# Coverage control for mobile sensing networks

Jorge Cortés, *Member IEEE*, Sonia Martínez, *Member IEEE*, Timur Karatas, Francesco Bullo, *Member IEEE*

***Abstract*—This paper presents control and coordination algorithms for groups of vehicles. The focus is on autonomous vehicle networks performing distributed sensing tasks where each vehicle plays the role of a mobile tunable sensor. The paper proposes gradient descent algorithms for a class of utility functions which encode optimal coverage and sensing policies. The resulting closed-loop behavior is adaptive, distributed, asynchronous, and verifiably correct.**

***Index Terms*—Coverage control, distributed and asynchronous algorithms, sensor networks, centroidal Voronoi partitions**

## I. Introduction

### *Mobile sensing networks*

The deployment of large groups of autonomous vehicles is rapidly becoming possible because of technological advances in networking and in miniaturization of electro-mechanical systems. In the near future, large numbers of robots will coordinate their actions through ad-hoc communication networks and will perform challenging tasks including search and recovery operations, manipulation in hazardous environments, exploration, surveillance, and environmental monitoring for pollution detection and estimation. The potential advantages of employing teams of agents are numerous. For instance, certain tasks are difficult, if not impossible, when performed by a single vehicle agent. Further, a group of vehicles inherently provides robustness to failures of single agents or communication links.

Working prototypes of active sensing networks have already been developed; see [1], [2], [3], [4]. In [3], launchable miniature mobile robots communicate through a wireless network. The vehicles are equipped with sensors for vibrations, acoustic, magnetic, and IR signals as well as an active video module (i.e., the camera or micro-radar is controlled via a pan-tilt unit). A second system is suggested in [4] under the name of Autonomous Oceanographic Sampling Network. In this case, underwater vehicles are envisioned measuring temperature, currents, and other distributed oceanographic signals. The vehicles communicate via an acoustic local area network and coordinate their motion in response to local sensing information and to evolving global data. This mobile sensing network is meant to provide the ability to sample the environment adaptively in space and time. By identifying evolving temperature and current gradients with higher accuracy and resolution than current static sensors, this technology could lead to the development and validation of improved oceanographic models.

### *Optimal sensor allocation and coverage problems*

A fundamental prototype problem in this paper is that of characterizing and optimizing notions of quality-of-service provided by an adaptive sensor network in a dynamic environment. To this goal, we introduce a notion of *sensor coverage* that formalizes an optimal sensor placement problem. This spatial resource allocation problem is the subject of a discipline called locational optimization [5], [6], [7], [8], [9].

Locational optimization problems pervade a broad spectrum of scientific disciplines. Biologists rely on locational optimization tools to study how animals share territory and to characterize the behavior of animal groups obeying the following interaction rule: each animal establishes a region of dominance and moves toward its center. Locational optimization problems are spatial resource allocation problems (e.g., where to place mailboxes in a city or cache servers on the internet) and play a central role in quantization and information theory (e.g., how to design a minimum-distortion fixed-rate vector quantizer). Other technologies affected by locational optimization include mesh and grid optimization methods, clustering analysis, data compression, and statistical pattern recognition.

Because locational optimization problems are so widely studied, it is not surprising that methods are indeed available to tackle coverage problems; see [5], [8], [10], [9]. However, most currently-available algorithms are not applicable to mobile sensing networks because they inherently assume a centralized computation for a limited size problem in a known static environment. This is not the case in multi-vehicle networks which, instead, rely on a distributed communication and computation architecture. Although an ad-hoc wireless network provides the ability to share some information, no global omniscient leader might be present to coordinate the group. The inherent spatially-distributed nature and limited communication capabilities of a mobile network invalidate classic approaches to algorithm design.

### *Distributed asynchronous algorithms for coverage control*

In this paper we design coordination algorithms implementable by a multi-vehicle network with limited sensing and communication capabilities. Our approach is related to the classic Lloyd algorithm from quantization theory; see [11]

Submitted on November 4, 2002, revised on June 16, 2003. Appeared in the IEEE Transactions on Robotics and Automation, vol 20, issue 2, year 2004, pages 243–255. This version contains minor corrections and was generated on April 24, 2008. Previous short versions of this paper appeared in the IEEE Conference on Robotics and Automation, Arlington, VA, May 2002, and Mediterranean Conference on Control and Automation, Lisbon, Portugal, July 2002.

Jorge Cortés, Timur Karatas and Francesco Bullo are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1308 W. Main St., Urbana, IL 61801, United States, Tels: +1-217-244-8734, +1-217-244-9414 and +1-217-333-0656, Fax: +1-217-244-1653, Email: {jcortes,tkaratas,bullo}@uiuc.edu

Sonia Martínez is with the Escola Universitària Politècnica de Vilanova i la Geltrú, Universidad Politécnica de Cataluña, Av. V. Balaguer s/n, Vilanova i la Geltrú, 08800, Spain, Tel: +34-938967743, Fax: +34-938967700, Email: soniam@mat.upc.es

for a reprint of the original report and [12] for a historical overview. We present Lloyd descent algorithms that take into careful consideration all constraints on the mobile sensing network. In particular, we design coverage algorithms that are adaptive, distributed, asynchronous, and verifiably asymptotically correct:

**Adaptive:** Our coverage algorithms provide the network with the ability to address changing environments, sensing task, and network topology (due to agents departures, arrivals, or failures).

**Distributed:** Our coverage algorithms are distributed in the sense that the behavior of each vehicle depends only on the location of its neighbors. Also, our algorithms do not require a fixed-topology communication graph, i.e., the neighborhood relationships do change as the network evolves. The advantages of distributed algorithms are scalability and robustness.

**Asynchronous:** Our coverage algorithms are amenable to asynchronous implementation. This means that the algorithms can be implemented in a network composed of agents evolving at different speeds, with different computation and communication capabilities. Furthermore, our algorithms do not require a global synchronization and convergence properties are preserved even if information about neighboring vehicles propagates with some delay. An advantage of asynchronism is a minimized communication overhead.

**Verifiable Asymptotically Correct:** Our algorithms guarantee monotonic descent of the cost function encoding the sensing task. Asymptotically, the evolution of the mobile sensing network is guaranteed to converge to so-called centroidal Voronoi configurations (i.e., configurations where the location of each generator coincides with the centroid of the corresponding Voronoi cell) that are critical points of the optimal sensor coverage problem.

Let us describe in some detail what are the contributions of this paper. Section II reviews certain locational optimization problems and their solutions as centroidal Voronoi partitions. Section III provides a continuous-time version of the classic Lloyd algorithm from vector quantization and applies it to the setting of multi-vehicle networks. In discrete-time, we propose a family of Lloyd algorithms. We carefully characterize convergence properties for both continuous and discrete-time versions (Appendix A collects some relevant facts on descent flows). We discuss a worst-case optimization problem, we investigate a simple uniform planar setting, and we present simulation results.

Section IV presents two asynchronous distributed implementations of Lloyd algorithm for ad-hoc networks with communication and sensing capabilities. Our treatment carefully accounts for the constraints imposed by the distributed nature of the vehicle network. We present two asynchronous implementations, one based on classic results on distributed gradient flows, the other based on the structure of the coverage problem. (Appendix B briefly reviews some known results on asynchronous gradient algorithms.)

Section V-A considers vehicle models with more realistic dynamics. We present two formal results on passive vehicle dynamics and on vehicles equipped with individual local controllers. We present numerical simulations of passive vehicle models and of unicycle mobile vehicles. Next, Section V-B describes density functions that lead the multi-vehicle network to predetermined geometric patterns. We present our conclusions and directions for future research in Section VI.

### *Review of distributed algorithms for cooperative control*

Recent years have witnessed a large research effort focused on motion planning and coordination problems for multi-vehicle systems. Issues include geometric patterns [13], [14], [15], [16], formation control [17], [18], gradient climbing [19], and conflict avoidance [20]. It is only recently, however, that truly distributed coordination laws for dynamic networks are being proposed; e.g., see [21], [22], [23].

Heuristic approaches to the design of interaction rules and emerging behaviors have been throughly investigated within the literature on behavior-based robotics; see [24], [25], [17], [26], [27], [28]. An example of coverage control is discussed in [29]. Along this line of research, algorithms have been designed for sophisticated cooperative tasks. However, no formal results are currently available on how to design reactive control laws, ensure their correctness, and guarantee their optimality with respect to an aggregate objective.

The study of distributed algorithms is concerned with providing mathematical models, devising precise specifications for their behavior, and formally proving their correctness and complexity. Via an automata-theoretic approach, the references [30], [31] treat distributed consensus, resource allocation, communication, and data consistency problems. From a numerical optimization viewpoint, the works in [32], [33] discuss distributed asynchronous algorithms as networking algorithms, rate and flow control, and gradient descent flows. Typically, both these sets of references consider networks with fixed topology, and do not address algorithms over ad-hoc dynamically changing networks. Another common assumption is that any time an agent communicates its location, it broadcasts it to every other agent in the network. In our setting, this would require a non-distributed communication set-up.

Finally, we note that the terminology "coverage" is also used in [34], [35] and references therein to refer to a different problem called the *coverage path planning problem*, where a single robot equipped with a limited footprint sensor needs to visit all points in its environment.

## II. From location optimization to centroidal Voronoi partitions

### *A. Locational optimization*

In this section we describe a collection of known facts about a meaningful optimization problem. References include the theory and applications of centroidal Voronoi partitions, see [10], and the discipline of facility location, see [6]. Along the paper, we interchangeably refer to the elements of the network as sensors, agents, vehicles, or robots. We let $\mathbb{R}_+$ be the set of nonnegative real numbers, $\mathbb{N}$ be the set of positive natural numbers and $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$.

Let $Q$ be a convex polytope in $\mathbb{R}^N$ including its interior, and let $\|\cdot\|$ denote the Euclidean distance function. We call a map $\phi : Q \to \mathbb{R}_+$ a *distribution density function* if it represents a measure of information or probability that some event take place over $Q$. In equivalent words, we can consider $Q$ to be the bounded support of the function $\phi$. Let $P = (p_1, \dots, p_n)$ be the *location of $n$ sensors*, each moving in the space $Q$. Because of noise and loss of resolution, the *sensing performance* at point $q$ taken from $i$th sensor at the position $p_i$ degrades with the distance $\|q - p_i\|$ between $q$ and $p_i$; we describe this degradation with a non-decreasing differentiable function $f : \mathbb{R}_+ \to \mathbb{R}_+$. Accordingly, $f(\|q - p_i\|)$ provides a quantitative assessment of how poor the sensing performance is.

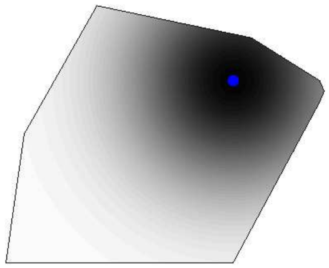

Fig. 1. Contour plot on a polygonal environment of the Gaussian density function $\phi = \exp(-x^2 - y^2)$.

*Remark 2.1:* As an example, consider $n$ mobile robots equipped with microphones attempting to detect, identify, and localize a sound-source. *How should we plan to robots' motion in order to maximize the detection probability?* Assuming the source emits a known signal, the optimal detection algorithm is a matched filter (i.e., convolve the known waveform with the received signal and threshold). The source is detected depending on the signal-to-noise-ratio, which is inversely proportional to the distance between the microphone and the source. Various electromagnetic and sound sensors have signal-to-noise ratios inversely proportional to distance.

Within the context of this paper, a *partition* of $Q$ is a collection of $n$ polytopes $\mathcal{W} = \{W_1, \dots, W_n\}$ with disjoint interiors whose union is $Q$. We say that two partitions $\mathcal{W}$ and $\mathcal{W}'$ are equal if $W_i$ and $W_i'$ only differ by a set of $\phi$-measure zero, for all $i \in \{1, \dots, n\}$.

We consider the task of minimizing the locational optimization function

$$\mathcal{H}(P, \mathcal{W}) = \sum_{i=1}^{n} \int_{W_i} f(\|q - p_i\|)\phi(q)dq, \tag{1}$$

where we assume that the $i$th sensor is responsible for measurements over its "dominance region" $W_i$. Note that the function $\mathcal{H}$ is to be minimized with respect to both (1) the sensors location $P$, and (2) the assignment of the dominance regions $\mathcal{W}$. The optimization is therefore to be performed with respect to the position of the sensors and the partition of the space. This problem is referred to as a facility location problem and in particular as a continuous $p$-median problem in [6].

*Remark 2.2:* Note that if we interchange the positions of any two agents, along with their associated regions of dominance, the value of the locational optimization function $\mathcal{H}$ is not affected. Equivalently, if $\Sigma_n$ denotes the discrete group of permutations of $n$ elements, then $\mathcal{H}(p_1, \dots, p_n, W_1, \dots, W_n) = \mathcal{H}(p_{\sigma(1)}, \dots, p_{\sigma(n)}, W_{\sigma(1)}, \dots, W_{\sigma(n)})$ for all $\sigma \in \Sigma_n$. To eliminate this discrete redundancy, one could take natural action of $\Sigma_n$ on $Q^n$, and consider $Q^n/\Sigma_n$ as the configuration space for the position $P$ of the $n$ vehicles.

## B. Voronoi partitions

One can easily see that, at fixed sensors location, the optimal partition of $Q$ is the *Voronoi partition* $\mathcal{V}(P) = \{V_1, \dots, V_n\}$ generated by the points $(p_1, \dots, p_n)$:

$$V_i = \{q \in Q \mid \|q - p_i\| \le \|q - p_j\|, \ \forall j \neq i\}.$$

We refer to [9] for a comprehensive treatment on Voronoi diagrams, and briefly present some relevant concepts. The set of regions $\{V_1, \dots, V_n\}$ is called the Voronoi diagram for the generators $\{p_1, \dots, p_n\}$. When the two Voronoi regions $V_i$ and $V_j$ are adjacent (i.e., they share an edge), $p_i$ is called a *(Voronoi) neighbor* of $p_j$ (and vice-versa). The set of indexes of the Voronoi neighbors of $p_i$ is denoted by $\mathcal{N}(i)$. Clearly, $j \in \mathcal{N}(i)$ if and only if $i \in \mathcal{N}(j)$. We also define the $(i,j)$-face as $\Delta_{ij} = V_i \cap V_j$. Voronoi diagrams can be defined with respect to various distance functions, e.g., the 1-, 2-, $s$-, and $\infty$-norm over $Q = \mathbb{R}^m$, see [36]. Some useful facts about the Euclidean setting are the following: if $Q$ is a convex polytope in a $N$-dimensional Euclidean space, the boundary of each $V_i$ is the union of $(N-1)$-dimensional convex polytopes.

In what follows, we shall write

$$\mathcal{H}_{\mathcal{V}}(P) = \mathcal{H}(P, \mathcal{V}(P)).$$

Note that using the definition of the Voronoi partition, we have $\min_{i \in \{1,\dots,n\}} f(\|q - p_i\|) = f(\|q - p_j\|)$ for all $q \in V_j$. Therefore,

$$\begin{aligned} \mathcal{H}_{\mathcal{V}}(P) &= \int_Q \min_{i \in \{1,\dots,n\}} f(\|q - p_i\|)\phi(q)dq\,, \\ &= E_{(Q,\phi)}\left[\min_{i \in \{1,\dots,n\}} f(\|q - p_i\|)\right], \end{aligned} \tag{2}$$

that is, the locational optimization function can be interpreted as an expected value composed with a min operation. This is the usual way in which the problem is presented in the facility location and operations research literature [6]. Remarkably, one can show [10] that

$$\frac{\partial \mathcal{H}_{\mathcal{V}}}{\partial p_i}(P) = \frac{\partial \mathcal{H}}{\partial p_i}(P, \mathcal{V}(P)) = \int_{V_i} \frac{\partial}{\partial p_i} f(\|q - p_i\|)\,\phi(q)dq, \tag{3}$$

i.e., the partial derivative of $\mathcal{H}_{\mathcal{V}}$ with respect to the $i$th sensor only depends on its own position and the position of its Voronoi neighbors. Therefore the computation of the derivative of $\mathcal{H}_{\mathcal{V}}$ with respect to the sensors' location is decentralized *in the sense of Voronoi*. Moreover, one can deduce some smoothness properties of $\mathcal{H}_{\mathcal{V}}$: since the Voronoi partition $\mathcal{V}$ depends at least continuously on $P = (p_1, \dots, p_n)$, the function $\mathcal{H}_{\mathcal{V}}$ is at least continuously differentiable.

### C. Centroidal Voronoi partitions

Let us recall some basic quantities associated to a region $V \subset \mathbb{R}^N$ and a mass density function $\rho$. The (generalized) mass, centroid (or center of mass), and polar moment of inertia are defined as

$$M_V = \int_V \rho(q)\,dq, \quad C_V = \frac{1}{M_V}\int_V q\,\rho(q)\,dq,$$
$$J_{V,p} = \int_V \|q-p\|^2\,\rho(q)\,dq.$$

Additionally, by the parallel axis theorem, one can write,

$$J_{V,p} = J_{V,C_V} + M_V\,\|p - C_V\|^2 \tag{4}$$

where $J_{V,C_V} \in \mathbb{R}_+$ is defined as the polar moment of inertia of the region $V$ about its centroid $C_V$.

Let us consider again the locational optimization problem (1), and suppose now we are strictly interested in the setting

$$\mathcal{H}(P,\mathcal{W}) = \sum_{i=1}^{n}\int_{W_i}\|q-p_i\|^2\phi(q)dq, \tag{5}$$

that is, we assume $f(\|q-p_i\|) = \|q-p_i\|^2$. The parallel axis theorem leads to simplifications for both the function $\mathcal{H}_{\mathcal{V}}$ and its partial derivative:

$$\mathcal{H}_{\mathcal{V}}(P) = \sum_{i=1}^{n} J_{V_i,C_{V_i}} + \sum_{i=1}^{n} M_{V_i}\,\|p_i - C_{V_i}\|^2$$
$$\frac{\partial\mathcal{H}_{\mathcal{V}}}{\partial p_i}(P) = 2M_{V_i}(p_i - C_{V_i}).$$

Here the mass density function is $\rho = \phi$. It is convenient to define

$$\mathcal{H}_{\mathcal{V},1} = \sum_{i=1}^{n} J_{V_i,C_{V_i}}\,, \quad \mathcal{H}_{\mathcal{V},2} = \sum_{i=1}^{n} M_{V_i}\|p_i - C_{V_i}\|^2\,.$$

Therefore, the (not necessarily unique) local minimum points for the location optimization function $\mathcal{H}_{\mathcal{V}}$ are *centroids* of their Voronoi cells, i.e., each location $p_i$ satisfies two properties simultaneously: it is the generator for the Voronoi cell $V_i$ and it is its centroid

$$C_{V_i} = \operatorname{argmin}_{p_i}\mathcal{H}_{\mathcal{V}}(P).$$

Accordingly, the critical partitions and points for $\mathcal{H}$ are called *centroidal Voronoi partitions*. We will refer to a sensors' configuration as a *centroidal Voronoi configuration* if it gives rise to a centroidal Voronoi partition. Of course, centroidal Voronoi configurations depend on the specific distribution density function $\phi$, and an arbitrary pair $(Q,\phi)$ admits in general multiple centroidal Voronoi configurations. This discussion provides a proof alternative to the one given in [10] for the necessity of centroidal Voronoi partitions as solutions to the continuous $p$-median location problem.

## III. Continuous and Discrete-Time Lloyd Descent for Coverage Control

In this section, we describe algorithms to compute the location of sensors that minimize the cost $\mathcal{H}$, both in continuous and in discrete-time. In Section III-A, we propose a continuous-time version of the classic Lloyd algorithm. Here, both the positions and partitions evolve in continuous time, whereas Lloyd algorithm for vector quantization is designed in discrete time. In Section III-B, we develop a family of variations of Lloyd algorithm in discrete time. In both settings, we prove that the proposed algorithms are *gradient descent flows*.

### A. A continuous-time Lloyd algorithm

Assume the sensors location obeys a first order dynamical behavior described by

$$\dot{p}_i = u_i.$$

Consider $\mathcal{H}_{\mathcal{V}}$ a cost function to be minimized and impose that the location $p_i$ follows a gradient descent. In equivalent control theoretical terms, consider $\mathcal{H}_{\mathcal{V}}$ a Lyapunov function and stabilize the multi-vehicle system to one of its local minima via dissipative control. Formally, we set

$$u_i = -k_{\text{prop}}(p_i - C_{V_i}), \tag{6}$$

where $k_{\text{prop}}$ is a positive gain, and where we assume that the partition $\mathcal{V}(P) = \{V_1,\dots,V_n\}$ is continuously updated.

*Proposition 3.1 (Continuous-time Lloyd descent):* For the closed-loop system induced by equation (6), the sensors location converges asymptotically to the set of critical points of $\mathcal{H}_{\mathcal{V}}$, i.e., the set of centroidal Voronoi configurations on $Q$. Assuming this set is finite, the sensors location converges to a centroidal Voronoi configuration.

*Proof:* Under the control law (6), we have

$$\frac{d}{dt}\mathcal{H}_{\mathcal{V}}(P(t)) = \sum_{i=1}^{n}\frac{\partial\mathcal{H}_{\mathcal{V}}}{\partial p_i}\,\dot{p}_i$$
$$= -2k_{\text{prop}}\sum_{i=1}^{n} M_{V_i}\|p_i - C_{V_i}\|^2 = -2k_{\text{prop}}\mathcal{H}_{\mathcal{V},2}(P(t)).$$

By LaSalle's principle, the sensors location converges to the largest invariant set contained in $\mathcal{H}_{\mathcal{V},2}^{-1}(0)$, which is precisely the set of centroidal Voronoi configurations. Since this set is clearly invariant for (6), we get the stated result. If $\mathcal{H}_{\mathcal{V},2}^{-1}(0)$ consists of a finite collection of points, then $P(t)$ converges to one of them, see Corollary A.2. ■

*Remark 3.2:* If $\mathcal{H}_{\mathcal{V},2}^{-1}(0)$ is finite, and $P(t) \to C$, then a sufficient condition that guarantees exponential convergence is that the Hessian of $\mathcal{H}_{\mathcal{V}}$ be positive definite at $C$. Establishing this property is a known open problem, see [10]. Note that this gradient descent is not guaranteed to find the global minimum. For example, in the vector quantization and signal processing literature [12], it is known that for bimodal distribution density functions, the solution to the gradient flow reaches local minima where the number of generators allocated to the two region of maxima are not optimally partitioned.

### B. A family of discrete-time Lloyd algorithms

Let us consider the following variations of Lloyd algorithm. Let $T$ be a continuous mapping $T : Q^n \to Q^n$ verifying the following two properties:

(a) for all $i \in \{1, \dots, n\}$, $\|T_i(P) - C_{V_i(P)}\| \le \|p_i - C_{V_i(P)}\|$, where $T_i$ denotes the $i$th component of $T$,
(b) if $P$ is not centroidal, then there exists a $j$ such that $\|T_j(P) - C_{V_j(P)}\| < \|p_j - C_{V_j(P)}\|$.

Property (a) guarantees that, if moved according to $T$, the agents of the network do not increase their distance to its corresponding centroid. Property (b) ensures that at least one robot moves at each iteration and strictly approaches the centroid of its Voronoi region. Because of this property, the fixed points of $T$ are the set of centroidal Voronoi configurations.

*Proposition 3.3 (Discrete-time Lloyd descent):* Let $T : Q^n \to Q^n$ be a continuous mapping satisfying properties (a) and (b). Let $P_0 \in Q^n$ denote the initial sensors' location. Then, the sequence $\{T^m(P_0) \mid m \in \mathbb{N}\}$ converges to the set of centroidal Voronoi configurations. If this set is finite, then the sequence $\{T^m(P_0) \mid m \in \mathbb{N}\}$ converges to a centroidal Voronoi configuration.

*Proof:* Consider $\mathcal{H}_{\mathcal{V}} : Q^n \to \mathbb{R}_+$ as an objective function for the algorithm $T$. Using the parallel axis theorem, $\mathcal{H}(P, \mathcal{W}) = \sum_{i=1}^n J_{W_i, C_{W_i}} + \sum_{i=1}^n M_{W_i} \|p_i - C_{W_i}\|^2$, and therefore

$$\mathcal{H}(P', \mathcal{W}) \le \mathcal{H}(P, \mathcal{W}) , \qquad (7)$$

as long as $\|p'_i - C_{W_i}\| \le \|p_i - C_{W_i}\|$ for all $i \in \{1, \dots, n\}$, with strict inequality if for any $i$, $\|p'_i - C_{W_i}\| < \|p_i - C_{W_i}\|$. In particular, $\mathcal{H}(C_{\mathcal{W}}, \mathcal{W}) \le \mathcal{H}(P, \mathcal{W})$, with strict inequality if $P \ne C_{\mathcal{W}}$, where $C_{\mathcal{W}}$ denotes the set of centroids of the partition $\mathcal{W}$. Moreover, since the Voronoi partition is the optimal one for fixed $P$, we also have

$$\mathcal{H}(P, \mathcal{V}(P)) \le \mathcal{H}(P, \mathcal{W}) , \qquad (8)$$

with strict inequality if $\mathcal{W} \ne \mathcal{V}(P)$.

Now, because of property (a) of $T$, inequality (7) yields

$$\mathcal{H}(T(P), \mathcal{V}(P)) \le \mathcal{H}(P, \mathcal{V}(P)) = \mathcal{H}_{\mathcal{V}}(P) ,$$

and the inequality is strict if $P$ is not centroidal by property (b) of $T$. In addition,

$$\mathcal{H}_{\mathcal{V}}(T(P)) = \mathcal{H}(T(P), \mathcal{V}(T(P))) \le \mathcal{H}(T(P), \mathcal{V}(P)) ,$$

because of (8). Hence, $\mathcal{H}_{\mathcal{V}}(T(P)) \le \mathcal{H}_{\mathcal{V}}(P)$, and the inequality is strict if $P$ is not centroidal. We then conclude that $\mathcal{H}_{\mathcal{V}}$ is a descent function for the algorithm $T$. The result now follows from the global convergence Theorem A.3 and Proposition A.4 in Appendix A. ■

*Remark 3.4:* Lloyd algorithm in quantization theory [11], [12] is usually presented as follows: given the location of $n$ agents, $p_1, \dots, p_n$, (i) construct the Voronoi partition corresponding to $P = (p_1, \dots, p_n)$; (ii) compute the mass centroids of the Voronoi regions found in step (i). Set the new location of the agents to these centroids; and return to step (i). Lloyd algorithm can also be seen as a fixed point iteration. Consider the mappings $LL_i : Q^n \to Q$ for $i \in \{1, \dots, n\}$

$$LL_i(p_1, \dots, p_n) = \left( \int_{V_i(P)} \phi(q) dq \right)^{-1} \int_{V_i(P)} q\phi(q) dq .$$

Let $LL : Q^n \to Q^n$ be defined by $LL = (LL_1, \dots, LL_n)$. Clearly, $LL$ is continuous (indeed, $C^1$), and corresponds to Lloyd algorithm. Now, $\|LL_i(P) - C_{V_i}\| = 0 \le \|p_i - C_{V_i}\|$, for all $i \in \{1, \dots, n\}$. Moreover, if $P$ is not centroidal, then the inequality is strict for all $p_i \ne C_{V_i}$. Therefore, $LL$ verifies properties (a) and (b).

## C. Remarks

(i) Note that different sensor performance functions $f$ in equation (1) correspond to different optimization problems. Provided one uses the Euclidean distance in the definition of $\mathcal{H}$ (cf. equation (1)), the standard Voronoi partition computed with respect to the Euclidean metric remains the optimal partition. For arbitrary $f$, it is not possible anymore to decompose $\mathcal{H}_{\mathcal{V}}$ into the sum of terms similar to $\mathcal{H}_{\mathcal{V},1}$ and $\mathcal{H}_{\mathcal{V},2}$. Nevertheless, it is still possible to implement the gradient flow via the expression for the partial derivative (3).
*Proposition 3.5:* Assume the sensors location obeys a first order dynamical behavior, $\dot{p}_i = u_i$. Then, for the closed-loop system induced by the gradient law (3), $u_i = -\partial \mathcal{H}_{\mathcal{V}} / \partial p_i$, the sensors location $P = (p_1, \dots, p_n)$ converges asymptotically to the set of critical points of $\mathcal{H}_{\mathcal{V}}$. Assuming this set is finite, the sensors location converges to a critical point.

(ii) More generally, various distance notions can be used to define locational optimization functions. Different performance function gives rise to corresponding notions of "center of a region" (any notion of geometric center, mean, or average is an interesting candidate). These can then be adopted in designing coverage algorithms. We refer to [36] for a discussion on Voronoi partitions based on non-Euclidean distance functions and to [5], [8] for a discussion on the corresponding locational optimization problems.

(iii) Next, let us discuss an interesting variation of the original problem. In [6], minimizing the expected minimum distance function $\mathcal{H}_{\mathcal{V}}$ in equation (2) is referred to as the *continuous $p$-median problem*. It is instructive to consider the worst-case minimum distance function, corresponding to the scenario where no information is available on the distribution density function. In other words, the network seeks to minimize the largest possible distance from any point in $Q$ to any of the sensor locations, i.e., to minimize the function

$$\max_{q \in Q} \left[ \min_{i \in \{1,\dots,n\}} \|q - p_i\| \right] = \max_{i \in \{1,\dots,n\}} \left[ \max_{q \in V_i} \|q - p_i\| \right] .$$

This optimization is referred to as the *$p$-center problem* in [6], [7]. One can design a strategy for the $p$-center problem analog to the Lloyd algorithm for the $p$-median problem: each vehicle moves, in continuous or discrete-time, toward the center of the minimum-radius sphere enclosing the polytope. We refer to [37] for a convergence analysis of the continuous-time algorithms.

In what follows, we shall restrict our attention to the $p$-median problem and to centroidal Voronoi partitions.

## D. Computations over polygons with uniform density

In this section, we investigate closed-form expression for the control laws introduced above. Assume the Voronoi region $V_i$ is a convex polygon (i.e., a polytope in $\mathbb{R}^2$) with $N_i$ vertexes labeled $\{(x_0, y_0), \dots, (x_{N_i-1}, y_{N_i-1})\}$ such as in Fig. 2. It is convenient to define $(x_{N_i}, y_{N_i}) = (x_0, y_0)$. Furthermore, we assume that the density function is $\phi(q) = 1$. By evaluating the

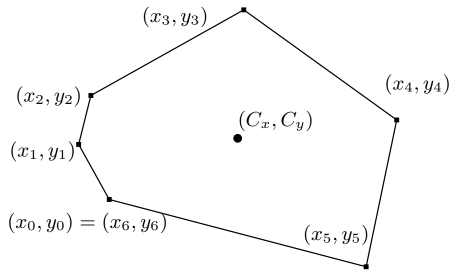

Fig. 2. Notation conventions for a convex polygon.

corresponding integrals, one can obtain the following closed-form expressions

$$\begin{aligned} M_{V_i} &= \frac{1}{2} \sum_{k=0}^{N_i-1} (x_k y_{k+1} - x_{k+1} y_k) \\ C_{V_i,x} &= \frac{1}{6M_{V_i}} \sum_{k=0}^{N_i-1} (x_k + x_{k+1})(x_k y_{k+1} - x_{k+1} y_k) \\ C_{V_i,y} &= \frac{1}{6M_{V_i}} \sum_{k=0}^{N_i-1} (y_k + y_{k+1})(x_k y_{k+1} - x_{k+1} y_k) \, . \end{aligned} \tag{9}$$

To present a simple formula for the polar moment of inertia, let $\bar{x}_k = x_k - C_{V_i,x}$ and $\bar{y}_k = y_k - C_{V_i,y}$, for $k \in \{0, \dots, N_i - 1\}$. Then, the polar moment of inertia of a polygon about its centroid, $J_{V_i,C}$ becomes

$$\begin{aligned} J_{V_i,C_{V_i}} = \frac{1}{12} \sum_{k=0}^{N_i-1} (\bar{x}_k \bar{y}_{k+1} - \bar{x}_{k+1} \bar{y}_k) \cdot \\ (\bar{x}_k^2 + \bar{x}_k x_{k+1} + \bar{x}_{k+1}^2 + \bar{y}_k^2 + \bar{y}_k \bar{y}_{k+1} + \bar{y}_{k+1}^2) \, . \end{aligned}$$

The proof of these formulas is based on decomposing the polygon into the union of disjoint triangles. We refer to [38] for analog expressions over $\mathbb{R}^N$.

Note also that the Voronoi polygon's vertexes can be expressed as a function of the neighboring vehicles. The vertexes of the $i$th Voronoi polygon that lie in the interior of $Q$ are the circumcenters of the triangles formed by $p_i$ and any two neighbors adjacent to $p_i$. The circumcenter of the triangle determined by $p_i$, $p_j$, and $p_k$ is

$$\begin{aligned} \frac{1}{4M} \Big( \|\alpha_{kj}\|^2 (\alpha_{ji} \cdot \alpha_{ik}) p_i + \|\alpha_{ik}\|^2 (\alpha_{kj} \cdot \alpha_{ji}) p_j \\ + \|\alpha_{ji}\|^2 (\alpha_{ik} \cdot \alpha_{kj}) p_k \Big) \, , \end{aligned} \tag{10}$$

where $M$ is the area of the triangle, and $\alpha_{ls} = p_l - p_s$.

Equation (9) for a polygon's centroid and equation (10) for the Voronoi cell's vertexes lead to a closed-form *algebraic* expression for the control law in equation (6) as a function of the neighboring vehicles' location.

## E. Numerical simulations

To illustrate the performance of the continuous-time Lloyd algorithm, we include some simulation results. The algorithm is implemented in `Mathematica` as a single centralized program. For the $\mathbb{R}^2$ setting, the code computes the bounded Voronoi diagram using the `Mathematica` package `ComputationalGeometry`, and computes mass, centroid, and polar moment of inertia of polygons via the numerical integration routine `NIntegrate`. Careful attention was paid to numerical accuracy issues in the computation of the Voronoi diagram and in the integration. We illustrate the performance of the closed-loop system in Fig. 3.

## IV. Asynchronous distributed implementations

In this section we show how the Lloyd gradient algorithm can be implemented in an asynchronous distributed fashion. In Section IV-A we describe our model for a network of robotic agents, and we introduce a precise notion of *distributed* evolution. Next, we provide two distributed algorithms for the local computation and maintenance of the Voronoi cells. Finally, in Section IV-C we propose two distributed asynchronous implementations of Lloyd algorithm: the first one is based on the gradient optimization algorithms as described in [32] and the second one relies on the special structure of the coverage problem.

## A. Modeling an asynchronous distributed network of mobile robotic agents

We start by modeling a robotic agent that performs sensing, communication, computation, and control actions. We are interested in the behavior of the asynchronous network resulting from the interaction of finitely many robotic agents. A framework to formalize the following concepts is the theory of distributed algorithms; see [30].

Let us here introduce the notion of *robotic agent with computation, communication, and control capabilities* as the $i$th element of a network. The $i$th agent has a processor with the ability of allocating continuous and discrete states and performing operations on them. Each vehicle has access to its unique identifier $i$. The $i$th agent occupies a location $p_i \in Q \subset \mathbb{R}^N$ and it is capable of moving in space, at any time $t \in \mathbb{R}_+$ for any period of time $\delta t \in \mathbb{R}_+$, according to a first order dynamics of the form:

$$\dot{p}_i(s) = u_i, \qquad \|u_i\| \leq 1 \, , \quad \forall s \in [t, t + \delta t]. \tag{11}$$

The processor has access to the agent's location $p_i$ and determines the *control pair* $(\delta t, u_i)$. The processor of the $i$th agent has access to a local clock $t_i \in \mathbb{R}_+$, and a *scheduling sequence*, i.e., an increasing sequence of times $\{T_{i,k} \in \mathbb{R}_+ \mid k \in \mathbb{N}_0\}$ such that $T_{i,0} = 0$ and $0 < t_{i,\text{min}} < T_{i,k+1} - T_{i,k} < t_{i,\text{max}}$. The processor of the $i$th agent is capable of transmitting information to any other agent within a closed disk of radius $R_i \in \mathbb{R}_+$. We assume the communication radius $R_i$ to be a quantity controllable by the $i$th processor and the corresponding communication bandwidth to be limited. We represent the information flow between the agents by means of "send"

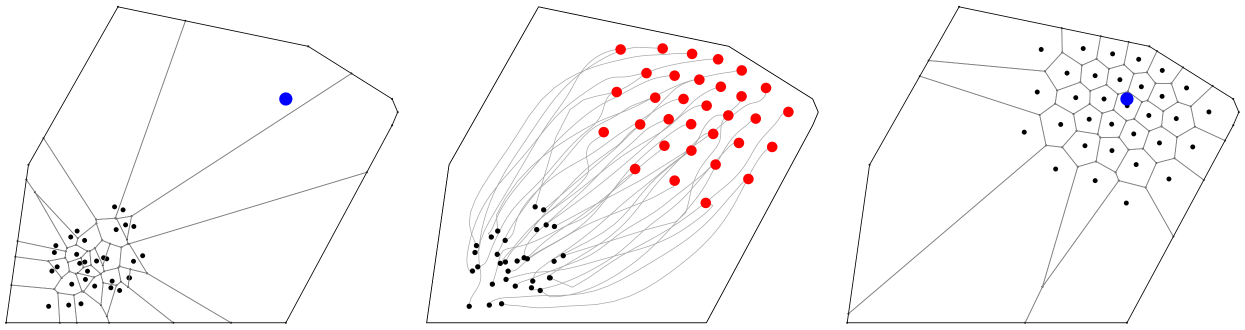

Fig. 3. Lloyd continuous-time algorithm for 32 agents on a convex polygonal environment, with the Gaussian density function of Fig. 1. The control gain in (6) is $k_{\text{prop}} = 1$ for all the vehicles. The left (respectively, right) figure illustrates the initial (respectively, final) locations and Voronoi partition. The central figure illustrates the gradient descent flow.

(within specified radius $R_i$) and "receive" commands with a finite number of arguments.

We shall alternatively consider networks of *robotic agents with computation, sensing, and control capabilities*. In this case, the processor of the $i$th agent has the same computation and control capabilities as before. Furthermore, we assume the processor can detect any other agent within a closed disk of radius $R_i \in \mathbb{R}_+$. We assume the sensing radius $R_i$ to be a quantity controllable by the processor.

*Remark 4.1:* We assume all communication between agents and all sensing of agents locations to be always accurate and instantaneous.

Consider the closed-loop system formed by the evolution of the $n$ agents of a network according to equation (11). The network evolution is said to be *Voronoi-distributed* if each $u_i(p_1, \dots, p_n)$ can be written as a function of the form $u_i(p_i, p_{i_1}, \dots, p_{i_m})$, with $i_k \in \mathcal{N}(i)$, $k \in \{1, \dots, m\}$. It is well known that there are at most $3n - 6$ neighborhood relationships in a planar Voronoi diagram [9, see Section 2.3]. As a consequence, the number of Voronoi neighbors of each site is on average less than or equal to 6, i.e., $m \leq 6$. (Recall that sites are Voronoi-neighbors if they share an edge, not just a vertex.) Accordingly, we argue that Voronoi-distributed algorithms lead to scalable networks. Finally, note that the set of indexes $\{i_1, \dots, i_m\}$ for a specific generator $p_i$ of a Voronoi-distributed dynamical system is not the same for all possible configurations of the network. In other words, the identity of the Voronoi neighbors changes along the evolution, i.e., the topology of the closed-loop system is *dynamic*.

## B. Voronoi cell computation and maintenance

A key requirement of the Lloyd algorithms presented in Section III is that each agent must be able to compute its own Voronoi cell. To do so, each agent needs to know the relative location (distance and bearing) of each Voronoi neighbor. The ability of locating neighbors plays a central role in numerous algorithms for localization, media access, routing, and power control in ad-hoc wireless communication networks; e.g., see [39], [40], [41] and references therein. Therefore, any motion control scheme might be able to obtain this information from the underlying communication layer. In what follows, we set out to provide a distributed asynchronous algorithm for the local computation and maintenance of Voronoi cells. The algorithm is related to the synchronous scheme in [41] and is based on basic properties of Voronoi diagrams.

We present the algorithm for a robotic agent with sensing capabilities (as well as computation and control). The processor of the $i$th agent allocates the information it has on the position of the other agents in the state variable $P^i$. The objective is to determine the smallest distance $R_i$ for agent $i$ which provides enough information to compute the Voronoi cell $V_i$. We start by noting that $V_i$ is a subset of the convex set

$$W(p_i, R_i) = \overline{B}(p_i, R_i) \cap \big( \cap_{j : \|p_i - p_j\| \leq R_i} S_{ij} \big), \qquad (12)$$

where $\overline{B}(p_i, R_i) = \{q \in Q \mid \|q - p_i\| \leq R_i\}$ and the half planes $S_{ij}$ are

$$\{q \in \mathbb{R}^N \mid \|q - p_i\| \leq \|q - p_j\|\}.$$

Provided $R_i$ is twice as large as the maximum distance between $p_i$ and the vertexes of $W(p_i, R_i)$, one can show that all Voronoi neighbors of $p_i$ are within distance $R_i$ from $p_i$ and the equality $V_i = W(p_i, R_i)$ holds. The minimum adequate sensing radius is therefore $R_{i,\min} = 2 \max_{q \in W(p_i, R_{i,\min})} \|p_i - q\|$. This argument guarantees the correctness of the ADJUST SENSING RADIUS ALGORITHM in Table I. The execution of this algorithm is illustrated in Fig. 4.

TABLE I
ADJUST SENSING RADIUS ALGORITHM

| | |
|---|---|
| **Name:** | ADJUST SENSING RADIUS ALGORITHM |
| **Goal:** | distributed Voronoi cell |
| **Requires:** | sensor with controllable radius $R_i$ |

```
At time $t_i$, local agent $i$ performs:
1: initialize $R_i$, detect all $p_j$ within radius $R_i$
2: update $P^i(t_i)$, compute $W(p_i(t_i), R_i)$
3: while $R_i < 2 \max_{q \in W(p_i(t_i), R_i)} \|p_i(t_i) - q\|$ do
4:     set $R_i := 2R_i$
5:     detect all $p_j$ within radius $R_i$
6:     update $P^i(t_i)$, compute $W(p_i(t_i), R_i)$
7: end while
8: set $R_i := 2 \max_{q \in W(p_i(t_i), R_i)} \|p_i(t_i) - q\|$
9: set $V_i := W_i(p_i(t_i), R_i)$
```

A similar ADJUST COMMUNICATION RADIUS ALGORITHM algorithm can be designed for a robotic agent with communication capabilities. The specifications go as in the previous

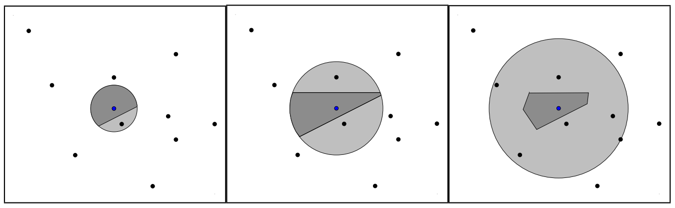

Fig. 4. An execution (from left to right) of the ADJUST SENSING RADIUS ALGORITHM: the sensing disk $\overline{B}(p_i, R_i)$ is in light gray, and the Voronoi cell estimate $W(p_i, R_i)$ is the darker gray region.

algorithm, except for the fact that steps `2:` and `7:` are substituted by

$$\begin{array}{l}\text{send } \big(\text{“request to reply”}, p_i(t_i)\big) \text{ within radius } R_i \\ \text{receive } \big(\text{“response”}, p_j\big) \text{ from all agents within radius } R_i\end{array}$$

Further, we have to require each agent to perform the following event-driven task: if the $i$th agent receives at any time $t_i$ a “request to reply” message from the $j$th agent located at position $p_j$, it executes

$$\text{send } \big(\text{“response”}, p_i(t_i)\big) \text{ within radius } \|p_i(t) - p_j\|$$

Next, we present the MONITORING ALGORITHM (cf. Table II), whose objective is to maintain the information about the Voronoi cell of the $i$th agent, and detect certain events. We consider robotic agents with sensing capabilities. We call an agent *active* if it is moving and we assume the $i$th agent can determine if any agent within radius $R_i$ is active or not. It turns out that (only) the following two events are of interest: (i) a Voronoi neighbor of the $i$th agent becomes active, and (ii) an active agent becomes a Voronoi neighbor of the $i$th agent. In both cases, we record the event by setting a Boolean variable *event* to `true` (as we shall later show, this will trigger an appropriate control action). The map $\mathrm{weight} : P^i \in \mathbb{R}^{N\times n} \mapsto (w_1, \dots, w_n) \in \mathbb{N}_0^n$ in Table II is defined by

$$w_j = \begin{cases} 3 & \text{if } j \in \mathcal{N}(i) \text{ and } j \text{ is active} \\ 1 & \text{if } j \in \mathcal{N}(i) \text{ and } j \text{ is not active} \\ 0 & \text{if } j \notin \mathcal{N}(i)\,. \end{cases}$$

We denote by $\mathrm{weight}_j$ the $j$th component of $\mathrm{weight}$. The algorithm is designed to run for times $t_i \in [t_0, t_0 + \delta t]$.

The correctness of the MONITORING ALGORITHM is guaranteed by the following argument: if an event of type (i) occurs at time $t_i \in [t_0, t_0 + \delta t]$, i.e., an agent (say, the $j$th) that is a Voronoi neighbor of the $i$th agent becomes active, then $\mathrm{weight}_j(P^i(t_i)) - w_j = 2$, and therefore *event* is set to `true`. Similarly, if an event of type (ii) occurs at time $t_i \in [t_0, t_0 + \delta t]$, i.e., a new active agent (say, the $j$th) becomes a Voronoi neighbor of the $i$th agent, then $\mathrm{weight}_j(P^i(t_i)) - w_j = 3$, and *event* is set to `true`.

### C. Asynchronous distributed implementations of coverage control

Let us now present two versions of Lloyd algorithm for the solution of the optimization problem (1) that can be implemented by an asynchronous distributed network of robotic agents. For simplicity, we assume that at time 0 all clocks are synchronized (although they later can run at different speeds) and that each agent knows at 0 the exact location of every other agent. The COVERAGE BEHAVIOR ALGORITHM I (cf. Table III) is designed for robotic agents with communication capabilities, and requires the ADJUST COMMUNICATION RADIUS ALGORITHM (while it does not require the MONITORING ALGORITHM).

TABLE II
MONITORING ALGORITHM

```
Name:      MONITORING ALGORITHM
Goal:      Cell maintenance & event detection
Requires:  (i) sensor with controllable radius $R_i$
           (ii) positive reals $t_0$, $\delta t$
           (iii) ADJUST SENSING RADIUS
                    ALGORITHM
-----------------------------------------------------------
Local agent $i$ performs for $t_i \in [t_0, t_0 + \delta t]$:
 1: initialize $P^i(t_0)$ and $V_i(t_0)$, set $w :=$ weight($P^i(t_0)$)
 2: while $t_i \le t_0 + \delta t$ do
 3:    run ADJUST SENSING RADIUS ALGORITHM
 4:    if for any $j$, $|\mathrm{weight}_j(P^i(t_i)) - w_j| \ge 2$ then
 5:       if $\mathrm{weight}_j(P^i(t_i)) - w_j \ge 2$ then
 6:          set event := true
 7:       end if
 8:       set $w :=$ weight($P^i(t_i)$)
 9:    end if
10: end while
```

TABLE III
COVERAGE BEHAVIOR ALGORITHM I

```
Name:      COVERAGE BEHAVIOR ALGORITHM I
Goal:      distributed optimal agent location
Requires:  (i) Voronoi cell computation
           (ii) centroid and mass computation
           (iii) positive real $\delta_0$
           (iv) ADJUST COMMUNICATION RADIUS
                   ALGORITHM
-----------------------------------------------------------
For $i \in \{1, \dots, n\}$, $i$th agent performs at $t_i = T_{i,0} = 0$:
 0: set $P^i(T_{i,0}) := (p^i_1(T_{i,0}), \dots, p^i_n(T_{i,0}))$
 0: compute Voronoi region $V_i(T_{i,0})$
 0: set $V_i := V_i(T_{i,0})$ and $R_i := 2\max_{q \in V_i} \|p_i - q\|$

For $i \in \{1, \dots, n\}$, the $i$th agent performs at time $t_i = T_{i,k}$ either one
of the following threads or both. For some $B_i \in \mathbb{N}$, we require that each
thread is executed at least once every $B_i$ steps of the scheduling sequence.

[Information thread]
 1: run ADJUST COMMUNICATION RADIUS ALGORITHM

[Control thread]
 1: compute centroid $C_{V_i}$ and mass $M_{V_i}$ of $V_i$
 2: apply control pair $\big(\delta_0,\ M_{V_i}(C_{V_i} - p_i(T_{i,k}))\big)$
```

As a consequence of the results in [32, Theorem 3.1 and Corollary 3.1] (see Appendix B, Theorem B.2 below for a brief exposition), we have the following result.

*Proposition 4.2:* Let $P_0 \in Q^n$ denote the initial sensors location. Let $\{T_k\}$ be the sequence in increasing order of all the scheduling sequences of the agents of the network. Assume $\inf_k\{T_k - T_{k-1}\} > 0$. Then, there exists a sufficiently small $\delta_* > 0$ such that if $0 < \delta_0 \le \delta_*$, the COVERAGE BEHAVIOR ALGORITHM I converges to the set of critical points of $\mathcal{H}_{\mathcal{V}}$, that is, the set of centroidal Voronoi configurations.

Next, we focus on distributed asynchronous implementations of Lloyd algorithm that take advantage of the special structure of the coverage problem. The COVERAGE BEHAVIOR ALGORITHM II (cf. Table IV) is designed for robotic agents with sensing capabilities, it requires the Monitoring and the Adjust sensing radius algorithms. Two advantages of this algorithm over the previous one are that there is no need for each agent to exactly go toward the centroid of its Voronoi cell nor to take a small step at each stage.

TABLE IV
COVERAGE BEHAVIOR ALGORITHM II

```
Name:       COVERAGE BEHAVIOR ALGORITHM II
Goal:       distributed optimal agent location
Requires:   (i) Voronoi cell computation
            (ii) centroid computation
            (iii) MONITORING ALGORITHM
-----------------------------------------------------------
For $i \in \{1,\dots,n\}$, $i$th agent performs at $t_i = T_{i,0} = 0$:
 0: set $P^i(T_{i,0}) := (p_1^i(T_{i,0}),\dots,p_n^i(T_{i,0}))$
 0: compute Voronoi region $V_i(T_{i,0})$
 0: set $V_i := V_i(T_{i,0})$ and $R_i := 2\max_{q\in V_i}\|p_i - q\|$
For $i \in \{1,\dots,n\}$, $i$th agent performs at $t_i = T_{i,k}$:
 1: choose $0 < \delta t_i < t_{i,min}$
 2: set $s := T_{i,k}$, compute centroid $C_{V_i}(s)$
 3: choose $u_i$, with $u_i \cdot (C_{V_i} - p_i(s)) \geq 0$, with strict inequality if $p_i(s) \neq$
    $C_{V_i}$, set event := false
 4: while $t_i \leq T_{i,k} + \delta t_i$ do
 5:   run MONITORING ALGORITHM for $(T_{i,k}, \delta t_i)$
 6:   while event = false do
 7:     $\dot{p}_i = u_i$
 8:   end while
 9:   set $s := t_i$, compute centroid $C_{V_i(s)}$
10:   choose $u_i$, with $u_i \cdot (C_{V_i} - p_i(s)) \geq 0$, with strict inequality if
      $p_i(s) \neq C_{V_i}$, set event := false
11: end while
```

*Remark 4.3:* The control law in step 7: of the COVERAGE BEHAVIOR ALGORITHM II can be defined via a saturation function. For instance, $\mathrm{SR} : \mathbb{R}^N \to \mathbb{R}^N$

$$\mathrm{SR}(x) = \begin{cases} x & \text{if } \|x\| \leq 1 \\ x/\|x\| & \text{if } \|x\| \geq 1 \end{cases}$$

Then set $u_i = \mathrm{SR}(C_{V_i} - p_i)$.

With respect to the correctness of the COVERAGE BEHAVIOR ALGORITHM II, one can consider the time instants when the computation of the centroid of the Voronoi region of any agent is made, together with the time instants when any agent decide to stop, and regard the execution of this algorithm as a discrete-time mapping. Resorting to the discussion in Section III-B on the convergence of the discrete Lloyd algorithms, one can prove that the Coverage behavior algorithm II verifies properties (a) and (b). As a consequence of Proposition 3.3, we then have the following result.

*Proposition 4.4:* Let $P_0 \in Q^n$ denote the initial sensors location. The COVERAGE BEHAVIOR ALGORITHM II converges to the set of critical points of $\mathcal{H}_{\mathcal{V}}$, that is, the set of centroidal Voronoi configurations.

## V. EXTENSIONS AND APPLICATIONS

In this section we investigate various extensions and applications of the algorithms proposed in the previous sections. We extend the treatment to vehicles with passive dynamics and we also consider discrete-time implementations of the algorithms for vehicles endowed with a local motion planner. Finally, we describe interesting ways of designing density functions to solve problems apparently unrelated to coverage.

### *A. Variations on vehicle dynamics*

Here, we consider vehicles systems described by more general linear and nonlinear dynamical models.

*Coordination of vehicles with passive dynamics*. We start by considering the extension of the control design to nonlinear control systems whose dynamics is passive [42]. Relevant examples include networks of vehicles and robots with general Lagrangian dynamics, as well as spatially invariant passive linear systems. Specifically, assume that for each $i \in \{1,\dots,n\}$, the $i$th vehicle state includes the spatial variable $p_i$, and that the vehicle's dynamics is passive with input $u_i$, output $\dot{p}_i$ and storage function $S_i : Q \to \mathbb{R}_+$. Furthermore, assume that the input preserving the zero dynamics manifold $\{\dot{p}_i = 0\}$ is $u_i = 0$.

For such systems, we devise a proportional derivative (PD) control via,

$$u_i = -k_{\text{prop}} M_{V_i}(p_i - C_{V_i}) - k_{\text{deriv}}\dot{p}_i, \qquad (13)$$

where $k_{\text{prop}}$ and $k_{\text{deriv}}$ are scalar positive gains. The closed-loop system induced by this control law can be analyzed with the Lyapunov function

$$\mathcal{E} = \frac{1}{2} k_{\text{prop}} \mathcal{H}_{\mathcal{V}} + \sum_{i=1}^{n} S_i,$$

yielding the following result.

*Proposition 5.1:* For passive systems, the control law (13) achieves asymptotic convergence of the sensors location to the set of centroidal Voronoi configurations. If this set is finite, then the sensors location converges to a centroidal Voronoi configuration.

*Proof:* Consider the evolution of the function $\mathcal{E}$,

$$\begin{aligned} \frac{d}{dt}\mathcal{E} &= \frac{1}{2} k_{\text{prop}} \frac{d}{dt}\mathcal{H}_{\mathcal{V}} + \sum_{i=1}^{n} \dot{S}_i \\ &\leq \sum_{i=1}^{n} \Big( k_{\text{prop}} M_{V_i}(p_i - C_{V_i}) \cdot \dot{p}_i + \dot{p}_i \cdot u_i \Big) \\ &= -k_{\text{deriv}} \sum_{i=1}^{n} \dot{p}_i^2 \leq 0\,. \end{aligned}$$

By LaSalle's principle, the sensors location converges to the largest invariant set contained in $\{\dot{p}_i = 0\}$. Given the assumption on the zero dynamics, we conclude that $p_i = C_{V_i}$ for $i \in \{1,\dots,n\}$, i.e., the largest invariant set corresponds to the set of centroidal Voronoi configurations. If this set is finite, LaSalle's principle also guarantees convergence to a specific centroidal Voronoi configuration. ■

In Fig. 5 we illustrate the performance of the control law (13) for vehicles with second-order dynamics $\ddot{p}_i = u_i$ and storage function $S_i = \frac{1}{2}\dot{p}_i^2$.

*Coordination of vehicles with local controllers*. Next, consider the setting where each vehicle has an arbitrary dynamics and is endowed with a local feedback and feedforward

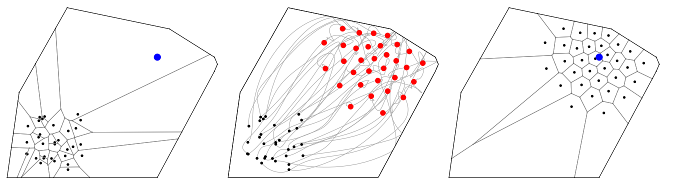

Fig. 5. Coverage control for 32 vehicles with second order dynamics. The environment and Gaussian density function are as in Fig. 3. The control gains are $k_{\text{prop}} = 6$ and $k_{\text{deriv}} = 1$.

controller. The controller is capable of strictly decreasing the distance to any specified position in $Q$ in a specified period of time $\delta$.

Assume the dynamics of the $i$th vehicle is described by $\dot{x}_i = f_i(t, x_i, u)$, where $x_i \in \mathbb{R}^m$ denotes its state, and $\pi_i : \mathbb{R}^{m_i} \to Q$ is such that $\pi_i(x_i) = p_i$. Assume also that for any $p_{\text{target}} \in Q$ and any $x_0 \in \mathbb{R}^m \setminus \pi_i^{-1}(p_{\text{target}})$, there exists $\overline{u}(t, x(t), p_{\text{target}})$ such that the solution $\overline{x}_i(t)$ of

$$\dot{\overline{x}}_i = f_i(t, \overline{x}_i(t), \overline{u}(t, \overline{x}_i(t), p_{\text{target}}))\,, \quad \overline{x}_i(0) = x_0\,,$$

verifies $\|\pi_i(\overline{x}_i(t_0 + \delta)) - p_{\text{target}}\| < \|\pi_i(\overline{x}_i(t_0)) - p_{\text{target}}\|$.

*Proposition 5.2:* Consider the following coordination algorithm. At time $t_k = k\delta$, $k \in \mathbb{N}$, each vehicle computes $V_i(t_k)$ and $C_{V_i}(t_k)$; then, for time $t \in [t_k, t_{k+1}[$, the vehicle executes $\overline{u}(t, x(t), C_{V_i}(t_k))$. For this closed-loop system, the sensors location converges to the set of centroidal Voronoi configurations. If this set is finite, then the sensors location converges to a centroidal Voronoi configuration.

The proof of this result readily follows from Proposition 3.3, since the algorithm verifies properties (a) and (b) of Section III-B.

As an example, we consider a classic model of mobile wheeled dynamics, the *unicycle model*. Assume the $i$th vehicle has configuration $(\theta_i, x_i, y_i) \in \mathrm{SE}(2)$ evolving according to

$$\dot{\theta}_i = \omega_i\,, \quad \dot{x}_i = v_i \cos\theta_i\,, \quad \dot{y}_i = v_i \sin\theta_i\,,$$

where $(\omega_i, v_i)$ are the control inputs for vehicle $i$. Note that the definition of $(\theta_i, v_i)$ is unique up to the discrete action $(\theta_i, v_i) \mapsto (\theta_i + \pi, -v_i)$. Given a target point $p_{\text{target}}$, we use this symmetry to require the equality $(\cos\theta_i,\ \sin\theta_i)\cdot(p_i - p_{\text{target}}) \leq 0$ for all time $t$. Should the equality be violated at some time $t = t_0$, we shall redefine $\theta_i(t_0^+) = \theta_i(t_0^-) + \pi$ and $v_i$ as $-v_i$ from time $t = t_0$ onward.

Following the approach in [43], consider the control law

$$\begin{aligned} \omega_i &= 2k_{\text{prop}} \arctan \frac{(-\sin\theta_i,\ \cos\theta_i)\cdot(p_i - p_{\text{target}})}{(\cos\theta_i,\ \sin\theta_i)\cdot(p_i - p_{\text{target}})} \\ v_i &= -k_{\text{prop}}(\cos\theta_i,\ \sin\theta_i)\cdot(p_i - p_{\text{target}}), \end{aligned}$$

where $k_{\text{prop}}$ is a positive gain. This feedback law differs from the original stabilizing strategy in [43] only in the fact that no final angular position is preferred. One can prove that $p_i = (x_i, y_i)$ is guaranteed to monotonically approach the target position $p_{\text{target}}$ when run over an infinite time horizon. We illustrate the performance of the proposed algorithm in Fig. 6.

## B. Geometric patterns and formation control

Here we suggest the use of decentralized coverage algorithms as formation control algorithms, and we present various density functions that lead the multi-vehicle network to predetermined geometric patterns. In particular, we present simple density functions that lead to segments, ellipses, polygons, or uniform distributions inside convex environments.

Consider a planar environment, let $k$ be a large positive gain, and denote $q = (x, y) \in Q \subset \mathbb{R}^2$. Let $a, b, c$ be real numbers, consider the line $ax + by + c = 0$, and define the density function

$$\phi_{\text{line}}(q) = \exp(-k(ax + by + c)^2).$$

Similarly, let $(x_c, y_c)$ be a reference point in $\mathbb{R}^2$, let $a, b, r$ be positive scalars, consider the ellipse $a(x - x_c)^2 + b(y - y_c)^2 = r^2$, and define the density function

$$\phi_{\text{ellipse}}(q) = \exp\big(-k(a(x - x_c)^2 + b(y - y_c)^2 - r^2)^2\big).$$

Fig. 7 illustrates the performance of the closed-loop network corresponding to this density function. During the simulations, we observed that the convergence to the desired pattern was rather slow.

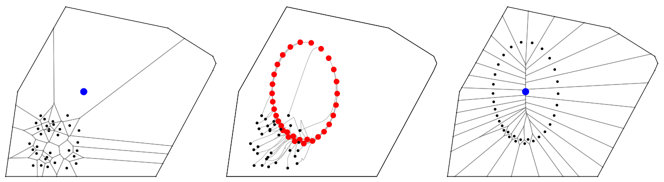

Fig. 7. Coverage control for 32 vehicles with $\phi_{\text{ellipse}}$. The parameter values are: $k = 500$, $a = 1.4$, $b = .6$, $x_c = y_c = 0$, $r^2 = .3$, and $k_{\text{prop}} = 1$.

Finally, define the smooth ramp function $\mathrm{SR}_\ell(x) = x(\arctan(\ell x)/\pi + (1/2))$, and the density function

$$\phi_{\text{disk}}(q) = \exp(-k\,\mathrm{SR}_\ell(a(x - x_c)^2 + b(y - y_c)^2 - r^2)).$$

This density function leads the multi-vehicle network to obtain a uniform distribution inside the ellipsoidal disk $a(x - x_c)^2 + b(y - y_c)^2 \leq r^2$. We illustrate this density function in Fig. 8.

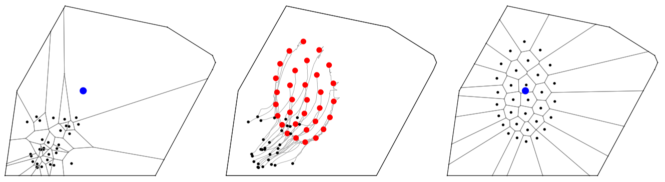

Fig. 8. Coverage control for 32 vehicles to an ellipsoidal disk. The density function parameters are the same as in Fig. 7, and $\ell = 10$, $k_{\text{prop}} = 1$.

It appears straightforward to generalize these types of density functions to the setting of arbitrary curves or shapes. The proposed algorithms are to be contrasted with the classic approach to formation control based on rigidly encoding the desired geometric pattern. One disadvantage of the proposed approach is the requirement for a careful numerical computation of Voronoi diagrams and centroids. We refer to [14], [15] for previous work on algorithms for geometric patterns, and to [17], [18] for formation control algorithms.

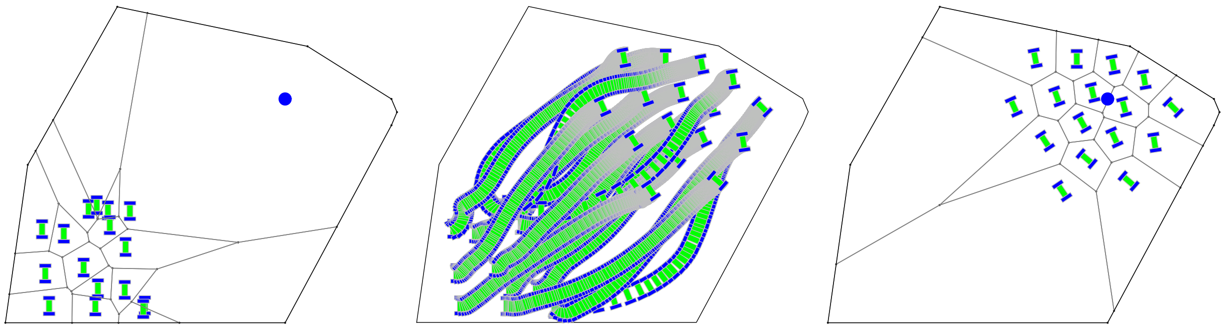

Fig. 6. Coverage control for 16 vehicles with mobile wheeled dynamics. The environment and Gaussian density function are as in Fig. 3, and $k_{\text{prop}} = 3$.

## VI. Conclusions

We have presented a novel approach to coordination algorithms for multi-vehicle networks. The scheme can be thought of as an interaction law between agents and as such it is implementable in a distributed scalable asynchronous fashion.

This paper leaves numerous important extensions open for further research. First, we envision considering the setting of structured environments (ranging all the way from simple non-convex polygon to more realistic ground, air and underwater environments); it might be useful for example to design distributed algorithms for the art gallery problem. Second, it is clearly important to consider non-isotropic sensors, such as cameras and directional microphones, as well as limited footprint sensors, as studied for example in the literature on coverage path planning. Third, we plan to extend the algorithms to provide collision avoidance guarantees and to vehicle dynamics which are not locally controllable. Finally, to investigate the effect of measurement errors on our proposed algorithms and to quantify their closed-loop robustness we are implementing these algorithms on a network of all-terrain vehicles. All these problems provide nontrivial challenges that go beyond our current treatment.

### *Acknowledgments*

This material is based upon work supported by ARO Grant DAAD 190110716, and DARPA/AFOSR MURI Award F49620-02-1-0325. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors.

## References

[1] C. R. Weisbin, J. Blitch, D. Lavery, E. Krotkov, C. Shoemaker, L. Matthies, and G. Rodriguez, "Miniature robots for space and military missions," *IEEE Robotics & Automation Magazine*, vol. 6, no. 3, pp. 9–18, 1999.

[2] E. Krotkov and J. Blitch, "The Defense Advanced Research Projects Agency (DARPA) tactical mobile robotics program," *International Journal of Robotics Research*, vol. 18, no. 7, pp. 769–76, 1999.

[3] P. E. Rybski, N. P. Papanikolopoulos, S. A. Stoeter, D. G. Krantz, K. B. Yesin, M. Gini, R. Voyles, D. F. Hougen, B. Nelson, and M. D. Erickson, "Enlisting rangers and scouts for reconnaissance and surveillance," *IEEE Robotics & Automation Magazine*, vol. 7, no. 4, pp. 14–24, 2000.

[4] T. B. Curtin, J. G. Bellingham, J. Catipovic, and D. Webb, "Autonomous oceanographic sampling networks," *Oceanography*, vol. 6, no. 3, pp. 86–94, 1993.

[5] A. Okabe, B. Boots, and K. Sugihara, "Nearest neighbourhood operations with generalized Voronoi diagrams: a review," *International Journal of Geographical Information Systems*, vol. 8, no. 1, pp. 43–71, 1994.

[6] Z. Drezner, Ed., *Facility Location: A Survey of Applications and Methods*, ser. Springer Series in Operations Research. New York, NY: Springer Verlag, 1995.

[7] A. Suzuki and Z. Drezner, "The $p$-center location problem in an area," *Location Science*, vol. 4, no. 1/2, pp. 69–82, 1996.

[8] A. Okabe and A. Suzuki, "Locational optimization problems solved through Voronoi diagrams," *European Journal of Operational Research*, vol. 98, no. 3, pp. 445–56, 1997.

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

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

[11] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982, presented as Bell Laboratory Technical Memorandum at a 1957 Institute for Mathematical Statistics meeting.

[12] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, 1998, commemorative Issue 1948-1998.

[13] H. Yamaguchi and T. Arai, "Distributed and autonomous control method for generating shape of multiple mobile robot group," in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, Munich, Germany, Sept. 1994, pp. 800–807.

[14] K. Sugihara and I. Suzuki, "Distributed algorithms for formation of geometric patterns with many mobile robots," *Journal of Robotic Systems*, vol. 13, no. 3, pp. 127–39, 1996.

[15] 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.

[16] 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.

[17] T. Balch and R. Arkin, "Behavior-based formation control for multirobot systems," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–39, 1998.

[18] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 905–8, 2001.

[19] R. Bachmayer and N. E. Leonard, "Vehicle networks for gradient descent in a sampled environment," in *IEEE Conf. on Decision and Control*, Las Vegas, NV, Dec. 2002, pp. 112–117.

[20] C. Tomlin, G. J. Pappas, and S. S. Sastry, "Conflict resolution for air traffic management: a study in multiagent hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 509–21, 1998.

[21] 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.

[22] E. Klavins, "Communication complexity of multi-robot systems," in *Algorithmic Foundations of Robotics V*, ser. STAR, Springer Tracts in

Advanced Robotics, J.-D. Boissonnat, J. W. Burdick, K. Goldberg, and S. Hutchinson, Eds., vol. 7. Berlin Heidelberg: Springer Verlag, 2003.

[23] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," Preprint, May 2003.

[24] R. A. Brooks, "A robust layered control-system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.

[25] R. C. Arkin, *Behavior-Based Robotics*. New York, NY: Cambridge University Press, 1998.

[26] M. S. Fontan and M. J. Mataric, "Territorial multi-robot task division," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 5, pp. 815–822, 1998.

[27] A. C. Schultz and L. E. Parker, Eds., *Multi-Robot Systems: From Swarms to Intelligent Automata*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 2002, proceedings from the 2002 NRL Workshop on Multi-Robot Systems.

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

[29] A. Howard, M. J. Mataric, 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.

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

[31] G. Tel, *Introduction to Distributed Algorithms*, 2nd ed. New York, NY: Cambridge University Press, 2001.

[32] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–12, 1986.

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

[34] H. Choset, "Coverage for robotics - a survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 113–126, 2001.

[35] E. U. Acar and H. Choset, "Sensor-based coverage of unknown environments: incremental construction of Morse decompositions," *International Journal of Robotics Research*, vol. 21, no. 4, pp. 345–366, 2002.

[36] R. Klein, *Concrete and abstract Voronoi diagrams*, ser. Lecture Notes in Computer Science. New York, NY: Springer Verlag, 1989, vol. 400.

[37] J. Cortés and F. Bullo, "Coordination and geometric optimization via distributed dynamical systems," *SIAM Journal on Control and Optimization*, May 2003, submitted.

[38] C. Cattani and A. Paoluzzi, "Boundary integration over linear polyhedra," *Computer-Aided Design*, vol. 22, no. 2, pp. 130–5, 1990.

[39] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu, "Geometric spanner for routing in mobile networks," in *ACM International Symposium on Mobile Ad-hoc Networking & Computing*, Long Beach, CA, Oct. 2001, pp. 45–55.

[40] X.-Y. Li and P.-J. Wan, "Constructing minimum energy mobile wireless networks," *ACM Journal of Mobile Computing and Communication Survey*, vol. 5, no. 4, pp. 283–286, 2001.

[41] M. Cao and C. Hadjicostis, "Distributed algorithms for Voronoi diagrams and application in ad-hoc networks," Preprint, Oct. 2002.

[42] A. J. van der Schaft, *L2-Gain and Passivity Techniques in Nonlinear Control*, 2nd ed. New York, NY: Springer Verlag, 1999.

[43] A. Astolfi, "Exponential stabilization of a wheeled mobile robot via discontinuous control," *ASME Journal on Dynamic Systems, Measurement, and Control*, vol. 121, no. 1, pp. 121–127, 1999.

[44] H. K. Khalil, *Nonlinear Systems*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1995.

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

## Appendix A
## Invariance and convergence principles

In this section we collect some relevant facts on descent flows both in the continuous and in the discrete-time settings. We do this following [44] and [45], respectively. We include Proposition A.4 as we are unable to locate it in the linear and nonlinear programming literature.

### *Continuous-time descent flows*

Consider the differential equation $\dot{x} = X(x)$, where $X : D \subset \mathbb{R}^N \to \mathbb{R}^N$ is locally Lipschitz and $D$ is an open connected set. A set $M$ is said to be (positively) invariant with respect to $X$ if $x(0) \in M$ implies $x(t) \in M$, for all $t \in \mathbb{R}$ (resp. $t \geq 0$). A descent function for $X$ on $\Omega$, $V : D \to \mathbb{R}$, is a continuously differentiable function such that $\mathcal{L}_X V \leq 0$ on $\Omega$. We denote by $E$ the set of points in $\Omega$ where $\mathcal{L}_X V(x) = 0$ and by $M$ be the largest invariant set contained in $E$. Finally, the distance from a point $x$ to a set $M$ is defined as $\operatorname{dist}(x, M) = \inf_{p \in M} \|x - p\|$.

*Lemma A.1 (LaSalle's principle):* Let $\Omega \subset D$ be a compact set that it is positively invariant with respect to $X$. Let $x(0) \in \Omega$ and $x_*$ be an accumulation point of $x(t)$. Then $x_* \in M$ and $\operatorname{dist}(x(t), M) \to 0$ as $t \to \infty$.

The following corollary is Exercise 3.22 in [44].

*Corollary A.2:* If the set $M$ is a finite collection of points, then the limit of $x(t)$ exists and equals one of them.

### *Discrete-time descent flows*

Let $X$ be a subset of $\mathbb{R}^N$. An algorithm $T$ is a continuous mapping from $X$ to $X$. A set $C$ is said to be positively invariant with respect to $T$ if $x_0 \in C$ implies $T(x_0) \in C$. A point $x_*$ is said to be a fixed point of $T$ if $T(x_*) = x_*$. We denote the set of fixed points of $T$ by $\Gamma$. A descent function for $T$ on $C$, $Z : X \to \mathbb{R}_+$, is any nonnegative real-valued continuous function satisfying $Z(T(x)) \leq Z(x)$ for $x \in C$, where the inequality is strict if $x \notin \Gamma$. Typically, $Z$ is the objective function to be minimized, and $T$ reflects this goal by yielding a point that reduces (or at least does not increase) $Z$.

*Lemma A.3 (Global convergence theorem):* Let $T : X \to X$ be an algorithm with a compact, positively invariant set $C \subset X$ and a descent function $Z$. Let $x_0 \in C$ and denote $x_m = T(x_{m-1})$, $m \geq 1$. Let $x_*$ be an accumulation point of the sequence $\{x_m \in C \mid m \in \mathbb{N}\}$. Then $x_* \in \Gamma$, $\operatorname{dist}(x_m, \Gamma) \to 0$ and $Z(x_m) \to Z(x_*)$ as $m \to \infty$.

*Proposition A.4:* If the set $\Gamma$ is a finite collection of points, then $\{x_m \in C \mid m \in \mathbb{N}\}$ converges and equals one of them.

*Proof:* Let $x_*$ be an accumulation point of $\{x_m\}$ and assume the whole sequence does not converge to it. Then, there exists an $\epsilon > 0$ such that for all $m_0$, there is a $m' > m_0$ such that $\|x_{m'} - x_*\| > \epsilon$. Let $d$ be the minimum of all the distances between the points in $\Gamma$. Fix $\epsilon' = \min\{d/2, \epsilon\}$. Since $T$ is continuous and $\Gamma$ is finite, there exists $\delta > 0$ such that $\|x - z\| < \delta$, with $z \in \Gamma$, implies $\|T(x) - z\| < \epsilon'$ (that is, for each $z \in \Gamma$, there exists such $\delta(z)$, and we take the minimum over $\Gamma$).

Now, since $\operatorname{dist}(x_m, \Gamma) \to 0$, there exists $m_1$ such that for all $m \geq m_1$, $\operatorname{dist}(x_m, \Gamma) < \delta$. Also, we know that there is a subsequence of $\{x_m \mid m \in \mathbb{N}\}$ which converges to $x_*$, let us denote it by $\{x_{m_k} \mid k \in \mathbb{N}\}$. For $\delta$, there exists $m_{k_1}$ such that for all $k \geq k_1$, we have $\|x_{m_k} - x_*\| < \delta$.

Let $m_0 = \max\{m_1, m_{k_1}\}$. Take $k$ such that $m_k \geq m_0$ Then,

$$\|x_{m_k+1} - x_*\| = \|T(x_{m_k}) - x_*\| < \epsilon' . \qquad (14)$$

Now we are going to prove that $\|x_{m_k+1}-x_*\| < \delta$. If $d/2 \leq \delta$, then this claim is straightforward, since $\epsilon' \leq d/2$. If $d/2 > \delta$, suppose that $\|x_{m_k+1} - x_*\| > \delta$. Since $m_k + 1 > m_0 \geq m_1$, then $\mathrm{dist}(x_{m_k+1}, \Gamma) < \delta$. Therefore, there exists $y \in \Gamma$ such that $\|x_{m_k+1} - y\| < \delta$. Necessarily, $y \neq x_*$. Now, by the triangle inequality, $\|x - y\| \leq \|x - x_{m_k+1}\| + \|x_{m_k+1} - y\|$. Then,

$$\|x_{m_k+1} - x_*\| \geq \|x - y\| - \|x_{m_{k+1}} - y\| \geq d - \delta > d/2\,,$$

which contradicts (14). Therefore, $\|x_{m_k+1} - x_*\| < \delta$. This argument can be iterated to prove that for all $m \geq m_0$, we have $\|x_m - x_*\| < \delta$. Let us take now $m' > m_0$ such that $\|x_{m'} - x_*\| > \epsilon$. Since $m' - 1 \geq m_0$, we have $\|x_{m'-1} - x_*\| < \delta$, and therefore

$$\|x_{m'} - x_*\| = \|T(x_{m'-1}) - x_*\| < \epsilon' \leq \epsilon\,,$$

which is a contradiction. Therefore, $\{x_m \mid m \in \mathbb{N}\}$ converges to $x_*$. ■

## Appendix B
## Asynchronous gradient algorithms

In this section, we present a brief account of the results in [32] concerning asynchronous gradient optimization algorithms. We do not review them in its full generality, but rather formulate them in a form readily applicable to our setting.

Let $H_1, \ldots, H_L$ be finite-dimensional real vector spaces and let $H = H_1 \times H_2 \times \cdots \times H_L$. If $x = (x_1, \ldots, x_L)$, $x_l \in H_l$, we refer to $x_l$ as the $l$th component of $x$. Let $\{1, \ldots, M\}$ be a set of processors that participate in the computation. The algorithms considered here evolve in discrete time. This restriction does not involve any loss of generality, since the events of interest (an update by a processor, a transmission of a message, etc.) may be indexed by a discrete variable. The value stored by the $i$th processor at time $n$ (global) is denoted by $x^i(n)$. This global clock is only need for analysis purposes. The processors may be working without having access to it: instead, they may have access to a local clock or to no clock at all.

Consider the *specialization setting* [32], where each processor updates a particular component of the vector $x$ specifically assigned to it and relies on the information provided by the other processors for the remaining components. Formally,

(i) $M = L$,
(ii) Processor $i$ may update only its own component $x_i^i$,
(iii) Processor $j$ only sends messages containing elements of $H_j$. If processor $i$ receives such a message, it uses it to reset the $x_j^i$ equal to the value received.

Let $T_i^i$ be the set of all times when processor $i$ performs a computation involving the $i$th component of $x$. If a message from processor $j$, containing an element of $H_j$, is received by processor $i$ at time $n$, let $t_j^{ij}(n)$ denote the time that this message was sent. The content of the message is therefore $x_j^j(t_j^{ij}(n))$. Naturally, it is assumed $t_j^{ij}(n) \leq n$ and we set $t_i^{ii}(n) = n$. Finally, $T_j^{ij}$ denotes the set of all times when processor $i$ receives a message from processor $j$.

Let $\mathcal{J} : H \to \mathbb{R}_+$ be a $C^1$-function whose derivative is a Lipschitz function. Consider the *deterministic gradient algorithm* given by

$$x_i^i(n+1) = \begin{cases} x_i^i(n) & n \notin T_i^i \\ -\gamma_0 \frac{\partial \mathcal{J}}{\partial x_i}(x^i(n)) & n \in T_i^i \end{cases} \tag{15a}$$

$$x_j^i(n+1) = \begin{cases} x_j^i(n) & n \notin T_j^{ij} \\ x_j^j(t_j^{ij}(n)) & n \in T_j^{ij} \end{cases} \tag{15b}$$

where $\gamma_0 > 0$, $n \in \mathbb{N}$ and $i, j \in \{1, \ldots, M\}$. The specific conclusions of Theorem 3.1 and Corollary 3.1 in [32] that we need for the specialization setting are presented in the following result.

*Theorem B.1:* Assume each processor $i$ communicates its components $x_i^i$ to every other processor at least once every $B_1$ time units, for some constant $B_1$. Then, there exists a constant $\gamma^* > 0$ such that if $0 < \gamma_0 \leq \gamma^*$, the deterministic gradient algorithm (15) verifies

$$\lim_{n\to\infty} \|x^i(n) - x^j(n)\| = 0 \quad \text{and} \quad \lim_{n\to\infty} \frac{\partial \mathcal{J}}{\partial x_i}(x^i(n)) = 0\,,$$

for all $i, j \in \{1, \ldots, M\}$.

In the particular case when, for each $i \in \{1, \ldots, M\}$, the partial derivative $\frac{\partial J}{\partial x_i}(x)$ only depends on $x_l$, with $l \in \mathcal{M}(x, i) \cup \{i\}$ for certain set $\mathcal{M}(x, i)$, the previous result can be restated in the following form.

*Theorem B.2:* Assume each processor $i$ communicates its components $x_i^i$ to every other processor in $\mathcal{M}(x, i)$ at least once every $B_1$ time units, for some constant $B_1$. Then, there exists a constant $\gamma^* > 0$ such that if $0 < \gamma_0 \leq \gamma^*$, the deterministic gradient algorithm (15) verifies

$$\lim_{n\to\infty} \|x_l^i(n) - x_l^j(n)\| = 0 \quad \text{and} \quad \lim_{n\to\infty} \frac{\partial \mathcal{J}}{\partial x_i}(x^i(n)) = 0\,,$$

for all $i, j \in \{1, \ldots, M\}$ and all $l \in (\mathcal{M}(x, i) \cup \{i\}) \cap (\mathcal{M}(x, j) \cup \{j\})$.