) = y_0.$$

For the relevant case where $a \geq 0$, $c \geq 0$, $b > 0$, and $a + b + c = 1$, we can show that $\lim_{\ell \to +\infty} y(\ell) = y_{\mathrm{ave}} \mathbf{1}$, where $y_{\mathrm{ave}} = \frac{1}{n} \mathbf{1}^T y_0$, and that the maximum time required for $\|y(\ell) - y_{\mathrm{ave}} \mathbf{1}\|_2 \leq \varepsilon \|y_0 - y_{\mathrm{ave}} \mathbf{1}\|_2$ is again of order $n^2 \log \varepsilon^{-1}$. Here $\mathbf{1} = (1, \dots, 1)^T$.

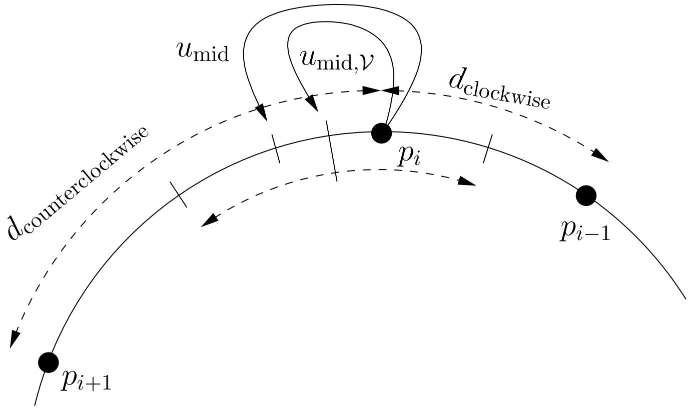


Figure 9. Clockwise and counterclockwise neighbors of an agent in a network evolving in $\mathbb{S}^1$. Control laws such as "go toward the midpoint u_{mid} of the locations of the clockwise and counterclockwise neighbors", or "go toward the midpoint $u_{\mathsf{mid},\mathcal{V}}$ of the Voronoi segment of the agent" give rise to linear dynamical systems described by circulant matrices. In the closed-loop system determined by $u_{\mathsf{mid},\mathcal{V}}$, the agents achieve a uniform distribution along $\mathbb{S}^1$. Oscillations instead persist when the law u_{mid} is adopted.

Invariance principle for non-deterministic dynamical systems
Here we briefly survey a recently-developed invariance principle for non-deterministic discrete-time dynamical systems. Let T be a *set-valued map* on $\mathbb{R}^n$, that is, a map that associated to a point in $\mathbb{R}^n$ a non-empty set in $\mathbb{R}^n$. A *trajectory* of T is a curve $p : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ with the property that

$$p(\ell + 1) \in T(p(\ell)).$$

In other words, given any initial $p_0 \in \mathbb{R}^n$, a trajectory of T is computed by recursively setting $p(\ell + 1)$ equal to an arbitrary element in $T(p(\ell))$. Therefore, T induces a non-deterministic discrete-time dynamical system [23, 51]. To study the stability of this type of discrete-time dynamical systems, we need to introduce a couple of new notions. According to [51], T is *closed at* $p \in \mathbb{R}^n$ if for all pairs of convergent sequences $p_k \to p$ and $p'_k \to p'$ such that $p'_k \in T(p_k)$, we have $p' \in T(p)$. In particular, every map $T : \mathbb{R}^n \to \mathbb{R}^n$ continuous at $p \in \mathbb{R}^n$ is closed at p. A set C is *weakly positively invariant with respect to* T if, for any initial condition $p_0 \in C$, there exists at least a trajectory of T starting at p_0 that remains in C, or equivalently, if there exists $p \in T(p_0)$ such that $p \in C$. Finally, a function $V : \mathbb{R}^n \to \mathbb{R}$ is *non-increasing along* T *on* $W \subset \mathbb{R}^n$ if $V(p') \leq V(p)$ for all $p \in W$ and $p' \in T(p)$. We are ready to state the following result from [23], see also [52].

Theorem 1 *(Invariance principle for closed set-valued maps) Let T be a set-valued map closed at p, for all $p \in W \subset \mathbb{R}^n$, and let $V : \mathbb{R}^n \to \mathbb{R}$ be a continuous function non-increasing along T on W. Assume that the trajectory $p : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ of the set-valued map T takes values in W and is bounded. Then there exists $c \in \mathbb{R}$ such that*

$$p(\ell) \longrightarrow M \cap V^{-1}(c) \quad \textit{as} \quad \ell \to +\infty,$$

where M is the largest weakly positively invariant set in $\{p \in \overline{W} \mid \exists p' \in T(p) \text{ with } V(p') = V(p)\}$.

Designing emergent behaviors

In this section, we elaborate on the role played by the tools introduced in the previous sections for the design and analysis of motion coordination algorithms. We do not enter into technical details throughout the discussion, and rather refer to various works for further reference. Our intention is to provide a first step toward the establishment of a rigorous systems-theoretic approach to motion coordination algorithms for a variety of spatially-distributed tasks.

Given a network of identical agents equipped with motion control and communication capabilities, the following subsections contain various approaches to the study of distributed and coordinated motions. Loosely speaking, a first approach is based on the *design* of *gradient flows*: here a coordination task and a proximity graph are typically specified together with a proximity graph imposing a communication constraint. A second approach is based on the *analysis* of *emergent behaviors*: in this case a notion of neighboring agents and an interaction law between them is usually given. A third approach is based on the identification of meaningful local objective functions whose optimization helps the network achieve the desired global task. Finally, the last and fourth approach relies on the composition of basic behaviors. Next, we discuss each of these approaches in detail, and illustrate their applicability in specific coordination tasks.

Designing the coordination algorithm from the aggregate objective function

The first step of this approach consists of identifying a global and aggregate objective function which is relevant to the desired coordination task. Once this objective function is determined, the next step is to analyze its differentiable properties and compute its (generalized) gradient. With this information, it is possible to characterize its critical points, that is, the desired network configurations. The next step is to identify those proximity graphs that allow the computation of the gradient of the objective function in a spatially distributed manner. If any of these proximity graphs can be determined with the capabilities of the mobile network, then a control law for each agent simply consists of following the gradient of the aggregate objective function. By the invariance principle, such a coordination algorithm automatically guarantees convergence of the closed-loop network trajectories to the set of critical points.

Distortion and area problems

The coordination algorithms for the distortion problem and for the area problem developed in [23] are examples of this approach. Given Q a convex polygon and $R > 0$, we can prove that the functions $\mathcal{H}_{\mathrm{C}}$ and $\mathcal{H}_{\mathrm{area},R}$ are differentiable almost everywhere and

$$\frac{\partial \mathcal{H}_{\mathrm{C}}}{\partial p_i}(P) = 2\,\mathrm{M}(V_i(P))\,(\mathrm{CM}(V_i(P)) - p_i)\,, \tag{2a}$$

$$\frac{\partial \mathcal{H}_{\mathrm{area},R}}{\partial p_i}(P) = \int_{\mathrm{arc}(\partial V_{i,R}(P))} n_{B(p_i,R)}\,\phi\,, \tag{2b}$$

where $n_{B(p,R)}(q)$ denotes the unit outward normal to $B(p,R)$ at $q \in \partial B(p,R)$ and, for each $i \in \{1,\dots,n\}$, $\mathrm{arc}(\partial V_{i,R}(P))$ denotes the union of the arcs in $\partial V_{i,R}(P)$. The symbols $\mathrm{M}(W)$ and $\mathrm{CM}(W)$ denote, respectively, the mass and the center of mass with respect to ϕ of $W \subset Q$. The critical points $P \in Q^n$ of $\mathcal{H}_{\mathrm{C}}$ satisfy $p_i = \mathrm{CM}(V_i(P))$ for all $i \in \{1,\dots,n\}$. Such configurations are usually referred to as *centroidal Voronoi configurations*, see [46]. The critical points $P \in Q^n$ of $\mathcal{H}_{\mathrm{area},R}$ have the property that each p_i is a local optimum for the area covered by $V_{i,R} = V_i \cap B(p_i,R)$ at fixed V_i. We refer to such configurations as *area-centered Voronoi configurations.*

From equation (2a) it is clear that the gradient of $\mathcal{H}_{\mathrm{C}}$ is spatially distributed over $\mathcal{G}_{\mathrm{D}}$, whereas from equation (2b) we deduce that the gradient of $\mathcal{H}_{\mathrm{area},R}$ is spatially distributed over

$\mathcal{G}_{\mathrm{LD}}(R)$. The gradient flows of $\mathcal{H}_{\mathrm{C}}$ and of $\mathcal{H}_{\mathrm{area},R}$ correspond to the coordination algorithms "move-toward-the-centroid of own Voronoi cell" and "move in the direction of the (weighted) normal to the boundary of own cell," respectively. Figures 10 and 12 show an example of the execution of these algorithms. Figures 11 and 13 illustrate the adaptive properties of these algorithms with respect to agent arrivals and departures.

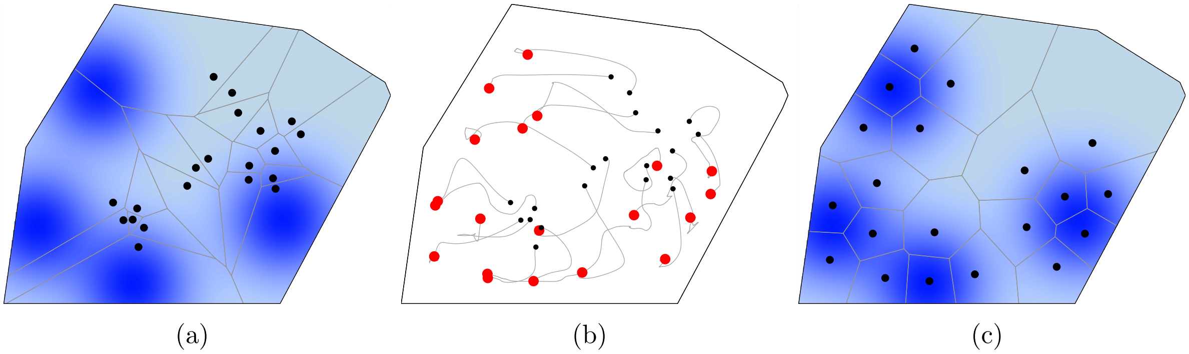


Figure 10. Distortion problem. Each one of the 20 mobile agents moves toward the centroid of its Voronoi cell. This strategy exactly corresponds to the network following the gradient $(2a)$ of the distortion function $\mathcal{H}_{\mathsf{C}}$. Areas of the convex polygon with greater importance are colored in darker blue. This coloring corresponds to the contour plot of the density function ϕ in the definition (1) of $\mathcal{H}_{\mathsf{C}}$. (a) and (c) show, respectively, the initial and final locations, with the corresponding Voronoi partitions. (b) illustrates the gradient descent flow.

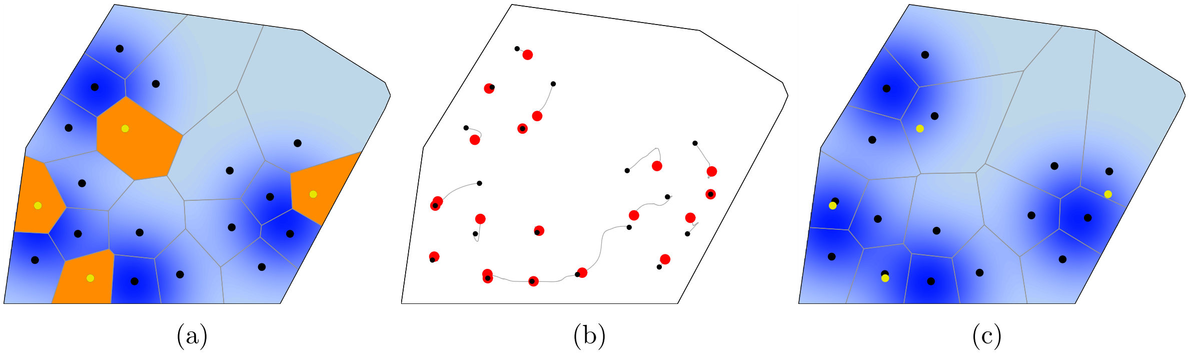


Figure 11. Adaptive network behavior under agent failures in the distortion problem. After the final configuration in Figure 10 is reached, four network agents (colored in yellow) fail and cease to provide coverage in their respective Voronoi cells (colored in orange). The rest of the network adapts to the new situation satisfactorily. (a) illustrates the location of the agents when the failures occur, and (c) shows the final location of the remaining agents. (b) illustrates the gradient descent flow since the failure occurred.

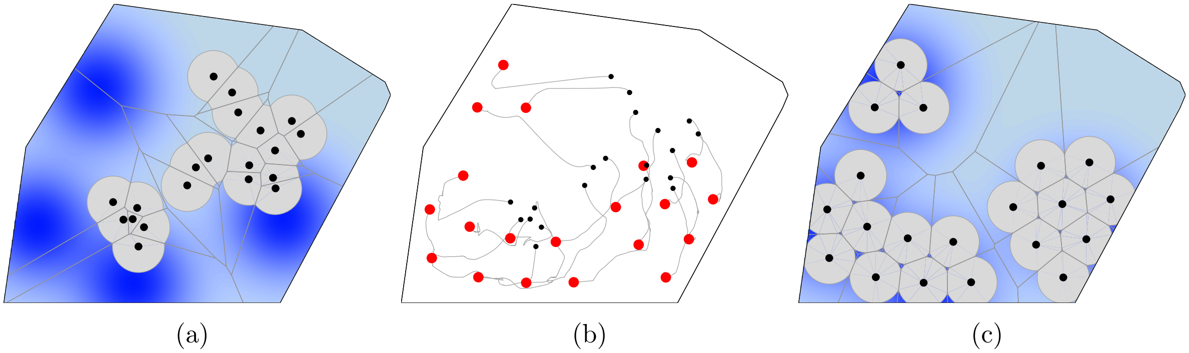


Figure 12. Area problem. Each one of the 20 mobile agents follows the gradient $(2b)$ of the area function $\mathcal{H}_{\mathsf{area},\frac{r}{2}}$. The density function ϕ, which specifies areas of greater importance, and the environment are the same as in Figure 10. (a) and (c) illustrate, respectively, the initial and final locations, with the corresponding Voronoi partitions. (b) illustrates the gradient descent flow. Each agent operates with a finite communication radius. For each agent i, the $\frac{r}{2}$-limited Voronoi cell $V_{i,\frac{r}{2}}(P)$ is plotted in light gray.

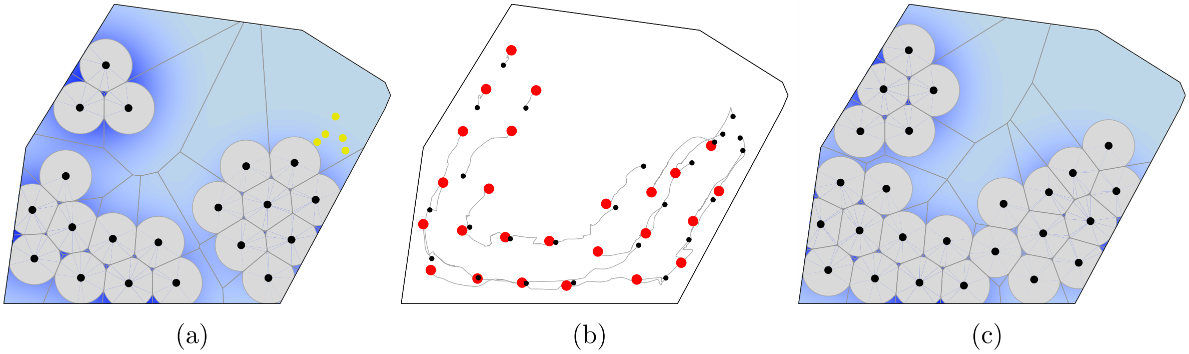


Figure 13. Adaptive network behavior under agent arrivals in the area problem. After the final configuration in Figure 12 is reached, five new agents (colored in yellow) enter the environment. The rest of the network adapts to the new situation satisfactorily. (a) illustrates the location of the agents when the arrival of the new agents occurs, and (c) shows the final location of the network. (b) illustrates the gradient descent flow from this event on.

Consensus

The asymptotic agreement algorithm developed in [33] to solve the consensus problem is another example of this approach. For a fixed undirected graph $G = (\{1, \dots, n\}, E)$, the function Φ_G is smooth, and its partial derivative takes the form

$$\frac{\partial \Phi_G}{\partial x} = Lx \,. \tag{3}$$

Clearly, this gradient is distributed with respect to the graph G itself. The implementation of the gradient control law leads to the algorithm $\dot{x}_i = \sum_{(i,j)\in E}(x_j - x_i)$, for $i \in \{1, \dots, n\}$, which asymptotically achieves average-consensus, that is, the final value upon which all agents agree can be proved to be equal to $\frac{1}{n}\sum_{i=1}^{n} x_i(0)$.

Cohesiveness

Another example of this approach are the various coordination algorithms developed in the literature to achieve cohesiveness [17, 20, 21]. For the complete graph $\mathcal{G}_{\text{complete}}$, the function $\mathcal{H}_{\text{cohe},\mathcal{G}_{\text{complete}}}$ is smooth almost everywhere and

$$\frac{\partial \mathcal{H}_{\text{cohe},\mathcal{G}_{\text{complete}}}}{\partial p_i}(P) = \sum_{j\neq i}^{n} \frac{\partial}{\partial p_i}\big(h(\|p_i - p_j\|)\big) = \sum_{p_j \in \mathcal{N}_{\mathcal{G}_{\text{disk}}(R_1),p_i}} \frac{\partial}{\partial p_i}\big(h(\|p_i - p_j\|)\big),$$

where we used the fact that dh/dR vanishes for $R \geq R_1$. According to the notions we introduced earlier, this gradient is spatially distributed over $\mathcal{G}_{\text{disk}}(R_1)$. The gradient descent algorithm guarantees that the network of agents asymptotically approaches the set of critical points of $\mathcal{H}_{\text{cohe},\mathcal{G}_{\text{complete}}}$.

Not always does the aggregate objective function enjoy the desirable property that its gradient is spatially distributed with respect to the required proximity graph. In other words, given an available information flow, the corresponding gradient algorithm can not always be computed. If this is the case, a possible approach is the following: (i) consider constant-factor approximations of the objective function, (ii) identify those approximations whose gradient is spatially distributed with respect to an appropriate proximity graph, and (iii) implement as coordination algorithm the one that makes each agent follow the gradient of the approximation.

Analyzing the coordinated behavior emerging from basic interaction laws

This approach consists of devising a simple control law, typically inspired by some sort of heuristic, that implemented over each agent of the network would reasonably perform the desired task. Once we have done this, we should (i) check that the resulting coordination algorithm is spatially distributed with regards to some appropriate proximity graph, and (ii) characterize its asymptotic convergence properties. One way of doing the latter is by finding an aggregate objective function that encodes the desired coordination task and by showing that this function is optimized along the execution of the coordination algorithm.

Move-away-from-closest-neighbor

Consider the coordination algorithm studied in [24] where each agent moves away from its closest neighbor (see Figure 14). This simple interaction law is spatially distributed over $\mathcal{G}_{\mathrm{D}}$. We can prove that along the evolution of the network, the aggregate cost function

$$\mathcal{H}_{\mathrm{SP}}(P) = \min_{i \neq j \in \{1,\dots,n\}} \left\{ \tfrac{1}{2}\|p_i - p_j\|, \operatorname{dist}(p_i, \partial Q) \right\}, \tag{4}$$

is monotonically non-decreasing. This function corresponds to the *non-interference problem*, where the network tries to maximize the coverage of the domain in such a way that the various communication radius of the agents do not overlap or leave the environment (because of interference). Under appropriate technical conditions, we can show that the critical points of $\mathcal{H}_{\mathrm{SP}}$ are configurations where each agent is at the incenter of its own Voronoi region (recall that the incenter set of a polygon is the set of centers of the maximum-radius spheres contained in the polygon).

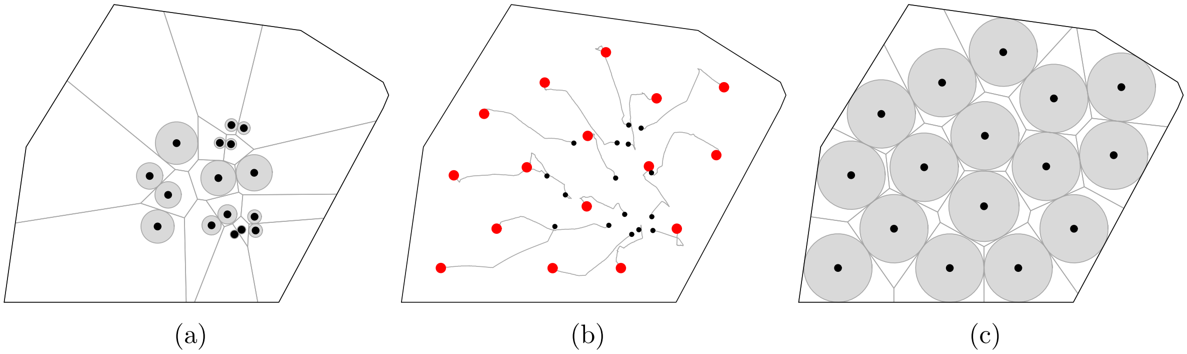


Figure 14. Non-interference problem. Each one of the 16 mobile agents moves away from its closest neighbor. The resulting network behavior maximizes the coverage of the environment in such a way that the various communication radius of the agents do not overlap or leave the domain. (a) and (c) illustrate, respectively, the initial and final locations, with corresponding Voronoi partitions. (b) illustrates the network evolution. For each agent i, the ball of maximum radius contained in the Voronoi cell $V_i(P)$ and centered at p_i is plotted in light gray.

Flocking

Consider the coordination algorithm analyzed in [16] for the flocking problem. Roughly speaking, flocking consists of agreeing over the direction of motion by the agents in the network. Let $\mathcal{G}$ be a proximity graph. Now, consider the coordination algorithm where each agent performs the following steps: (i) detects its neighbors' (according to $\mathcal{G}$) heading; (ii) computes the average of its neighbors' heading and its own heading, and (iii) updates its heading to the computed average. Clearly, this algorithm is spatially distributed over $\mathcal{G}$. Moreover, assuming that $\mathcal{G}$ remains connected throughout the evolution, we can show that the agents asymptotically acquire the same heading.

Designing the coordination algorithm from local objective functions

This approach has common elements with the two approaches presented previously. Now, to derive a control law for each specific agent, we assume that the neighboring agents of that agent, or some spatial structure attributed to it, remain fixed. We then define a local objective function, which is somehow related with the global aggregate objective function encoding the desired coordination task, and devises a control law to optimize it. The specific control strategy might be heuristically derived or arise naturally from the gradient information of the local objective function. Once the coordination algorithm is set up, it should be checked that the algorithm is spatially distributed and its asymptotic convergence properties should be characterized.

Non-interference problem

Consider the aggregate objective function $\mathcal{H}_{\mathrm{SP}}$ defined in equation (4). Consider the alternative expression,

$$\mathcal{H}_{\mathrm{SP}}(P) = \min_{i \in \{1,\dots,n\}} \mathrm{sm}_{V_i(P)}(p_i)\,,$$

where $\mathrm{sm}_W(p)$ is the distance from p to the boundary of the convex polygon W, that is, $\mathrm{sm}_W(p) = \mathrm{dist}(p, \partial W)$. Now, for $i \in \{1, \dots, n\}$, consider $\mathrm{sm}_{V_i(P)}$ as a local objective function. Assuming that the Voronoi cell $V_i(P)$ remains fixed, then we can implement the (generalized) gradient ascent of $\mathrm{sm}_{V_i(P)}$ as the control law for the agent p_i. We can show [24] that this interaction law precisely corresponds to the strategy "move-away-from-closest-neighbor" discussed earlier (see section entitled "Move-away-from-closest-neighbor"). A related strategy consists of each agent moving toward the incenter of its own Voronoi cell. The latter strategy can also be shown to make $\mathcal{H}_{\mathrm{SP}}$ monotonically non-decreasing and to enjoy analogous asymptotic convergence properties.

Worst-case problem

Consider the aggregate objective function

$$\mathcal{H}_{\mathrm{DC}}(P) = \max_{q \in Q} \big\{ \min_{i \in \{1,\dots,n\}} \|q - p_i\| \big\} = \max_{i \in \{1,\dots,n\}} \mathrm{lg}_{V_i(P)}(p_i)\,,$$

where $\mathrm{lg}_W(p)$ is the maximum distance from p to the boundary of the convex polygon W, that is, $\mathrm{lg}_W(p) = \max_{q \in W} \|q - p_i\|$. Now, for $i \in \{1, \dots, n\}$, consider $\mathrm{lg}_{V_i(P)}$ as a local objective function. Assuming that the Voronoi cell $V_i(P)$ remains fixed, then we can implement the (generalized) gradient descent of $\mathrm{lg}_{V_i(P)}$ as the control law for the agent p_i. We can show [24] that this interaction law precisely corresponds to the strategy "move-toward-the-furthest-away-vertex-in-own-cell." A related strategy consists of each agent moving toward the circumcenter of its own Voronoi cell (recall that the circumcenter of a polygon is the center of the minimum-radius sphere that contains it). Both strategies can be shown to make $\mathcal{H}_{\mathrm{DC}}$ monotonically non-increasing and enjoy similar asymptotic convergence properties. These ideas can be combined in other settings with different capabilities of the mobile agents, for instance, in higher dimensional spaces (see Figure 15).

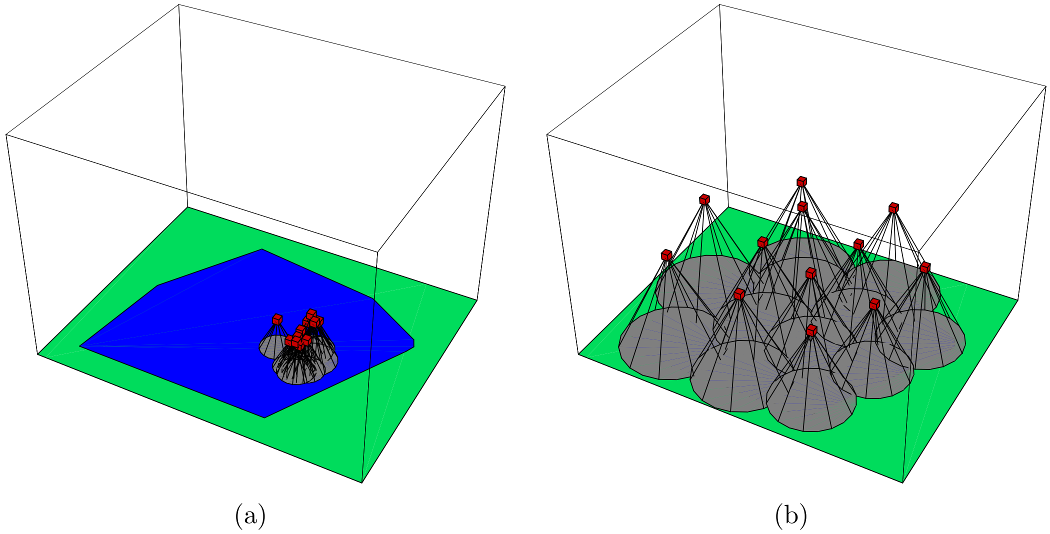


Figure 15. Worst-case scenario. The network tries to maximize the coverage (illumination) of a convex polygon. Each one of the 12 mobile agents illuminates a vertical cone with a fixed and common aspect ratio. Each agent determines its Voronoi region within the planar polygon (the same as in Figure 14). Then, each agent moves its horizontal position toward the circumcenter of its Voronoi cell and its vertical position to the minimal height spanning its own Voronoi cell. (a) and (b) illustrate, respectively, the initial and final locations.

Rendezvous

Let $\mathcal{G}$ be a proximity graph. Consider the Circumcenter Algorithm over $\mathcal{G}$, where each agent performs the following steps: (i) detects its neighbors according to $\mathcal{G}$; (ii) computes the circumcenter of the point set comprised of its neighbors and of itself, and (iii) moves toward this circumcenter while maintaining connectivity with its neighbors. To maintain connectivity, the allowable motion of each agent is conveniently restricted (see [26, 27, 29] for further details).

Note that with step (ii), assuming that all other agents remain fixed, each agent minimizes the local objective function given by the maximum distance from the agent to all its neighbors in the proximity graph $\mathcal{G}$. By construction, this coordination algorithm is spatially distributed over the proximity graph $\mathcal{G}$. Moreover, we can prove that the evolution of the aggregate objective function V_{diam} is monotonically non-increasing along the execution of the Circumcenter Algorithm. Using the invariance principle for closed algorithms, we can indeed characterize the asymptotic correctness properties of the Circumcenter Algorithm over $\mathcal{G}$. See Figure 16 for an illustration of its execution.

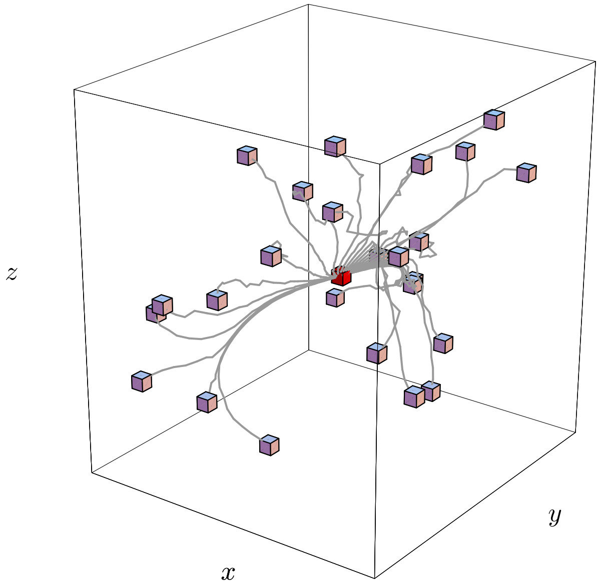


Figure 16. Circumcenter Algorithm in Euclidean space. Each one of the 25 mobile agents move toward the circumcenter of the point set comprised of its neighbors and of itself. The resulting network behavior asymptotically achieves rendezvous at a point. Indeed, the invariance principle allows us to establish the algorithm's correctness under fairly general conditions. For instance, in this figure at each time step each agent randomly selects $\mathcal{G}_{\mathsf{disk}}(r)$ or $\mathcal{G}_{\mathsf{LD}}(\frac{r}{2})$ to compute its set of neighbors.

Designing the coordination algorithm by composing different behaviors

This final approach builds on the methods presented above. An idea for the composition of behaviors is to implement one coordination algorithm on most of the network agents and a second coordination algorithm on the other agents. Coupling two algorithms in this parallel fashion results in interesting overall network behaviors. For example, we may prescribe an open-loop motion on some of the network agents (for instance, specifying that some particular agents must stay fixed or follow a desired path) and implement a feedback law for the others. Examples of this approach include (1) the formation control strategy in [27] to make the network form a straight line, and (2) the leader-following algorithm given in [16] to make the network flock in a pre-specified direction. Along these lines, it is interesting to explore more general parallel, serial and hierarchical approaches to the composition of behaviors.

Conclusions

We believe that the set of recent tools (proximity graphs, spatially distributed maps, aggregate objective functions, circulant matrices, and invariance principles) surveyed in this paper are important in distributed motion coordination. These technical tools play an key role in the various design approaches to coordination algorithms reported here, see Table 1. We believe that the coming years will witness an intense development of the field of distributed coordination and of its practical use in applications for multiple vehicles and sensor networks.

Agent motion direction	**Formal description**	**Distributed information**	**Lyapunov function**	**Asymptotic convergence**	**Ref.**
centroid of Voronoi cell	$\dot{p}_i = \mathrm{CM}(V_i(P)) - p_i$	Voronoi neighbors	$\mathcal{H}_{\mathrm{C}}$	centroidal Voronoi configurations	[22]
weighted average normal of $\frac{r}{2}$-limited Voronoi cell	$\dot{p}_i = \int_{\mathrm{arc}(\partial V_{i,\frac{r}{2}}(P))} n_{B(p_i,\frac{r}{2})}\, \phi$	r-disk neighbors	$\mathcal{H}_{\mathrm{area},\frac{r}{2}}$	area-centered Voronoi configurations	[23]
average of neighbors	$\dot{p}_i = \sum_{j \in \mathcal{N}_G(i)} (p_j - p_i)$	neighbors in fixed G	Φ_G	Consensus	[33]
away from closest neighbor	$\dot{p}_i = \mathrm{Ln}(\partial\, \mathrm{sm}_{V_i(P)})(P)$	Voronoi neighbors	$\mathcal{H}_{\mathrm{SP}}$	Incenter Voronoi configurations	[24]
furthest-away vertex in Voronoi cell	$\dot{p}_i = -\,\mathrm{Ln}(\partial\, \mathrm{lg}_{V_i(P)})(P)$	Voronoi neighbors	$\mathcal{H}_{\mathrm{DC}}$	Circumcenter Voronoi configurations	[24]
circumcenter of neighbors' and own position	$p_i(t+1) = p_i(t) + \lambda_i^* \cdot (\mathrm{CC}(\mathrm{M}_i) - p_i)$	r-disk neighbors	V_{diam}	rendezvous	[26]

Table 1. Summary of motion coordination algorithms. The tools presented throughout the paper play a key role in the design and the analysis of the network behavior resulting from these coordination algorithms. In the interest of brevity, we refer to the corresponding references for the notation employed.

Acknowledgments

This material is based on work supported in part by ONR YIP Award N00014-03-1-0512, NSF SENSORS Award IIS-0330008 and NSF CAREER Award ECS-0546871. Sonia Martínez's work was supported in part by a Fulbright Postdoctoral Fellowship from the Spanish Ministry of Education and Science.

References

[1] M. B. Miller and B. L. Bassler, "Quorum sensing in bacteria," *Annual Review of Microbiology*, vol. 55, pp. 165–199, 2001.

[2] A. Okubo, "Dynamical aspects of animal grouping: swarms, schools, flocks and herds," *Advances in Biophysics*, vol. 22, pp. 1–94, 1986.

[3] J. K. Parrish, S. V. Viscido, and D. Grunbaum, "Self-organized fish schools: an examination of emergent properties," *Biological Bulletin*, vol. 202, pp. 296–305, 2002.

[4] A. R. Sinclair, *The African Buffalo, A Study of Resource Limitation of Population.* Chicago, IL: The University of Chicago Press, 1977.

[5] S. Gueron and S. A. Levin, "Self-organization of front patterns in large wildebeest herds," *Journal of Theoretical Biology*, vol. 165, pp. 541–552, 1993.

[6] G. W. Barlow, "Hexagonal territories," *Animal Behavior*, vol. 22, pp. 876–878, 1974.

[7] T. D. Seeley and S. C. Buhrman, "Group decision-making in swarms of honey bees," *Behavioral Ecology and Sociobiology*, vol. 45, pp. 19–31, 1999.

[8] K. J. Stewart and A. H. Harcourt, "Gorillas vocalizations during rest periods - signals of impending departure," *Behaviour*, vol. 130, pp. 29–40, 1994.

[9] S. Boinski and A. F. Campbell, "Use of trill vocalizations to coordinate troop movement among whitefaced capuchins - a 2nd field-test," *Behaviour*, vol. 132, pp. 875–901, 1995.

[10] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin, "Effective leadership and decision-making in animal groups on the move," *Nature*, vol. 433, pp. 513–516, 2005.

[11] L. Conradt and T. J. Roper, "Group decision-making in animals," *Nature*, vol. 421, no. 6919, pp. 155–158, 2003.

[12] U.S. Fish and Wildlife Service, `http://www.fws.gov`.

[13] I. Suzuki and M. Yamashita, "Distributed anonymous mobile robots: Formation of geometric patterns," *SIAM Journal on Computing*, vol. 28, no. 4, pp. 1347–1363, 1999.

[14] C. Belta and V. Kumar, "Abstraction and control for groups of robots," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 865–875, 2004.

[15] E. W. Justh and P. S. Krishnaprasad, "Equilibria and steering laws for planar formations," *Systems & Control Letters*, vol. 52, no. 1, pp. 25–38, 2004.

[16] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[17] H. Tanner, A. Jadbabaie, and G. J. Pappas, "Stability of flocking motion," tech. rep., Department of Computer and Information Science, University of Pennsylvania, Jan. 2003.

[18] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, 2006. To appear.

[19] E. Klavins, R. Ghrist, and D. Lipsky, "A grammatical approach to self-organizing robotic systems," *IEEE Transactions on Automatic Control*, 2005. To appear.

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

[21] P. Ögren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, 2004.

[22] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[23] J. Cortés, S. Martínez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM. Control, Optimisation & Calculus of Variations*, vol. 11, pp. 691–719, 2005.

[24] J. Cortés and F. Bullo, "Coordination and geometric optimization via distributed dynamical systems," *SIAM Journal on Control and Optimization*, vol. 44, no. 5, pp. 1543–1574, 2005.

[25] E. Frazzoli and F. Bullo, "Decentralized algorithms for vehicle routing in a stochastic time-varying environment," in *IEEE Conf. on Decision and Control*, (Paradise Island, Bahamas), pp. 3357–3363, Dec. 2004.

[26] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, "Distributed memoryless point convergence algorithm for mobile robots with limited visibility," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 5, pp. 818–828, 1999.

[27] J. Lin, A. S. Morse, and B. D. O. Anderson, "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, New York: Springer Verlag, 2004.

[28] Z. Lin, M. Broucke, and B. Francis, "Local control strategies for groups of mobile autonomous agents," *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 622–629, 2004.

[29] J. Cortés, S. Martínez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Transactions on Automatic Control*, vol. 51, no. 6, 2006. To appear.

[30] A. M. Bruckstein, N. Cohen, and A. Efrat, "Ants, crickets, and frogs in cyclic pursuit," Tech. Rep. 9105, Technion – Israel Institute of Technology, Haifa, Israel, July 1991. Center for Intelligent Systems.

[31] J. A. Marshall, M. E. Broucke, and B. A. Francis, "Formations of vehicles in cyclic pursuit," *IEEE Transactions on Automatic Control*, vol. 49, no. 11, pp. 1963– 1974, 2004.

[32] V. Sharma, M. Savchenko, E. Frazzoli, and P. Voulgaris, "Time complexity of sensor-based vehicle routing," in *Robotics: Science and Systems* (S. Thrun, G. Sukhatme, S. Schaal, and O. Brock, eds.), Cambridge, MA: MIT Press, 2005.

[33] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[34] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.

[35] W. Ren and R. W. Beard, "Consensus seeking in multi-agent systems using dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[36] V. J. Lumelsky and K. R. Harinarayan, "Decentralized motion planning for multiple mobile robots: the cocktail party model," *Autonomous Robots*, vol. 4, no. 1, pp. 121–135, 1997.

[37] R. C. Arkin, *Behavior-Based Robotics.* Cambridge, MA: MIT Press, 1998.

[38] T. Balch and L. E. Parker, eds., *Robot Teams: From Diversity to Polymorphism.* Natick, MA: A K Peters Ltd., 2002.

[39] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed scalable solution to the area coverage problem," in *International Conference on Distributed Autonomous Robotic Systems (DARS02)*, (Fukuoka, Japan), pp. 299–308, June 2002.

[40] J. W. Jaromczyk and G. T. Toussaint, "Relative neighborhood graphs and their relatives," *Proceedings of the IEEE*, vol. 80, no. 9, pp. 1502–1517, 1992.

[41] N. A. Lynch, *Distributed Algorithms.* San Mateo, CA: Morgan Kaufmann Publishers, 1997.

[42] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific, 1997.

[43] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*. New York: Springer Verlag, 2 ed., 2000.

[44] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley Series in Probability and Statistics, New York: John Wiley, 2 ed., 2000.

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

[46] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: Applications and algorithms," *SIAM Review*, vol. 41, no. 4, pp. 637–676, 1999.

[47] R. Diestel, *Graph Theory*, vol. 173 of *Graduate Texts in Mathematics*. New York: Springer Verlag, 2 ed., 2000.

[48] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. Philadelphia, PA: SIAM, 2001.

[49] P. J. Davis, *Circulant Matrices*. Providence, RI: American Mathematical Society, 2 ed., 1994.

[50] S. Martínez, F. Bullo, J. Cortés, and E. Frazzoli, "On synchronous robotic networks – Part I: Models, tasks and complexity notions. & Part II: Time complexity of rendezvous and deployment algorithms," *IEEE Transactions on Automatic Control*, Apr. 2005. Submitted.

[51] D. G. Luenberger, *Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley, 2 ed., 1984.

[52] A. R. Teel, "Nonlinear systems: discrete-time stability analysis." Lecture Notes, University of California at Santa Barbara, 2004.

Sonia Martínez received the Licenciatura degree in mathematics from the Universidad de Zaragoza, Zaragoza, Spain, in 1997, and the Ph.D. degree in engineering mathematics from the Universidad Carlos III de Madrid, Madrid, Spain, in 2002. From September 2002 to September 2003, she was a Visiting Assistant Professor of Applied Mathematics at the Technical University of Catalonia, Spain. She has been a Postdoctoral Fulbright Fellow at the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign (from October 2003 to August 2004), and at the Center for Control, Dynamical Systems, and Computation, University of California, Santa Barbara (from September 2004 to November 2005). She is currently an Assistant Professor at the Department of Mechanical and Aerospace Engineering, University of California at San Diego. She received the Best Student Paper award at the 2002 IEEE Conference on Decision and Control for her work on underactuated mechanical control systems.

Jorge Cortés (Department of Applied Mathematics and Statistics, University of California at Santa Cruz, 1156 High Street, Santa Cruz, CA 95064, fax 1-831-459-4829, jcortes@ucsc.edu) received the Licenciatura degree in mathematics from the Universidad de Zaragoza, Zaragoza, Spain, in 1997, and the Ph.D. degree in engineering mathematics from the Universidad Carlos III de Madrid, Madrid, Spain, in 2001. He has been a Postdoctoral Research Associate at the Systems, Signals and Control Department, University of Twente, Enschede, The Netherlands (from January to June 2002), and at the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign (from August 2002 to September 2004). He is currently an Assistant Professor at the Department of Applied Mathematics and Statistics, Baskin School of Engineering, University of California at Santa Cruz. He is the author of the book "Geometric, control and numerical aspects of nonholonomic systems" (New York: Springer Verlag, 2002).

Francesco Bullo received the Laurea degree in Electrical Engineering from the University of Padova, Italy, in 1994, and the Ph.D. degree in Control and Dynamical Systems from the California Institute of Technology in 1999. From 1998 to 2004, he was an Assistant Professor with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign. He is currently an Associate Professor at the Mechanical and Environmental Engineering Department, University of California, Santa Barbara. He is the coauthor (with Andrew D. Lewis) of the book "Geometric Control of Mechanical Systems" (New York: Springer Verlag, 2004, 0-387-22195-6). He is currently an associate editor for the IEEE Transactions on Automatic Control, SIAM Journal of Control and Optimization, and ESAIM: Control, Optimization, and the Calculus of Variations.