# Motion Coordination with Distributed Information

## The challenge of obtaining global behavior from local interaction

Sonia Martínez Jorge Cortés Francesco Bullo

*Think globally, act locally*

René J. Dubos, 1972

Motion coordination is a remarkable phenomenon in biological systems and an extremely useful tool for groups of vehicles, mobile sensors, and embedded robotic systems. For many applications, teams of mobile autonomous agents need the ability to deploy over a region, assume a specified pattern, rendezvous at a common point, or move in a synchronized manner. These coordination tasks must often be achieved with minimal communication between agents and, therefore, with limited information about the global state of the system.

The scientific motivation for studying motion coordination is the analysis of emergent and self-organized swarming 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 among molecules and cells [1] to the behavioral ecology of animal groups [2]. Flocks of birds and schools of fish can travel in formation and act as one unit (see [3], and Figures 1 and 2), allowing these animals to defend themselves against predators and protect their territories. Wildebeest and other animals exhibit complex collective behaviors when migrating, such as obstacle avoiding, leader election, and formation keeping (see [4, 5] and Figure 3). Certain foraging behaviors include individual animals partitioning their environment into nonoverlapping 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 coordinated behaviors emerge despite the fact that each individual lacks global knowledge of the network state and can plan its motion by observing only its closest neighbors.

At the same time, an engineering motivation for studying motion coordination stems from increasing interest in groups of embedded systems, such as multivehicle and sensor networks. Indeed, groups of autonomous agents with computing, communication, and mobility capabilities are expected to 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, research on cooperative control has increased tremendously over the last few years. Key aspects of distributed, or leader-less, motion coordination include pattern formation [12, 13, 14], flocking [15, 16, 17], self-assembly [18], swarm aggregation [19], gradient climbing [20], deployment and task allocation [21, 22, 23, 24], rendezvous [25, 26, 27, 28], cyclic pursuit [29, 30], vehicle routing [31], and consensus [32, 33, 34]. Heuristic approaches to the design of interaction rules and emergent behaviors are investigated in the literature on behavior-based robotics [35, 36, 37, 38].

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

First, we are interested in characterizing the distributed character of cooperative strategies. Our approach is based on the notion of proximity graph, drawn from computational geometry [39]. Proximity graphs model agent-to-agent interactions that depend on the agents' locations in space, as in wireless or line-of-sight communication. Proximity graphs thus facilitates the modeling of information flow among mobile agents.

Second, we consider representations of motion-coordination tasks that facilitate the analysis of coordination algorithms. We discuss aggregate objective functions from geometric optimization for tasks such as deployment, rendezvous, cohesiveness, and consensus.

We use nonsmooth analysis to identify the extreme points of the aggregate objective functions, which typically encode the desired network configurations.

Third, we discuss techniques for assessing the performance of coordination algorithms. In particular, we use a combination of system-theoretic and linear-algebraic tools to establish stability and convergence of motion-coordination algorithms. This treatment includes methods from circulant and Toeplitz tridiagonal matrices as well as a version of the invariance principle for set-valued discrete-time dynamical systems.

Finally, we focus on the design of distributed coordination algorithms for specific tasks. Given a coordination task to be performed by the network as well as a proximity graph representing communication constraints, a first approach is based on gradient flows. 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 given. A third approach is based on optimizing local objective functions to achieve the desired global task. The last approach relies on the composition of simple behaviors to design more complex strategies. We apply these approaches to several examples of coordination algorithms developed in the literature.

## Proximity Graphs and Spatially Distributed Maps

An algorithm is distributed, as opposed to centralized, if the algorithm relies on local information rather than global knowledge. Precise notions of distributed algorithms for networks with fixed topology are given in the literature on automata theory and parallel computing [40, 41]. In the context of networks of mobile agents, where the topology changes dynamically, we borrow the notion of proximity graph from computational geometry to arrive at a satisfactory definition of spatially distributed algorithms.

## Basic geometric notions

A *partition* of a set $S \subset \mathbb{R}^2$ is a subdivision of $S$ into components with disjoint interiors. Given $S \subset \mathbb{R}^2$ and a set $\mathcal{P} = \{p_1, \dots, p_n\} \subset S$ of $n$ distinct points, the *Voronoi partition [42] of $S$ generated by $\mathcal{P}$* is the collection of sets $\{V_1(\mathcal{P}), \dots, V_n(\mathcal{P})\}$ defined by $V_i(\mathcal{P}) = \{q \in S \mid \|q - p_i\| \leq \|q - p_j\| \text{ for all } p_j \in \mathcal{P}\}$. We refer to $V_i(\mathcal{P})$ as the *Voronoi cell* of $p_i$. A Voronoi partition is depicted in Figure 5(a); see also the sidebar "Georgy Fedoseevich Voronoy and His Diagrams."

For $p \in \mathbb{R}^2$ and $r > 0$, let $B(p, r)$ denote the closed ball in $\mathbb{R}^2$ centered at $p$ with radius $r$, and let $\partial B(p, r)$ denote the boundary of $B(p, r)$. For a set $\mathcal{P} = \{p_1, \dots, p_n\} \subset S$ of $n$ distinct points, the *$r$-limited Voronoi partition* inside $S$ is the collection of sets $\{V_{1,r}(\mathcal{P}), \dots, V_{n,r}(\mathcal{P})\}$, where $V_{i,r}(\mathcal{P}) = V_i(\mathcal{P}) \cap B(p_i, r)$. This name is justified by the fact that the $r$-limited Voronoi partition is the Voronoi partition of $\cup_i B(p_i, r) \cap S$. Figure 5(b) illustrates this geometric construction. We refer to $V_{i,r}(\mathcal{P})$ as the *$r$-limited Voronoi cell* of $p_i$.

## Proximity graphs and their properties

A *proximity graph* $\mathcal{G}$ is a function that associates to finite collections of distinct points in Euclidean space a graph whose vertices are the points and whose edges depend on the relative positions of the points. More precisely, given a set $\mathcal{P} = \{p_1, \dots, p_n\} \subset \mathbb{R}^d$ of $n$ distinct points, the proximity graph $\mathcal{G}$ at $\mathcal{P}$, denoted by $\mathcal{G}(\mathcal{P})$, is an undirected graph with vertex set $\mathcal{P}$ and with edge set $\mathcal{E}_{\mathcal{G}}(\mathcal{P})$, where $\mathcal{E}_{\mathcal{G}}(\mathcal{P}) \subseteq \{\{p, q\} \in \mathcal{P} \times \mathcal{P} \mid p \neq q\}$, so that self-loops are not allowed. A related notion is that of state-dependent graphs, see [43]. Proximity graphs provide a natural means for modeling the interconnection topology of a network of robotic agents. In general, the interconnection topology depends on the agents' relative locations as well as the agents' sensing or communication capabilities. An alternative approach to dynamic networks consists of regarding the interconnection topology of the network as a known time-dependent function, independent of the state evolution of the individual robotic agents [15, 32, 33, 34].

The following proximity graphs are discussed in [22, 39, 42]:

(i) the *$r$-disk graph* $\mathcal{G}_{\text{disk}}(r)$, where $r > 0$, in which two agents are neighbors if their locations are 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* $\mathcal{G}_{\text{D}}$, in which 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 $V_i(\mathcal{P}) \cap V_j(\mathcal{P}) \neq \emptyset$. Note that the intersection of two Voronoi cells may only include portions of the boundary of each Voronoi cell;

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

(iv) the *visibility graph* $\mathcal{G}_{\mathrm{vis},Q}$, where $Q$ is a set in $\mathbb{R}^d$, in which two agents are neighbors if their positions are visible to each other, that is, $\{p_i, p_j\} \in \mathcal{E}_{\mathcal{G}_{\mathrm{vis},Q}}(\mathcal{P})$ if the closed segment from $p_i$ to $p_j$ is contained in $Q$.

Figures 6 and 7 illustrate these four proximity graphs. Additionally, we introduce the *complete graph* $\mathcal{G}_{\mathrm{complete}}$, in which any two agents are neighbors. This notion allows us to model fully interconnected networks. The connectivity properties of these graphs play an important role in coordination problems, see [22], but are outside of the scope of this article. Next, we define the set of neighbors of a vertex in a proximity graph. Given a set $\mathcal{P} = \{p_1, \dots, p_n\} \subset \mathbb{R}^d$ and a proximity graph $\mathcal{G}$, the *set of neighbors* of $p_i \in \mathcal{P}$ according to $\mathcal{G}$ is

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

## 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 the tuple $(p_1, \dots, p_n) \in$

$(\mathbb{R}^d)^n$ and the associated point set $\{p_1, \dots, p_n\} \subset \mathbb{R}^d$; we denote both quantities by $P$. An exposition that does not rely on this assumption is given in [28].

Given a set $Y$, consider a map $T : (\mathbb{R}^d)^n \to Y^n$. Given a proximity graph $\mathcal{G}$, we say that $T$ is *spatially distributed over* $\mathcal{G}$ if the $j$th component $T_j$ of the map $T$ evaluated at any $P = (p_1, \dots, p_n) \in (\mathbb{R}^d)^n$ is a function only of $p_j$ and of the vertices in $\mathcal{G}(P)$ that are neighbors of $p_j$. In other words, through information about the location of its neighbors according to $\mathcal{G}$, agent $j$ has sufficient information to compute the value $T_j(P)$.

It is also useful to determine when a given proximity graph contains sufficient information to compute a second proximity graph. Given proximity graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, we say that $\mathcal{G}_1$ is *spatially distributed over* $\mathcal{G}_2$ if each agent, when informed about the location of its neighbors according to $\mathcal{G}_2$, has sufficient information to determine its set of neighbors according to $\mathcal{G}_1$. As a first example, 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 has sufficient information to determine its neighbors according to any proximity graph. As a second example, since two points having intersecting $r$-limited Voronoi cells must be less than $2r$ apart, the $r$-limited Delaunay graph $\mathcal{G}_{\mathrm{LD}}(r)$ is spatially distributed over the $(2r)$-disk graph $\mathcal{G}_{\mathrm{disk}}(2r)$. This fact plays a role in certain coordination algorithms where agents who need to know the location of their neighbors according to the graph $\mathcal{G}_{\mathrm{LD}}(r)$ can establish that information through their knowledge of their neighbors in $\mathcal{G}_{\mathrm{disk}}(2r)$.

# Encoding Coordination Tasks

Our second goal is to develop methods for expressing motion-coordination tasks. Achieving a coordination task corresponds to moving the agents and changing their state in order to maximize or minimize appropriate objective functions. The aggregate behavior of the entire mobile network is thus evaluated by means of appropriate objective functions. In what follows, we illustrate how locational optimization functions from geometric optimization are helpful in formalizing various network objectives. We discuss deployment problems in convex and nonconvex environments as well as consensus, rendezvous, and cohesiveness problems; appropriate objective functions are considered for each scenario. 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 a given environment to achieve maximum coverage. The notion of "coverage" can be defined in many possible ways, as illustrated in the following discussion.

Let $Q \subset \mathbb{R}^d$ be a convex polytope. A *density function* $\phi : Q \to [0, \infty)$ is a summable function. The function $\phi$ quantifies the relative importance of different points in the environment; for example, up to a rescaling, $\phi$ might measure the probability that an event of interest takes place in the environment. A *performance function* $f : [0, \infty) \to \mathbb{R}$ is a nonincreasing and piecewise differentiable function with finite jump discontinuities. This function describes the utility of placing an agent at a certain distance from a given location in the environment. To illustrate this notion, consider a sensing scenario in which the agents are equipped with acoustic sensors that measure 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 $i$th 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 $\phi$ and a performance function $f$, we are interested in maximizing the expected value of the coverage performance provided by the group of agents for points in the environment $Q$. We thus define $\mathcal{H} : Q^n \to \mathbb{R}$ 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 of the locations $p_1, \dots, p_n$, $\mathcal{H}$ is an aggregate objective function. This objective function is commonly studied in locational optimization [22, 44]. We seek to find local maximizers of $\mathcal{H}$.

Different choices of performance function give rise to different aggregate objective functions with particular features. We now examine some relevant cases:

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

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

where $J(W,p)$ is the polar moment of inertia of the set $W \subset Q$ about the point $p$, see [45]. In signal compression, $-\mathcal{H}_{\mathrm{C}}$ is the distortion function, while the same function appears in many disciplines such as facility location, numerical integration, and clustering analysis, see [44].

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

$$\mathcal{H}_{\mathrm{area},R}(P) = \mathrm{area}_{\phi}(\cup_{i=1}^{n} B(p_i, R)), \tag{3}$$

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

## Aggregate objective functions for visibility-based deployment

Given a nonconvex polytope $Q \subset \mathbb{R}^d$ 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

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

In two dimensions, the function $\mathcal{H}_{\mathrm{vis}}$ measures the area of the subset of $Q$ composed of points that are visible from at least one of the agents located at $p_1, \dots, p_n$. Therefore, we seek to find maximizers of $\mathcal{H}_{\mathrm{vis}}$. By including a density function $\phi : Q \to [0, \infty)$ in the definition of $\mathcal{H}_{\mathrm{vis}}$, more importance can be assigned to some regions of the environment (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 $G$ [46] is the $n \times n$ matrix with entries

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

where degree($i$) is the number of neighbors of node $i$. The Laplacian matrix is symmetric, positive semi-definite, and singular, and has rank $n - 1$ if and only if $G$ is connected. Following [32], we define the *disagreement function* or *Laplacian potential* $\Phi_G : \mathbb{R}^n \to [0, \infty)$ associated with $G$ by

$$\Phi_G(x) = x^T L x = \frac{1}{2} \sum_{\{i,j\} \in E} (x_j - x_i)^2 . \tag{4}$$

For $i \in \{1, \dots, n\}$, the variable $x_i$ is associated with the agent $p_i$. The variable $x_i$ can represent physical quantities such as 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 every pair of neighboring nodes in the graph $G$ agree. Therefore, $\Phi_G(x)$ 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 a spatial version of consensus, which we refer to as rendezvous.

## Aggregate objective function for rendezvous

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

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

It is clear that $V_{\mathrm{diam}}(P) = 0$ if and only if $p_i = p_j$ for all $i, j \in \{1, \dots, n\}$. Therefore, each global minimizer of $V_{\mathrm{diam}}$ corresponds to a network configuration in which the agents rendezvous. The map $V_{\mathrm{diam}} : (\mathbb{R}^d)^n \to [0, \infty)$ is locally Lipschitz (with Lipschitz constant 1) and invariant under permutations of its arguments.

## Aggregate objective functions for cohesiveness

Let us consider one final example of an aggregate objective function that encodes a motion-coordination task. A *repulsion/attraction function* $h : (0, \infty) \to \mathbb{R}$ 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 every point 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 $R \geq R_1$. Assuming that $h$ is constant for all sufficiently large distances models limited interactions between agents. Figure 8 illustrates a typical repulsion/attraction function.

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

$$\mathcal{H}_{\text{cohe},\mathcal{G}}(P) = \sum_{\{p_i,p_j\}\in\mathcal{E}_{\mathcal{G}}(P)} h(\|p_i - p_j\|), \tag{5}$$

where $h$ is a repulsion/attraction function. The minimizers of $\mathcal{H}_{\text{cohe},\mathcal{G}}$ correspond to cohesive network configurations. Specifically, for groups of 2 or 3 agents, minimizers of $\mathcal{H}_{\text{cohe},\mathcal{G}}$ are configurations in which the distance between all neighboring agents are within the interval $[R_*, R'_*]$. This objective function and its variations, which are used in [19] and [20] for the complete graph and in [16] for the $r$-disk graph, can be employed to guarantee collision avoidance and cohesiveness of the mobile network.

## Correctness and Performance Analysis of Coordination Algorithms

In this section we briefly mention techniques for analyzing cooperative control problems. A coordination algorithm, which consists of a control law for each agent of the network, is either a vector field or a map depending on whether the dynamical model is defined in continuous time or discrete time. From another viewpoint, a coordination algorithm for a group of agents gives rise to a set of coupled dynamical systems.

The first scientific concern regarding a coordination algorithm is the investigation of its correctness. We loosely understand that an algorithm behaves correctly when certain sets that encode the desired behaviors are invariant and attractive for the evolution of the closed-loop network. Typically, the stability and convergence analysis is a complicated matter for various reasons. For example, the couplings between the dynamical systems change as the topology of the mobile network changes. Additionally, we often face discontinuous vector fields modeling the evolution of the network. Finally, we might be faced with the task of analyzing nondeterministic dynamical systems because of design choices when devising the coordination algorithm (at each time instant throughout the evolution, each agent can choose among multiple possible control actions, as opposed to a single one [22]), communication, control, and sensor errors during the execution of the coordination algorithm [25, 28], or as a way of dealing with asynchronism (for example, the asynchronous, deterministic evolution of a mobile network may be subsumed into a larger set of synchronous, nondeterministic evolutions [26]).

A second concern regarding a coordination algorithm is the investigation of its complexity. Loosely speaking, we wish to characterize how quickly a coordination algorithm completes the required task as well as how costly the algorithm is in terms of computation, exchanged messages, and energy consumption. In other words, the performance of a coordination algorithm is quantified by the growth rate of various cost functions for increasing numbers of network agents.

Among the analysis methods used for stability and complexity analysis, we roughly distinguish between linear techniques (ergodic, stochastic [15], and circulant matrices [30] from matrix analysis, graph Laplacians and algebraic connectivity [15, 32] from algebraic graph theory), and nonlinear techniques (symmetries of differential equations [14], invariance principles for differential inclusions and nondeterministic dynamical systems [23], graph grammars [18] from automata theory). For reasons of brevity, we do not include here a comprehensive account of these methods. Instead, two relevant techniques are discussed in the sidebars "Invariance Principle for Nondeterministic Dynamical Systems" and "Tridiagonal Toeplitz and Circulant Matrices."

## Designing Emergent Behaviors

We now discuss four approaches to designing motion coordination algorithms. Although successful examples exist for all four design approaches, there does not exist yet a rigorous system-theoretic approach to general motion-coordination problems. Our intention is therefore to provide a first step toward establishing effective design methods by exploiting the modeling and analysis tools discussed in the previous sections.

Given a network of identical agents equipped with motion control and communication capabilities, the following subsections discuss techniques for analyzing distributed and coordinated motions. The first approach is based on the *design* of *gradient flows*, where a coordination task is specified together with a proximity graph imposing a communication constraint. The second approach is based on the *analysis* of *emergent behaviors*, where a notion of neighboring agents and an interaction law between them is given. The 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 approach relies on the composition of basic behaviors. Although the four approaches have similarities, they are applicable under distinct circumstances and, together, provide a set of useful heuristics. When applicable, the gradient-flow approach is preferable over the others because the convergence analysis is often easier. 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 the gradient-flow approach consists of identifying a global aggregate objective function that is relevant to the desired coordination task, as illustrated in the section entitled "Encoding Coordination Tasks." Once this objective function is determined, the next step is to analyze its smoothness properties and compute its (generalized) gradient. With this information, it is possible to characterize the objective function's critical points, that is, the desired network configurations. The next step is to identify proximity graphs that allow the computation of the gradient of the objective function in a spatially distributed manner. If any of these proximity graphs is spatially distributed (in the sense defined in the section entitled "Spatially distributed maps") over the communication graph of the mobile network, then a control law for each agent consists of following the gradient of the aggregate objective function. By the invariance principle, such a coordination algorithm guarantees convergence of the closed-loop network trajectories to the set of critical points.

*Distortion and area problems*

Coordination algorithms for the distortion and area problems can be based on the gradient-flow approach [22]. Given a convex polygon $Q$ and $R > 0$, the functions $\mathcal{H}_{\mathrm{C}}$ and $\mathcal{H}_{\mathrm{area},R}$ defined by (2) and (3), respectively, are differentiable almost everywhere and satisfy

$$\frac{\partial \mathcal{H}_{\mathrm{C}}}{\partial p_i}(P) = 2\,\mathrm{M}(V_i(P))\,(\mathrm{CM}(V_i(P)) - p_i), \tag{6}$$

$$\frac{\partial \mathcal{H}_{\mathrm{area},R}}{\partial p_i}(P) = \int_{\mathrm{arc}(\partial V_{i,R}(P))} n_{B(p_i,R)}\,\phi, \tag{7}$$

where $n_{B(p,R)}(q)$ is the outward unit normal to $B(p,R)$ at a boundary point $q \in \partial B(p,R)$ and, for each $i \in \{1,\dots,n\}$, $\mathrm{arc}(\partial V_{i,R}(P))$ is the union of all arcs in $\partial V_{i,R}(P)$. In (7), one can also substitute $\mathrm{arc}(\partial V_{i,R}(P))$ by $V_{i,R}(P) \cap \partial B(p_i,R)$. The symbols $\mathrm{M}(W)$ and $\mathrm{CM}(W)$ denote, respectively, the mass and the center of mass with respect to $\phi$ 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* [44]. 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 of $V_{i,R}(P) = V_i(P) \cap B(p_i,R)$ at fixed $V_i(P)$. These configurations are *area-centered Voronoi configurations.*

From (6) it is clear that the gradient of $\mathcal{H}_{\mathrm{C}}$ is spatially distributed over $\mathcal{G}_{\mathrm{D}}$, whereas from (7) 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 9 and 10 illustrate the execution of these algorithms. Figures 11 and 12 illustrate the adaptive properties of these algorithms with respect to agent arrivals and departures.

*Consensus*

Another example of the gradient-flow approach is the asymptotic agreement algorithm in [32]. For a fixed undirected graph $G = (\{1, \dots, n\}, E)$, the function $\Phi_G$ in (4) is smooth, and its partial derivative takes the form

$$\frac{\partial \Phi_G}{\partial x} = 2Lx. \tag{8}$$

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

*Cohesiveness*

The gradient-flow approach is also used for coordination algorithms that achieve cohesiveness [16, 19, 20]. For the complete graph $\mathcal{G}_{\mathrm{complete}}$, the function $\mathcal{H}_{\mathrm{cohe},\mathcal{G}_{\mathrm{complete}}}$, given in (5), is smooth almost everywhere and satisfies

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

The second equality uses the fact that $(dh/dR)(R)$ vanishes for $R \geq R_1$. This gradient is spatially distributed over $\mathcal{G}_{\mathrm{disk}}(R_1)$. Using the negative gradient of this cost function for the control law guarantees that the network of agents asymptotically approaches the set of critical points of $\mathcal{H}_{\mathrm{cohe},\mathcal{G}_{\mathrm{complete}}}$.

The aggregate objective function does not always 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 cannot always be computed. Characterizing all functions whose gradient is spatially distributed with respect to a given proximity graph is an open mathematical challenge. If the gradient cannot be computed with the given information flow, 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 the coordination algorithm that makes each agent follow the gradient of the approximation. This approach is followed in [22].

## Analyzing the coordinated behavior emerging from basic interaction laws

The emergent-behavior approach, typically inspired by heuristics, consists of devising for each network agent a simple control law that can perform the desired task. The resulting coordination algorithm must be spatially distributed with regard to an appropriate proximity graph, and its correctness must be established. Convergence can be characterized by finding an aggregate objective function that encodes the desired coordination task and showing that this function is optimized along the execution of the coordination algorithm.

*Move-away-from-closest-neighbor*

Consider the coordination algorithm where each agent moves away from its closest neighbor [23] (see Figure 13). This simple interaction law is spatially distributed over $\mathcal{G}_{\mathrm{D}}$. It can be proved that, along the evolution of the network, the aggregate cost function

$$\mathcal{H}_{\mathrm{SP}}(P) = \min_{i \neq j \in \{1,\dots,n\}} \left\{ \tfrac{1}{2}\|p_i - p_j\|, \mathrm{dist}(p_i, \partial Q) \right\}, \tag{9}$$

is nondecreasing. The function $\mathcal{H}_{\mathrm{SP}}$ corresponds to the *noninterference problem*, where the network tries to maximize coverage of the domain in such a way that the communication radii 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 for which each agent is at the incenter of its own Voronoi region, where recall that the incenter set of a polygon is the set of centers of the maximum-radius spheres contained in the polygon.

*Flocking*

Flocking consists of reaching consensus on the direction of motion by the agents in the network. Let $\mathcal{G}$ be a proximity graph. Now, consider the coordination algorithm [47] in which each agent performs the following steps: (i) detects its neighbors' headings; (ii) computes the average of its neighbors' headings 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, it can be shown [15] that the agents asymptotically acquire the same heading.

## Designing the coordination algorithm from local objective functions

The local-objective approach has common elements with the two approaches discussed above. 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 to the global aggregate objective function encoding the desired coordination task. Next, we devise a control law to optimize the local function. The specific control strategy might be heuristically derived or might arise naturally from the gradient information of the local objective function. Once the coordination algorithm is set up, we can determine whether the algorithm is spatially distributed and characterize its asymptotic convergence properties.

*Noninterference problem*

Consider the aggregate objective function $\mathcal{H}_{\mathrm{SP}}$ defined in (9). 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, $\mathrm{sm}_W(p) = \mathrm{dist}(p, \partial W)$. Note that both $\mathcal{H}_{\mathrm{SP}}$ and $\mathrm{sm}_W$ are Lipschitz, but not differentiable. Now, for $i \in \{1,\dots,n\}$, consider $\mathrm{sm}_{V_i(P)}$ as a local objective function. The control law for agent $p_i$ is determined as follows: we fix the Voronoi cell $V_i(P)$, compute the generalized gradient of the function $\mathrm{sm}_{V_i(P)}$ (which now depends only on $p_i$ because $V_i(P)$ is held fixed), and implement the resulting gradient ascent as the direction of motion. The resulting dynamical system is generally discontinuous, and can be studied by means of nonsmooth stability analysis [48, 49, 50]. It can be shown [23] that this interaction law corresponds precisely to the strategy "move-away-from-closest-neighbor" discussed earlier (see the 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 $\mathcal{H}_{\mathrm{SP}}$ nondecreasing and to possess analogous asymptotic convergence properties.

*Worst-case problem*

Consider the aggregate objective function

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

where $\mathrm{lg}_W(p)$ is the maximum distance from $p$ to the boundary of the convex polygon $W$, that is, $\mathrm{lg}_W(p) = \max_{q \in W} \|q - p_i\|$. Note that both $\mathcal{H}_{\mathrm{DC}}$ and $\mathrm{lg}_W$ are Lipschitz, but not differentiable. Now, for $i \in \{1, \ldots, n\}$, consider $\mathrm{lg}_{V_i(P)}$ as a local objective function. The control law for agent $p_i$ is determined as follows: we fix the Voronoi cell $V_i(P)$, compute the generalized gradient of the function $\mathrm{lg}_{V_i(P)}$ (which now depends only on $p_i$ because $V_i(P)$ is held fixed), and implement the resulting gradient descent as the direction of motion. It can be shows [23] 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, where the circumcenter of a polygon is the center of the minimum-radius sphere that contains it. Both strategies can be shown to make $\mathcal{H}_{\mathrm{DC}}$ nonincreasing and possess 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 14).

*Rendezvous*

Let $\mathcal{G}$ be a proximity graph spatially distributed over, and with the same connected components as, the $r$-disk graph. Consider the circumcenter algorithm over $\mathcal{G}$, where each agent $i \in \{1, \dots, n\}$ performs the following steps: (i) detects its neighbors $\mathcal{N}_{\mathcal{G},p_i}(P)$ according to $\mathcal{G}$; (ii) computes the circumcenter $\mathrm{CC}(\mathcal{M}_i)$ of the point set $\mathcal{M}_i = \{p_i\} \cup \mathcal{N}_{\mathcal{G},p_i}(P)$ 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 restricted as described in the following paragraph.

At any given time, if two neighboring agents $p_i$, $p_j$ are restricted to move in the closed ball $B\big(\frac{p_i+p_j}{2}, \frac{r}{2}\big)$, then they remain neighbors in the $r$-disk graph. We are therefore interested in computing the point in the segment $[p_i, \mathrm{CC}(\mathcal{M}_i)]$ that is closest to $\mathrm{CC}(\mathcal{M}_i)$ and, at the same time, belongs to the constraint set $C_{p,r}(\mathcal{N}_{\mathcal{G}_{\mathrm{disk}(r)},i}(P))$ defined as the intersection of all $B\big(\frac{p_i+p_k}{2}, \frac{r}{2}\big)$ for $p_k \in \mathcal{N}_{\mathcal{G}_{\mathrm{disk}(r)},i}(P)$. To do this, given $q_0$ and $q_1$ in $\mathbb{R}^d$, and a convex closed set $Q \subset \mathbb{R}^d$ with $q_0 \in Q$, consider the "from to inside" function defined by

$$\mathrm{fti}(q_0, q_1, Q) = \begin{cases} q_1, & \text{if } q_1 \in Q, \\ [q_0, q_1] \cap \partial Q, & \text{if } q_1 \notin Q. \end{cases}$$

The circumcenter algorithm is then the following: during the time interval $[t, t+1]$, agent $i$ moves from $p_i(t)$ to

$$\mathrm{fti}(p_i(t), \mathrm{CC}(\mathcal{M}_i(t)), C_{p_i(t),r}(\mathcal{N}_{\mathcal{G}_{\mathrm{disk}(r)},p_i(t)}(P(t)))),$$

For further details, see [25, 26, 28].

Note that, by moving toward the circumcenter, assuming that all other agents remain fixed, each agent minimizes the local objective function given by the maximum distance from the agent to all of 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 $V_{\text{diam}}$ is nonincreasing along the execution of the circumcenter algorithm. Using the invariance principle for closed algorithms, as described in the sidebar "Invariance Principle for Nondeterministic Dynamical Systems," we can characterize the asymptotic correctness properties of the circumcenter algorithm over $\mathcal{G}$, as illustrated in Figure 15.

## Designing the coordination algorithm by composing different behaviors

This final approach builds on the methods presented above. The idea is to combine different behaviors and examine the resulting coordination algorithm. In particular, behaviors can be combined by implementing one coordination algorithm on most of the network agents and a second coordination algorithm on the remaining 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 particular agents stay fixed or follow a desired path) and implement a feedback law for the others. Examples of this approach include the formation control strategy in [26] to make the network form a straight line, as well as the leader-following algorithm given in [15] 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

In summary, this article has surveyed methods to model spatially distributed problems, encode various coordination tasks through appropriate cost functions, analyze stability and convergence properties, and design motion-coordination schemes. Specific technical tools include proximity graphs, spatially distributed maps, aggregate objective functions, circulant matrices, and invariance principles. These tools play a key role in the various coordination algorithms reported here, as summarized in Table 1. We believe that numerous research issues remain open in the ongoing effort to design algorithms that are efficient, robust, and scalable to large size networks. We expect the coming years to witness an intense development of the field of distributed coordination and of its practical use in applications for multiple vehicles and sensor networks.

## 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.

# 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] 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.

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

[14] 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.

[15] 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.

[16] H. Tanner, A. Jadbabaie, and G. J. Pappas, “Flocking in fixed and switching networks,” *IEEE Transactions on Automatic Control*, 2007, to appear.

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

[18] E. Klavins, R. Ghrist, and D. Lipsky, “A grammatical approach to self-organizing

robotic systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 6, pp. 949–962, 2006.

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

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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, Dec. 2004, pp. 3357–3363.

[25] 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.

[26] 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*, ser. Lecture Notes in Control and Information Sciences, V. Kumar, N. E. Leonard, and A. S. Morse, Eds. New York: Springer Verlag, 2004, vol. 309, pp. 257–282.

[27] 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.

[28] 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. 8, pp. 1289–1298, 2006.

[29] A. M. Bruckstein, N. Cohen, and A. Efrat, "Ants, crickets, and frogs in cyclic pursuit," Center for Intelligent Systems, Technion, Haifa, Israel, Tech. Rep. CIS 9105, July 1991, available electronically at `http://www.cs.technion.ac.il/tech-reports`.

[30] 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.

[31] 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, pp. 297–304.

[32] 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.

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

[34] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control: collective group behavior through local interaction," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, 2007.

[35] 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.

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

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

[38] 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, June 2002, pp. 299–308.

[39] 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.

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

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

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

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

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

[45] D. McGill and W. W. King, *Engineering mechanics, statics and an introduction to dynamics*, 3rd ed. Boston, MA: PWS Publishing Company, 1995.

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

[47] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical Review Letters*, vol. 75, no. 6-7, pp. 1226–1229, 1995.

[48] F. H. Clarke, *Optimization and Nonsmooth Analysis*, ser. Canadian Mathematical Society Series of Monographs and Advanced Texts. John Wiley, 1983.

[49] A. F. Filippov, *Differential Equations with Discontinuous Righthand Sides*, ser. Mathematics and Its Applications. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1988, vol. 18.

[50] A. Bacciotti and F. Ceragioli, "Stability and stabilization of discontinuous systems and nonsmooth Lyapunov functions," *ESAIM. Control, Optimisation & Calculus of Variations*, vol. 4, pp. 361–376, 1999.

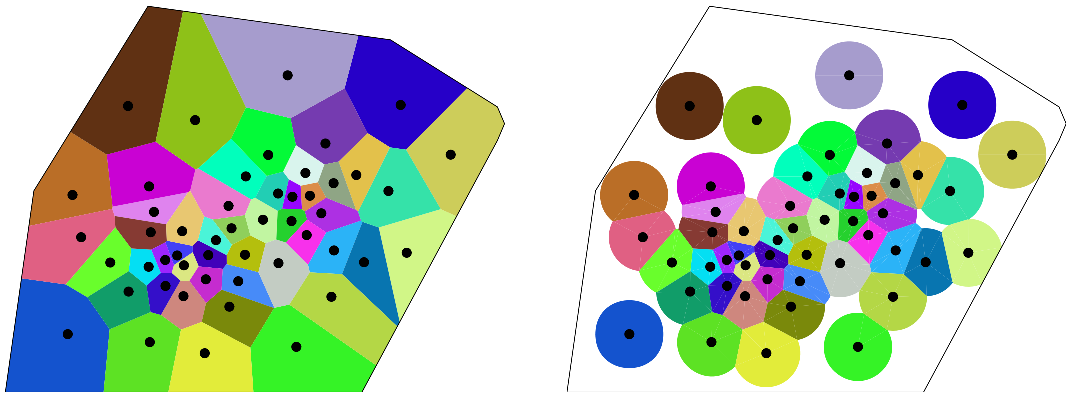
[51] U.S. Fish and Wildlife Service, `http://www.fws.gov`.

**Figure 1. School of fish. Groups of animals can 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 no individual has global knowledge of the group state. Snow geese fly in formation during migration. Photograph reproduced with permission of the Eastern Shore of Virginia National Wildlife Refuge Staff, U.S. Fish and Wildlife Service [51].**

**Figure 3. Herd of wildebeest in the Serengeti National Park, Tanzania. Wildebeest and other animals exhibit complex coordinated behaviors when migrating, such as obstacle avoiding, leader election, and formation keeping. Aerial photograph reprinted from [4] with permission from University of Chicago Press.**

**Figure 4. Territories of male Tilapia mossambica. Some species of fish exhibit territorial behavior by globally partitioning the environment into nonoverlapping 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 known as a Voronoi partition. 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. Photograph reprinted from [6] with permission from Elsevier.**



(a) (b)

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

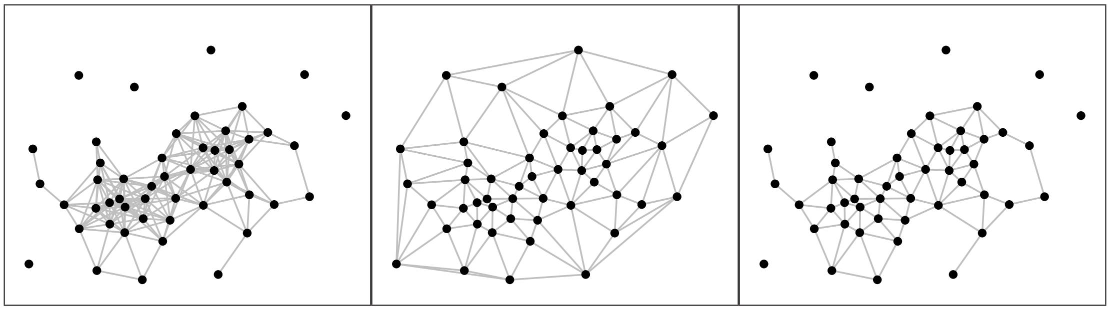
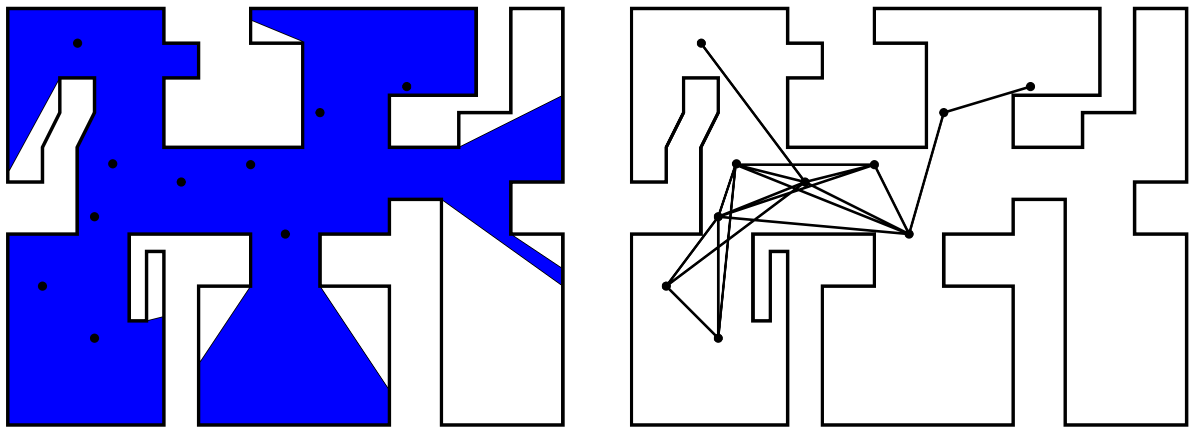


**Figure 6. Proximity graphs in $\mathbb{R}^2$. Proximity graphs provide a natural way to mathematically model the interconnection topology among the agents resulting from their sensing or communication capabilities. From left to right, these plots illustrate the $(2r)$-disk, Delaunay, and $r$-limited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



(a)

(b)

**Figure 7. Visibility-based deployment. A network of agents equipped with omnidirectional cameras can see the blue-colored region of the nonconvex environment $Q$ in (a). The underlying visibility graph $\mathcal{G}_{\mathsf{vis},Q}$ 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.**

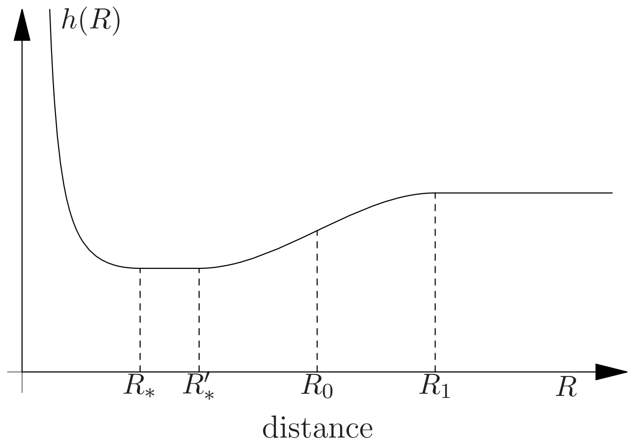
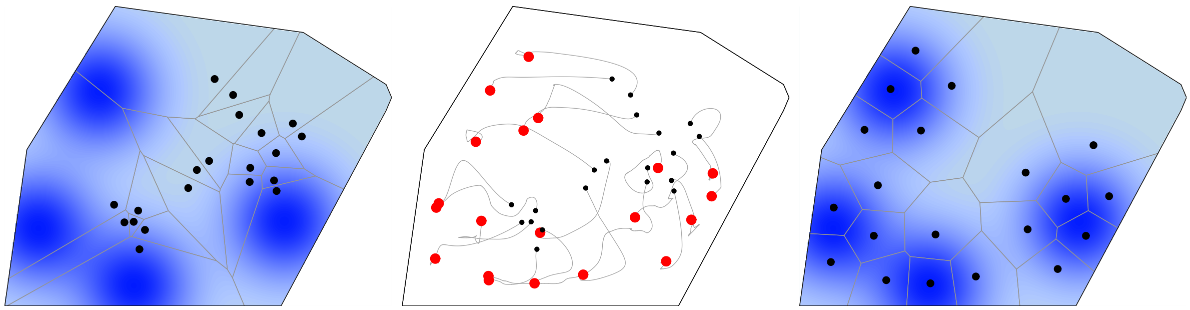
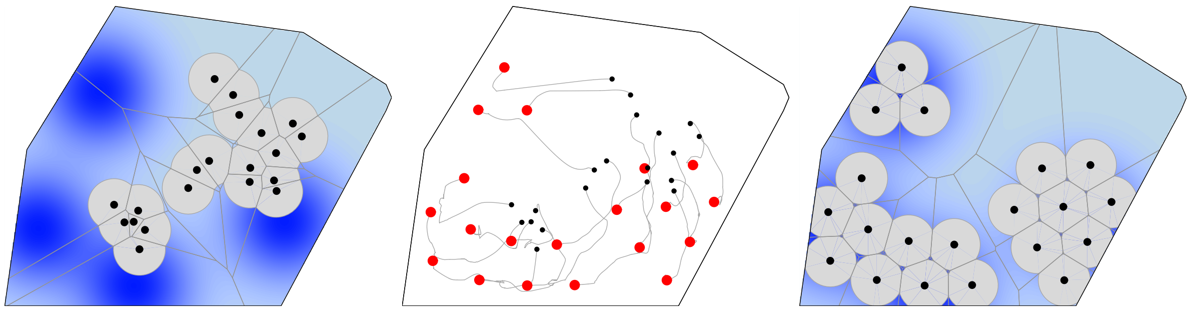



**Figure 8. Sample repulsion/attraction function. Repulsion/attraction functions are used to define aggregate objective functions that encode desirable network configurations in which all neighboring agents are located within a distance in the interval $[R_*, R'_*]$. These functions play a key role in cohesiveness problems for mobile networks.**



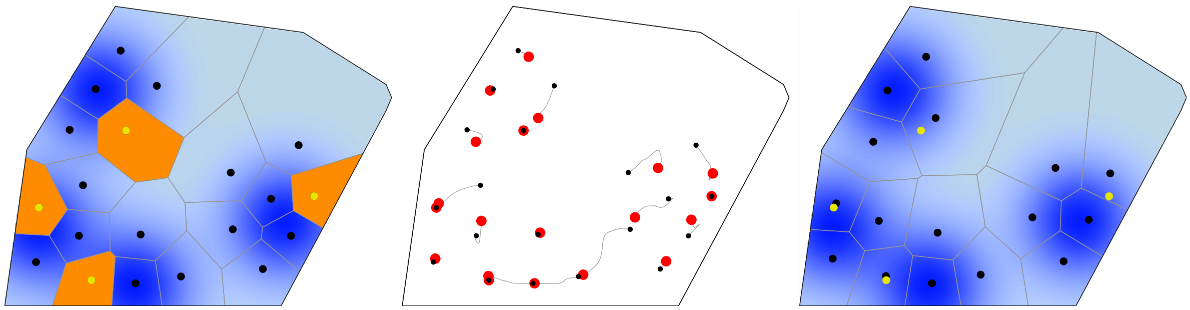
(a) (b) (c)

**Figure 9. Distortion problem. Each of the $20$ mobile agents moves toward the centroid of its Voronoi cell. This strategy corresponds to the network following the gradient $(6)$ 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 $\phi$ 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.**



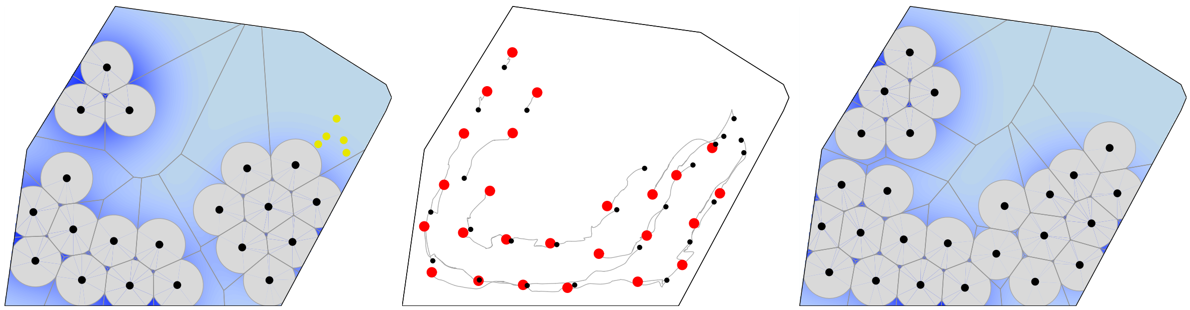
(a) (b) (c)

**Figure 10. Area problem. Each of the $20$ mobile agents follows the gradient $(7)$ of the area function $\mathcal{H}_{\text{area},r}$. The density function $\phi$, which specifies areas of greater importance, and the environment are the same as in Figure 9. (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 $2r$. For each agent $i$, the $r$-limited Voronoi cell $V_{i,r}(P)$ is plotted in light gray.**



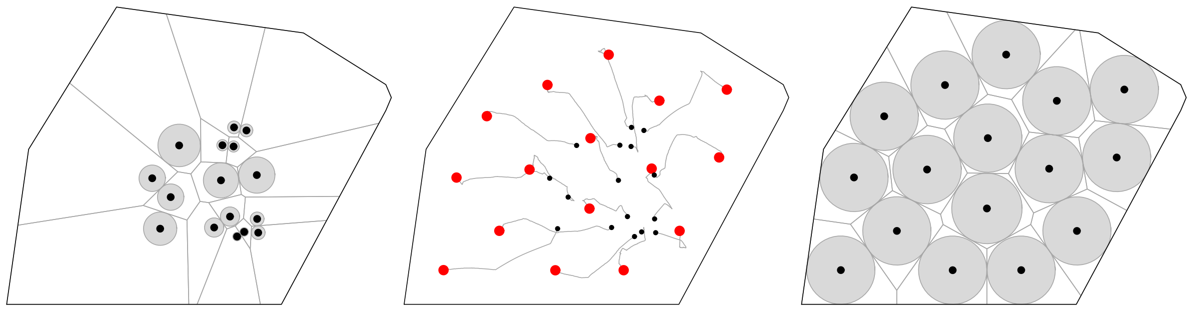
(a) (b) (c)

**Figure 11. Adaptive network behavior under agent failures in the distortion problem. After the final configuration in Figure 9 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. (a) illustrates the location of the agents when the failures occur, while (c) shows the final location of the remaining agents. (b) illustrates the gradient descent flow since the failure occurred.**



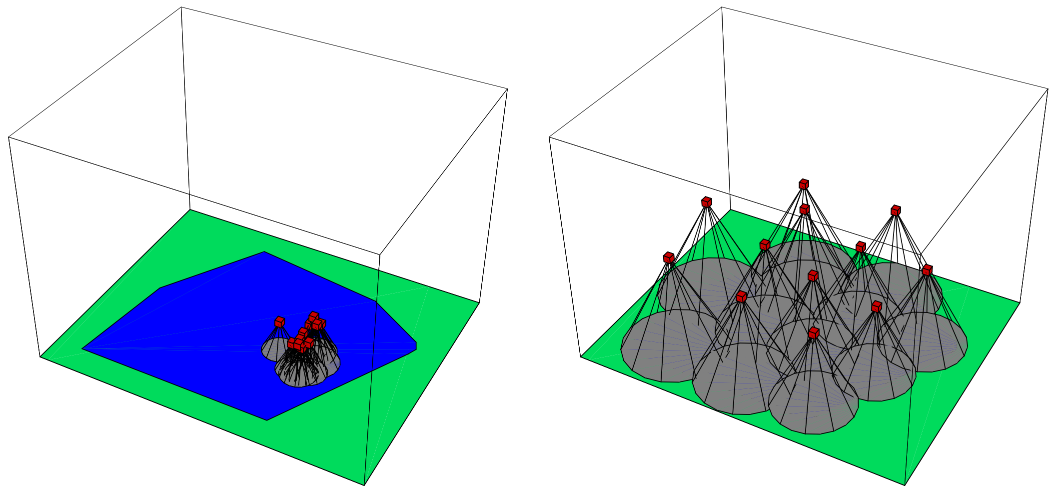
(a) (b) (c)

**Figure 12. Adaptive network behavior under agent arrivals in the area problem. After the final configuration in Figure 10 is reached, five new agents (colored in yellow) enter the environment. The rest of the network adapts to the new situation. (a) illustrates the location of the agents when the arrival of the new agents occurs, while (c) shows the final location of the network. (b) illustrates the gradient descent flow after this event.**



(a) (b) (c)

**Figure 13. Noninterference problem. Each 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 communication radii 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 in (a) and (c).**



(a) (b)

**Figure 14. Worst-case scenario. The network tries to maximize the coverage (illumination) of a convex polygon. Each 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 13). 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.**

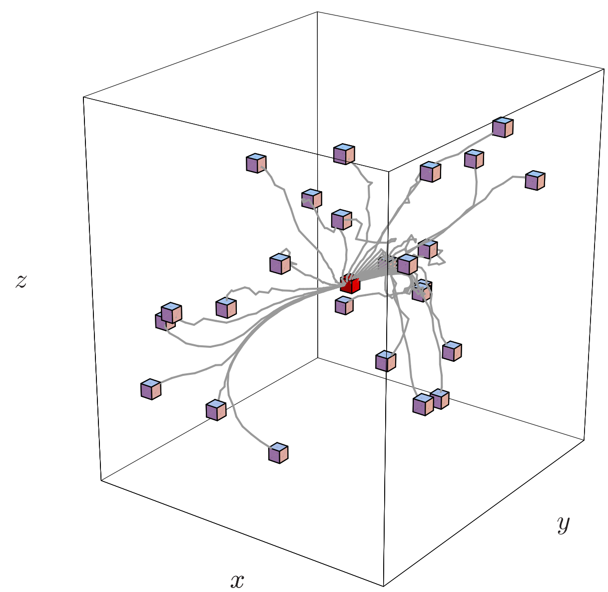



**Figure 15. Circumcenter algorithm in Euclidean space. Each of the $25$ mobile agents moves toward the circumcenter of the point set comprised of its neighbors and of itself. The resulting network behavior asymptotically achieves rendezvous at a point. Indeed, the invariance principle allows us to establish the algorithm's correctness under fairly general conditions. In the execution plotted in this figure, at each time step, each agent randomly selects $\mathcal{G}_{\mathsf{disk}}(2r)$ or $\mathcal{G}_{\mathsf{LD}}(r)$ to compute its set of neighbors.**

| **Agent motion direction** | **Formal algorithm description** | **Distributed information** | **Lyapunov function** | **Asymptotic behavior** | **Ref.** |
|---|---|---|---|---|---|
| centroid of Voronoi cell | $\dot{p}_i = \mathrm{CM}(V_i(P)) - p_i$ | Voronoi neighbors | $\mathcal{H}_{\mathrm{C}}$ | centroidal Voronoi configurations | [21] |
| weighted average normal of $r$-limited Voronoi cell | $\dot{p}_i = \int_{\mathrm{arc}(\partial V_{i,r}(P))} n_{B(p_i,r)}\, \phi$ | $(2r)$-disk neighbors | $\mathcal{H}_{\mathrm{area},r}$ | area-centered Voronoi configurations | [22] |
| average of neighbors | $\dot{p}_i = \sum_{j \in \mathcal{N}_G(i)} (p_j - p_i)$ | neighbors in fixed $G$ | $\Phi_G$ | Consensus | [32] |
| 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 | [23] |
| 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 | [23] |
| circumcenter of neighbors' and own position | $p_i(t+1) = \mathrm{fti}(p_i, \mathrm{CC}(\mathcal{M}_i), C_{p_i,r}(\mathcal{N}_{\mathcal{G}_{\mathrm{disk}}(r),p_i}(P)))$ | $r$-disk neighbors | $V_{\mathrm{diam}}$ | rendezvous | [25] |

**Table 1. Summary of motion-coordination algorithms. The tools presented throughout the article play a key role in the design and analysis of the network behavior resulting from these coordination algorithms. Here, $\partial f$ is the generalized gradient of a locally Lipschitz function $f$, and $\mathrm{Ln}(S)$ is the minimal-norm element of the set $S$. The remaining notation is defined throughout the article.**

**Sidebar: Georgy Fedoseevich Voronoy and His Diagrams**

As reported by the Mathematics Geneaology Project and by Wikipedia, Georgy Fedoseevich Voronoy (1868-1908) studied at the University of St. Petersburg under the supervision of Andrei Andreyevich Markov. After earlier work by Gauss and Dirichlet, he studied the now widely adopted Voronoi diagrams in his work on quadratic forms [S1]. The detailed survey [S2] discusses history, properties, and applications of Voronoi diagrams.

## References


[S1] G. F. Voronoi, "Nouveles applications des paramétres continus á la théorie de formas quadratiques," *Journal für die Reine und Angewandte Mathematik*, vol. 134, pp. 198–287, 1908.

[S2] F. Aurenhammer, "Voronoi diagrams: A survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, 1991.

**Sidebar: Invariance Principle for Nondeterministic Dynamical Systems**

Here we briefly survey a recently developed invariance principle for nondeterministic discrete-time dynamical systems. This principle can serve to establish correctness for distributed systems with switching topologies. Let $T$ be a *set-valued map* on $\mathbb{R}^n$, that is, a map that associate to every point in $\mathbb{R}^n$ a nonempty set in $\mathbb{R}^n$. Let $\mathbb{Z}_{\geq 0}$ be the set of nonnegative integers. A *trajectory* of $T$ is a map $p : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ with the property

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

In other words, given any initial $p_0 \in \mathbb{R}^n$, a trajectory of $T$ is computed by recursively choosing $p(\ell + 1)$ to be an arbitrary element of $T(p(\ell))$. Therefore, $T$ induces a nondeterministic discrete-time dynamical system [S3]. To study the stability of this type of discrete-time dynamical systems, we need to introduce the following notions. According to [S3], $T$ 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 $T : \mathbb{R}^n \to \mathbb{R}^n$ that is continuous at $p \in \mathbb{R}^n$ is closed at $p$. A set $C$ is *weakly positively invariant with respect to* $T$ if, for every initial condition $p_0 \in C$, there exists at least one 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 *nonincreasing 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 [22].

**Theorem 1** *Let $T$ be a set-valued map on $\mathbb{R}^n$ and $W \subset \mathbb{R}^n$. Assume that, for all $p \in W \subset \mathbb{R}^n$, $T$ is closed at $p$, and let $V : \mathbb{R}^n \to \mathbb{R}$ be a continuous function nonincreasing along*

*$T$ on $W$. Assume that the trajectory $p : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ of the set-valued map $T$ remains 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 $\{p \in \overline{W} \mid \text{there exists } p' \in T(p) \text{ with } V(p') = V(p)\}$, where $\overline{W}$ denotes the closure of $W$.*

## References

[S3] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. Reading, MA: Addison-Wesley, 1984.

**Sidebar: Tridiagonal Toeplitz and Circulant Matrices**

Here, we discuss convergence rates for linear dynamical systems defined by Toeplitz and circulant matrices [S4, S5]. For $n \geq 2$ and $a, b, c \in \mathbb{R}$, we define the $n \times n$ matrices

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

We refer to $\mathrm{Trid}_n$ and $\mathrm{Circ}_n$ 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 [S6] and in cyclic pursuit [29, 30]. In Figure S1, we illustrate two algorithms in which the control action of each agent depends on the location of the agent's clockwise and counterclockwise neighbors.

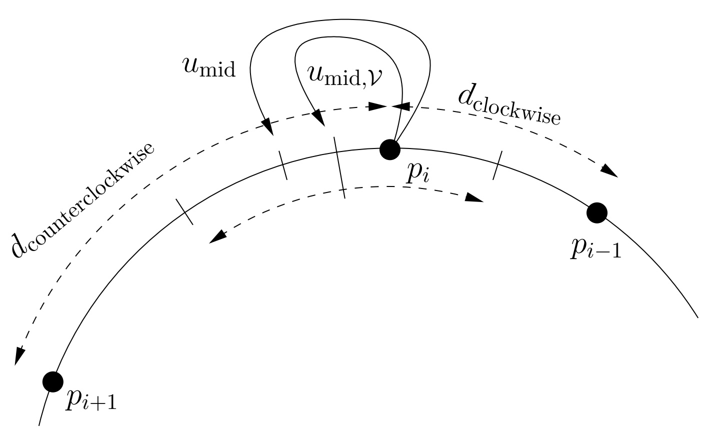



**Figure S1. Clockwise and counterclockwise neighbors of an agent in a network of robots moving along a circular domain. Control laws such as "go toward the midpoint $u_{\mathrm{mid}}$ of the locations of the clockwise and counterclockwise neighbors",**

**or "go toward the midpoint $u_{\mathbf{mid},\mathcal{V}}$ of the Voronoi segment of the agent" give rise to linear dynamical systems described by circulant matrices. In the closed-loop system determined by $u_{\mathbf{mid},\mathcal{V}}$, the agents achieve a uniform distribution along the circular domain. In contrast, oscillations persist when the law $u_{\mathbf{mid}}$ is adopted.**

A salient feature of these matrices is that their eigenvalues and their dependence on $n$ can be explicitly computed [S6]. 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 case $a = c \neq 0$ and $|b| + 2|a| = 1$, $x$ converges exponentially fast to 0, while the maximum time required for $\|x(\ell)\|_2 \leq \varepsilon \|x_0\|_2$ is of order $n^2 \log \varepsilon^{-1}$ for small $\varepsilon$. 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 case $a \geq 0$, $c \geq 0$, $b > 0$, and $a + b + c = 1$, $y$ converges exponentially fast to $y_{\mathrm{ave}}\mathbf{1}$, where $y_{\mathrm{ave}} = \frac{1}{n}\mathbf{1}^T y_0$, while 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 $\mathbf{1} = (1, \ldots, 1)^T$.

## References

[S4] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. Philadelphia, PA: SIAM, 2001.

[S5] P. J. Davis, *Circulant Matrices*, 2nd ed. Providence, RI: American Mathematical Society, 1994.

[S6] S. Martínez, F. Bullo, J. Cortés, and E. Frazzoli, "On synchronous robotic networks – Part I: Models, tasks, and complexity . & Part II: Time complexity of rendezvous and deployment algorithms," *IEEE Transactions on Automatic Control*, Apr. 2005, submitted.

**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 in the Department of Mechanical and Aerospace Engineering, University of California, San Diego. She received the Best Student Paper award at the 2002 IEEE Conference on Decision and Control for 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, Spain, in 1997 and his Ph.D. degree in engineering mathematics from the Universidad Carlos III de Madrid, Spain, in 2001. Since 2004, he has been an assistant professor in the Department of Applied Mathematics and Statistics, University of California, Santa Cruz. He previously held postdoctoral positions at the Systems, Signals, and Control Department of the University of Twente, and at the Coordinated Science Laboratory of the University of Illinois at Urbana-Champaign. He is the author of *Geometric,* *Control and Numerical Aspects of Nonholonomic Systems* (Springer Verlag, 2002). He is currently an associate editor for the *European Journal of Control*.

**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 in the Department of Mechanical Engineering, University of California, Santa Barbara. He is a coauthor (with Andrew D. Lewis) of *Geometric Control of Mechanical Systems* (Springer Verlag, 2004). He is currently an associate editor for the *IEEE Transactions on Automatic Control* and for *SIAM Journal of Control and Optimization*.