On Coverage Control for Limited Range Multi-Robot Systems

Federico Pratissoli, Beatrice Capelli, Lorenzo Sabattini

Abstract— **This paper presents a coverage based control algorithm to coordinate a group of autonomous robots. Most of the solutions presented in the literature rely on an *exact* Voronoi partitioning, whose computation requires complete knowledge of the environment to be covered. This can be achieved only by robots with unlimited sensing capabilities, or through communication among robots in a limited sensing scenario. To overcome these limitations, we present a distributed control strategy to cover an unknown environment with a group of robots with limited sensing capabilities and in the absence of reliable communication. The control law is based on a *limited* Voronoi partitioning of the sensing area, and we demonstrate that the group of robots can optimally cover the environment using only information that is locally detected (without communication). The proposed method is validated by means of simulations and experiments carried out on a group of mobile robots.**

I. Introduction

Various strategies have been introduced for implementing coverage control with networked mobile robots. In particular, most relevant to this paper are the results reported in [1], [2]. In these works, the authors presented decentralized coordination algorithms for groups of mobile agents based on Voronoi diagrams [3] and proximity graphs [4]. This approach gives a simple solution that guarantees the convergence of the networked robots to a configuration that maximizes the coverage of the environment. This strategy is based on a Voronoi partitioning of the whole environment, which is assumed to be known [5] or measurable by the robots [6]. There exist several algorithms to construct an *exact* Voronoi diagram by assuming that each robot can obtain the position of all the others [7] or by means of communication among robots and broadcasting of messages [8]–[10].

However, in most practical scenarios, robots do not necessarily have information about the location of all other robots and may not have the possibility to rely on a communication network: hence, the Voronoi partitioning needs to be studied in a sensor range constrained scenario. Thus, instead of an exact Voronoi cell we consider a *limited* Voronoi cell, which is a partitioning of the area within the sensing range of the robot. A distributed methodology was presented in [1], [2] to compute the exact Voronoi diagram for a limited sensing network, which is based on the algorithm presented in [11]. The exact Voronoi cell of each robot can be gradually refined by computing the partitioning within an adjustable sensing range and incrementally increasing the sensing range. The Adjust-Sensing-Radius algorithm has become the standard one in the field of distributed coverage control to solve the problem of the Voronoi partitioning. However, the objective of this algorithm is to determine the smallest distance R_i for agent i which provides enough information to compute the exact Voronoi cell V_i [1]. The algorithm requires sensors with controllable sensing range in order to determine the relative location of each Voronoi neighbor (more details will be given in Section II). In practical applications, the limited sensing capability of the robot is a constraint and rarely the sensing radius can be adjusted to detect sufficient information to compute the exact Voronoi diagram. An algorithm was proposed in [7] to efficiently compute the Voronoi cell. However, it is based on a reliable communication among the robots, which in real scenarios can not be always guaranteed (such as in underwater environments [12]). We consider a control strategy constrained by the information detected by the sensor (a communication network is not needed) and, thus, based on the Voronoi partitioning of the area within the sensing range (limited Voronoi diagram). An algorithm studied to perform efficiently this computation is described in [13].

The effect of the characteristics of robot sensing capabilities on the performance of coverage control algorithms have been recently addressed in the literature. In [14]–[16], the authors introduced agents heterogeneity into the standard coverage problem in order to consider the different sensing modalities. Differently, in [17] the authors focused on the anisotropic aspect of the limited sensing range. They proposed an alternative coverage problem based on a definition of a new proximity graph.

It is worth noting that all the aforementioned works are based on an exact Voronoi partitioning of the environment: hence, the control algorithm requires the location of the Voronoi neighbors and the knowledge of the boundary of the environment to compute each robot's Voronoi cell. The control strategy we propose can be applied to a group of robots deployed in an unknown environment, since the only information required by the control law can be locally acquired by on board sensors.

Moreover, we would like to remark that only a few works can be found in literature in which coverage control is implemented and tested on real robots: an example is provided by [6], where the authors tested the proposed strategy on a swarm of mobile robots. Similarly, in this work we validate the proposed methodology in real experiments on ground robots.

Contribution: In this paper we propose a definition and a characterization of the limited Voronoi partitioning, that allows robots to individually compute the partitioning of the

Department of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Italy {federico.pratissoli, beatrice.capelli, lorenzo.sabattini}@unimore.it

environment based on information available within a limited sensing range. The limited Voronoi partitioning will then be exploited to propose a novel control strategy that allows us to perform coverage (1) of an unknown environment, (2) in a limited sensing scenario, and (3) without requiring communication among the robots. The proposed control strategy is validated in simulations, and in real experiments on ground robots.

II. Notation and Definitions

We denote by $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{R}_{>0}$ the set of natural, real, real non-negative, and real positive numbers. Given $x \in \mathbb{R}^n$, let $\|x\|$ be the Euclidean norm.

Let $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ be a graph characterized by a set $\mathcal{U}$ of vertices and a set $\mathcal{E} \subseteq \mathcal{U} \times \mathcal{U}$ of edges. Given an edge $(i, j) \in \mathcal{E}$, then the vertex i is a neighbor of the vertex j. Let $\mathcal{N}_{\mathcal{G}}(i)$ be the set of neighbors of the vertex i in $\mathcal{G}$. A graph $\mathcal{G}$ is said to be *undirected* if $(i, j) \in \mathcal{E}$ implies $(j, i) \in \mathcal{E}$.

Let $\mathbb{F}(\mathbb{R}^2)$ be the collection of finite point sets in $\mathbb{R}^2$. We can denote an element of $\mathbb{F}(\mathbb{R}^2)$ as $\mathcal{P} = \{p_1, \ldots, p_n\} \subset \mathbb{R}^2$, where $\{p_1, \ldots, p_n\}$ are points in $\mathbb{R}^2$.

The *Voronoi partitioning* can be defined on a polygonal environment in $\mathbb{R}^2$. In the rest of the paper, we will use $Q \subset \mathbb{R}^2$ to denote such convex polygonal environment: it will be used, in particular, to denote the environment to be covered by the robots. An arbitrary point in Q is denoted by $q \in Q$. Let then $\mathcal{P}$ be a set of n points $\{p_1, \ldots, p_n\}$ in Q. The *exact* Voronoi partitioning generated by $\mathcal{P}$ consists in the set $\mathcal{V}(\mathcal{P}) = \{V_1(\mathcal{P}), \ldots, V_n(\mathcal{P})\}$, where:

$$V_i(\mathcal{P}) = \{q \in Q | \|q - p_i\| \leq \|q - p_j\|, \forall p_j \in \mathcal{P}\}. \quad (1)$$

In the following, for the sake of brevity, we will use the notation V_i to refer to $V_i(\mathcal{P})$. Two agents are said to be *Voronoi neighbors* if $V_i \cap V_j \neq \emptyset$. We refer to [18] for a major discussion about the Voronoi diagrams.

Let us define the *proximity graph* as a graph $\mathcal{G}$ in which the edge set $\mathcal{E}_{\mathcal{G}}$ depends on the location of the vertices. In this paper we consider graphs defined for points $\{p_1, \ldots, p_n\}$ in $\mathbb{R}^2$. Hence, we can define a proximity graph function $\mathcal{G} : \mathbb{F}(\mathbb{R}^2) \to \mathbb{G}(\mathbb{R}^2)$ that associates to $\mathcal{P} \in \mathbb{F}(\mathbb{R}^2)$ an undirected graph with the set $\mathcal{P}$ of vertices and the set $\mathcal{E}_{\mathcal{G}}(\mathcal{P})$ of edges, where $\mathcal{E}_{\mathcal{G}} : \mathbb{F}(\mathbb{R}^2) \to \mathbb{F}(\mathbb{R}^2 \times \mathbb{R}^2)$ has the property that $\mathcal{E}_{\mathcal{G}}(\mathcal{P}) \subseteq \mathcal{P} \times \mathcal{P} \backslash \ \mathrm{diag}(\mathcal{P} \times \mathcal{P})$.

We denote, for $p \in \mathbb{R}^2$ and $R \in \mathbb{R}_{>0}$, the closed and open ball in $\mathbb{R}^2$ centered at p with radius R with $\overline{B}(p, R) = \left\{q \in \mathbb{R}^2 | \|q - p\| \leq R\right\}$ and $B(p, R) = \left\{q \in \mathbb{R}^2 | \|q - p\| < R\right\}$, respectively.

Throughout the paper, we will use 1_S to represent the indicator function, which is defined as $1_S(q) = 1$, if $q \in S$, and $1_S(q) = 0$, if $q \notin S$.

The following proximity graphs are relevant to our discussion:

1) the *R-disk graph* $\mathcal{P} \mapsto \mathcal{G}_{disk}(\mathcal{P}, R) = (\mathcal{P}, \mathcal{E}_{\mathcal{G}_{disk}}(\mathcal{P}, R))$, where the edges are defined as:
$$\mathcal{E}_{\mathcal{G}_{disk}}(\mathcal{P}, R) = \{(p_i, p_j) \in \mathcal{P} \times \mathcal{P} \backslash \mathrm{diag}(\mathcal{P} \times \mathcal{P}) \mid \|p_i - p_j\| \leq R\};$$
2) the *Delaunay graph* $\mathcal{P} \mapsto \mathcal{G}_D(\mathcal{P}) = (\mathcal{P}, \mathcal{E}_{\mathcal{G}_D}(\mathcal{P}))$, where the edges are defined as:
$$\mathcal{E}_{\mathcal{G}_D}(\mathcal{P}) = \{(p_i, p_j) \in \mathcal{P} \times \mathcal{P} \backslash \mathrm{diag}(\mathcal{P} \times \mathcal{P}) \mid V_i(\mathcal{P}) \cap V_j(\mathcal{P}) \neq \emptyset\};$$
3) the *R-limited Delaunay graph* $\mathcal{P} \mapsto \mathcal{G}_{LD}(\mathcal{P}, R) = (\mathcal{P}, \mathcal{E}_{\mathcal{G}_{LD}}(\mathcal{P}, R))$, where edges $(p_i, p_j) \in \mathcal{P} \times \mathcal{P} \backslash \mathrm{diag}(\mathcal{P} \times \mathcal{P})$ are defined as:
$$(V_i(\mathcal{P}) \cap \overline{B}(p_i, R/2)) \cap (V_j(\mathcal{P}) \cap \overline{B}(p_j, R/2)) \neq \emptyset.$$

For each proximity graph $\mathcal{G} = \{\mathcal{G}_{disk}, \mathcal{G}_D, \mathcal{G}_{LD}\}$ we can define the respective set of neighbors of the point p as $\mathcal{N}_{\mathcal{G}}(p, \mathcal{P}) = \{q \in \mathcal{P} | (p, q) \in \mathcal{E}_{\mathcal{G}}(\mathcal{P} \cup \{p\})\}$.

III. Problem Statement

We consider a multi-robot system constituted by n robots that move in a 2-dimensional space. We assume each robot to be modeled as a single integrator system,[1] whose position $p_i \in \mathbb{R}^2$ evolves according to

$$\dot{p}_i = u_i, \quad (2)$$

where $u_i \in \mathbb{R}^2$ is the control input, $\forall i = 1, \ldots, n$. The set of robots is represented by $\mathcal{P} = \{p_1, \ldots, p_n\}$. We consider the following assumptions:

1) *Convex unknown environment*: the environment to be covered by the multi-robot system is supposed to be represented by a convex polytope Q, which is not known a priori by the robots.
2) *Limited sensing capabilities*: each robot is able to measure the position of neighboring robots and objects, and to detect the boundaries of the environment Q, within its limited sensing range (defined by a ball with radius $R \in \mathbb{R}_{>0}$);
3) *No communication capabilities*: robots do not exchange information among each other.

Based on these assumptions, each robot computes its control inputs based only on directly measurable information, within the sensing range. Hence, the information exchange infrastructure can be modeled as the R-disk graph $\mathcal{G}_{disk}(\mathcal{P}, R)$).

The problem addressed in this paper is then formalized as follows:

Problem *Define a distributed control strategy that allows a multi-robot system with limited sensing capabilities to perform coverage of an unknown convex environment Q without explicit communication.*

[1] We would like to remark that, even though the single integrator is a very simplified model, it can still effectively be exploited to control real mobile robots: using a sufficiently good trajectory tracking controller, the single integrator model can be used to generate velocity references for widely used mobile robotic platforms, such as wheeled mobile robots [19], and unmanned aerial vehicles [20].

IV. Background on Coverage Control

We will now briefly summarize the *standard* solution to the coverage problem, as presented in [1].

A performance function can be defined, to be maximized in order to obtain the optimal coverage of the group of robots over Q. The performance function is chosen to model how reliable is the measurement, at point $q \in Q$, performed by robot i whose position is p_i, as a function of the distance $\|q - p_i\|$. Therefore, the performance function can be defined as non-increasing differentiable function $f(\|q - p_i\|) : \mathbb{R}_{\geq 0} \to \mathbb{R}$.

Moreover, an integrable probability density function $\phi : Q \to \mathbb{R}_{\geq 0}$ can be defined in order to characterize the portions in Q where an event of interest occurs. Namely, ϕ is used to capture the relative importance of determined areas in the environment. Such probability density function may be known a priori to all the robots, or may be estimated, at run-time, based on local measurements, as discussed in [6]: hence, without loss of generality, we will hereafter assume the probability density function to be known to all the robots.

Given an *exact* Voronoi partitioning $\mathcal{V}(P)$ of the environment Q in, so called, n Voronoi cells $\{V_1, \dots, V_n\}$ (see Fig. 1a), the *optimization* function $H : Q \to \mathbb{R}$ can be formulated as follows:

$$\mathcal{H}(\mathcal{P}, \mathcal{V}) = \sum_{i=1}^{n} \int_{V_i} f(\|q - p_i\|)\phi(q)dq, \tag{3}$$

where each node is in charge of covering its own cell, and with a better coverage corresponding to a higher value of the function. In the literature, the performance function is usually chosen as: $f(x) = -x^2$. Therefore, the optimization function becomes:

$$\mathcal{H}_{\mathcal{V}}(\mathcal{P}) = -\sum_{i=1}^{n} \int_{V_i} \|q - p_i\|^2 \phi(q)dq = -\sum_{i=1}^{n} J_{V_i,p_i}, \tag{4}$$

where $\mathcal{H}_{\mathcal{V}}(\mathcal{P}) = \mathcal{H}(\mathcal{P}, \mathcal{V})$ and J_{V_i,p_i} indicates the polar moment of inertia of the region V_i about the point p_i. In order to solve the optimization problem, the gradient of the optimization function can be computed, obtaining:

$$\frac{\partial \mathcal{H}_{\mathcal{V}}}{\partial p_i}(\mathcal{P}) = 2M_{V_i}(C_{V_i} - p_i), \tag{5}$$

where M_{V_i} and C_{V_i} denote respectively the mass and the center of mass with respect to the density function ϕ of the Voronoi cell $V_i \subset Q$ of the robot i in position p_i. Therefore, the mass and centroid can be computed as follows:

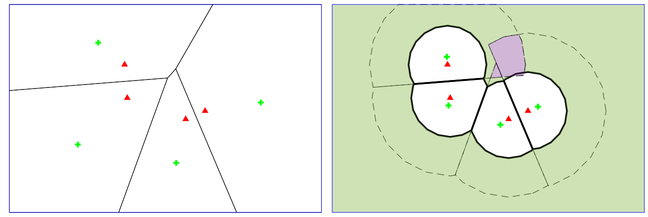
$$M_{V_i} = \int_{V_i} \phi(q)dq, \; C_{V_i} = \frac{1}{M_{V_i}} \int_{V_i} q\phi(q)dq. \tag{6}$$

For more details we refer the reader to [1], [6].

According to (5), the configuration of points $\mathcal{P}$ which maximizes the optimization function $\mathcal{H}_{\mathcal{V}}(\mathcal{P})$ coincides with the *centroids* of the respective Voronoi cells. In other words, the solution to the maximization problem is achieved when each agent is located at the centroid of its Voronoi region, such that $p_i = C_{V_i}, \; \forall i$. Such configurations are called *centroidal Voronoi configurations*, see [3]. $\mathcal{H}_{\mathcal{V}}(\mathcal{P})$ can be used as a Lyapunov-like function and we can design a control law that drives the group of robots to a centroidal Voronoi configuration, locally maximizing the coverage in the environment. In particular, the control input for each robot can be computed according to the Lloyd algorithm [1], as follows:

$$u_i = k(C_{V_i} - p_i), \tag{7}$$

where $k \in \mathbb{R}_{>0}$ is a proportional gain.



(a) *Exact* Voronoi $\mathcal{V}$. (b) *Limited* Voronoi $\mathcal{V}^r$.

Fig. 1. Comparison between *exact* and *limited* Voronoi. The environment Q is reported in *blue*, the robots are the *red* triangles, and the centroid of the single Voronoi cell are the *green* crosses, the single Voronoi cell are delimited with *solid black* lines. Fig. 1b also reports in *dashed black* lines the wrong partitioning built considering the whole sensing range R, and in *purple* the overlapping areas.

The exact Voronoi partitioning $\mathcal{V}(\mathcal{P})$ of the region Q is assumed to be continuously updated. Thus, it is assumed that the region Q is known or that each robot is able to measure it. The network of the group of robots is defined by the Delaunay proximity graph $\mathcal{G}_D(\mathcal{P})$. To compute the i^{th} component of the $\mathcal{H}_{\mathcal{V}}(\mathcal{P})$ function, it is necessary for the robot i to know its own position and the positions of its neighbors over the graph $\mathcal{G}_D(\mathcal{P})$. Usually the positions of all the robots are assumed to be known or measurable.

A modified optimization function is then introduced in [2], where coverage is achieved considering the R-limited Delaunay graph $\mathcal{G}_{LD}(\mathcal{P}, R)$, where two robots are neighbors if their respective Voronoi cells are neighboring, *and* if their distance is smaller than the communication range R. It is worth noting, however, that this graph is build upon the exact Voronoi partitioning $\mathcal{V}(\mathcal{P})$, whose computation requires, as already discussed, full knowledge of the region Q and at least of the Delaunay neighbors. In the next section we will propose an alternative partitioning approach, that overcomes these issues, requiring only local knowledge of the portion of the region Q and neighbors that can be directly observed by each robot.

V. Limited Voronoi

In this section, we introduce a novel methodology to obtain a partitioning of the environment that, differently from [1], [2], considers the limited sensing range of the agents as a constraint, and thus exploits only the information directly detectable by the robots: we will hereafter refer to this concept as the *limited* Voronoi partitioning.

To achieve this, for each robot with sensing range R, and position p_i we consider:

1) the set of neighbors $\mathcal{N}_{\mathcal{G}_{disk}(R)}(p_i, \mathcal{P})$ on the R-disk graph,

2) the set of points $P_i = \{p_i \cup \mathcal{N}_{\mathcal{G}_{disk}(R)}(p_i, \mathcal{P})\}$,
3) a ball with radius equal to half of the sensing range, namely

$$\overline{B}(p_i, r), \quad \text{with} \quad r = \frac{R}{2} \tag{8}$$

The *limited* Voronoi partitioning generated by $\mathcal{P}$ is then defined as $\mathcal{V}^r(\mathcal{P}) = \{V_1^r(P_1), \dots, V_n^r(P_n)\}$, where:

$$V_i^r(P_i) = \tag{9}$$
$$= \{q \in \overline{B}_{\cap Q}(p_i, r) \mid \|q - p_i\| \leq \|q - p_j\|, \forall p_j \in \mathcal{N}_{\mathcal{G}_{disk}(R)}(p_i, \mathcal{P})\},$$

where $\overline{B}_{\cap Q}(p_i, r) = \{Q \cap \overline{B}(p_i, r)\}$ is the intersection between the environment and the ball of radius r for robot i, as introduced in (8). Fig. 1 reports the difference between the exact Voronoi partitioning of Q, performed with the knowledge of the agents' positions, and the limited Voronoi partitioning introduces in this paper, which is computed with the information directly sensed by the robot.

We can then introduce the *r-limited Voronoi graph* $\mathcal{P} \mapsto \mathcal{G}_{LV}(\mathcal{P}, r) = (\mathcal{P}, \mathcal{E}_{\mathcal{G}_{LV}}(\mathcal{P}, r))$, where edges are defined as:

$$\mathcal{E}_{\mathcal{G}_{LV}}(\mathcal{P}, r) =$$
$$\{(p_i, p_j) \in \mathcal{P} \times \mathcal{P} \backslash \text{diag}(\mathcal{P} \times \mathcal{P}) \mid V_i^r(P_i) \cap V_j^r(P_j) \neq \emptyset\}.$$

We will hereafter show that the introduction of the balls of radius r in (8) is instrumental for the definition of a valid partitioning of the environment, that can be computed based only on locally available information, i.e., exchanging information over the graph $\mathcal{G}_{disk}(R)$.

Lemma 1 *The limited Voronoi partitioning defined in* (9) *is equivalent to the exact Voronoi partitioning limited to the portion of the environment Q that is covered by the union of the areas within the robots' balls of radius r introduced in* (8)*, i.e.,* $\cup_{i=1}^n \overline{B}_{\cap Q}(p_i, r)$

Proof: The R-limited Delaunay graph $\mathcal{G}_{LD}(R)$ is the proximity graph corresponding to the exact Voronoi partitioning limited to the robots' balls of radius r, namely

$$V_i(P) \cap \overline{B}(p_i, r)$$

According to [4, Prop. 2.9], $\mathcal{G}_{LD}(R)$ is spatially distributed over the graph $\mathcal{G}_{disk}(R)$. This implies that the R-disk graph encodes sufficient information to compute the R-limited Delaunay graph $\mathcal{G}_{LD}(R)$. Hence, it is possible to conclude that

$$V_i^r(P_i) = V_i(P) \cap \overline{B}(p_i, r). \tag{10}$$

which proves the statement. ■

Remark 1 *Given Lemma 1, $\mathcal{G}_{LV}(r)$ is spatially distributed over $\mathcal{G}_{disk}(R)$.*

It is possible to show that the proposed definition of limited Voronoi diagram is well posed: namely, for a sufficiently large sensing range, or for a sufficiently large number of robots, the limited Voronoi diagram corresponds to the exact Voronoi diagram.

Lemma 2 *Consider the definition of limited Voronoi diagram given in* (9)*. For any n, a value $\overline{r}$ exists such that, for $r \geq \overline{r}$, then $\mathcal{V}^r(P) = \mathcal{V}(P)$.*

Sketch of the Proof: This Lemma can be proven showing that it is always possible to find $\overline{r}$ such that $Q \subseteq \cup_{i=1}^n \overline{B}(p_i, r)$: in this case $\mathcal{V}^r(P) = \mathcal{V}(P)$.

Choosing $\overline{r}$ as the maximum distance between two points in Q, then $Q \subseteq \overline{B}(p_i, r)$, which implies $Q \subseteq \cup_{i=1}^n \overline{B}(p_i, r)$. ■

Lemma 3 *Consider the definition of limited Voronoi diagram given in* (9)*. For any r, a value $\overline{n}$ exists such that, for $n \geq \overline{n}$, then $\mathcal{V}^r(P) = \mathcal{V}(P)$.*

Sketch of the Proof: As for Lemma 2, it is always possible to find $\overline{n}$ such that $Q \subseteq \cup_{i=1}^n \overline{B}(p_i, r)$: in this case $\mathcal{V}^r(P) = \mathcal{V}(P)$. For a given value r, $\overline{n}$ can be found considering the minimum number of balls whose union covers the entire area Q. ■

Remark 2 *It is worth noting that considering radius r is fundamental for achieving a valid environment partitioning. In fact, if we considered* (9) *with R instead of r, such that $q \in \overline{B}_{\cap Q}(p_i, R)$, the Voronoi partitioning generated would not be a valid partitioning. This is due to the fact that $\mathcal{G}_D \cap \mathcal{G}_{disk}(R)$ is not spatially distributed over $\mathcal{G}_{disk}(R)$ [4, Sec. 2.2.1], as opposed to $\mathcal{G}_{LV}(r)$ (see Remark 1). An example is shown in Fig. 1b, where dashed lines show the invalid partitioning generated considering R, and overlapping areas are shown in purple. In fact, a valid partitioning of the environment is a collection of subsets of $\mathbb{R}^2$ that have disjoint interiors.*

Remark 3 *The proposed limited Voronoi partitioning allows us to take full advantage of the limited sensing capabilities of the robots, i.e., using all the information over $\mathcal{G}_{disk}(R)$, and to obtain a valid partitioning of the environment computing it over $\overline{B}(p_i, r)$ (see Remark 2).*

VI. PROPOSED CONTROL STRATEGY

In this section, building upon the concept of limited Voronoi partitioning introduced in Section V, we propose a control strategy to solve Problem 1.

In particular, we propose the following control law:

$$u_i = k(C_{V_i^r} - p_i), \tag{11}$$

where $k \in \mathbb{R}_{>0}$ is a proportional gain and $C_{V_i^r}$ is the centroid of the cell V_i^r defined by the r-limited Voronoi diagram of robot i. We will hereafter show that this control strategy leads to optimally deploying the limited sensing range robots within the unknown environment Q. For this purpose, we consider the environment Q to be partitioned according to an exact Voronoi partitioning $\mathcal{V}(\mathcal{P})$, as defined in (1). Hence, inspired by [2], we consider the following performance function:

$$f^r(x) = -x^2 1_{[0,r]}(x) - r^2 1_{(r,\infty)}(x), \tag{12}$$

which, as we will show below, leads to optimally deploying robots in their corresponding Voronoi cell considering the limited sensing capabilities. The performance function is intuitively chosen to be a non-increasing, piece-wise differentiable and continuous function which provides an indication of the robot sensor performance [1]. Hence, we consider the standard approach to the problem as $f(x) = -x^2$ for values within the range r, i.e., inside the ball $\overline{B}(p_i, r)$ (first term of (12)). Also, we consider a constant function otherwise, as $f(x) = -r^2$ for values outside the range r, i.e., outside the ball $\overline{B}(p_i, r)$, to have a continuous function (second term of (12)). Following the definition in (3), the optimization function can then be derived as

$$\mathcal{H}_{\mathcal{V}}^r(\mathcal{P}) = -\sum_{i=1}^{n}\int_{V_i(\mathcal{P})} \|q-p_i\|^2 1_{[0,r]}\phi(q)dq \\ - r^2\sum_{i=1}^{n}\int_{V_i(\mathcal{P})} 1_{(r,\infty)}\phi(q)dq. \quad (13)$$

Theorem 1 *Consider a multi-robot system composed of n robots with limited sensing range whose dynamics evolves according to* (2)*. Then, the control law* (11) *leads to maximize the optimization function* (13)*.*

Proof: We will show that control law (11) implements a gradient ascent of the optimization function (13). Namely, we will show that the time derivative of the optimization function, that is

$$\dot{\mathcal{H}}_{\mathcal{V}}^r(\mathcal{P}) = \sum_{i=1}^{n} \frac{\partial \mathcal{H}_{\mathcal{V}}^r(\mathcal{P})}{\partial p_i}^T \dot{p}_i, \quad (14)$$

is non-negative.

For this purpose, consider the definition of the optimization function in (13). Following the same computation as in (4), the first term can be rewritten by means of the polar moment of inertia. Furthermore, the second term represents the area, weighted with function ϕ, of the portion of Q that is not covered by the union of the balls $\overline{B}(p_i, r)$ (in *green* in Fig. 1b). Hence, we can rewrite (13) as:

$$\mathcal{H}_{\mathcal{V}}^r(\mathcal{P}) = -\sum_{i=1}^{n} J_{V_i(P)\cap\overline{B}(p_i,r),p_i} \\ - r^2 Area_{\phi}(Q\setminus \cup_{i=1}^{n} B(p_i,r)), \quad (15)$$

with $Area_{\phi}(S) = \int_S \phi(q)dq$, $S \subset Q$.

Given [2, Thm. 2.2], considering the choice of the performance function (12), we can compute the partial derivatives of the optimization function $\mathcal{H}_{\mathcal{V}}^r(\mathcal{P})$ as follows:

$$\frac{\partial \mathcal{H}_{\mathcal{V}}^r(\mathcal{P})}{\partial p_i} = 2\int_{V_i(P)\cap\overline{B}(p_i,r)} (q-p_i)\phi(q)dq \\ = 2M_{V_i\cap\overline{B}(p_i,r)}(C_{V_i\cap\overline{B}(p_i,r)} - p_i), \quad (16)$$

where $M_{V_i\cap\overline{B}(p_i,r)}$ and $C_{V_i\cap\overline{B}(p_i,r)}$ are respectively the mass and the centroid of the cell given by the intersection of $\overline{B}(p_i, r)$ with the exact Voronoi partitioning of the environment.

Considering the control law (11) and the dynamics (2), the time derivative of the optimization function (14) can be written as follows:

$$\dot{\mathcal{H}}_{\mathcal{V}}^r(\mathcal{P}) = \sum_{i=1}^{n}\left[2M_{V_i\cap\overline{B}(p_i,r)}(C_{V_i\cap\overline{B}(p_i,r)} - p_i)\right]^T k(C_{V_i^r} - p_i), \quad (17)$$

Defining $\kappa_i = k\,2M_{V_i\cap\overline{B}(p_i,r)} > 0$, we can rewrite (17) as follows:

$$\dot{\mathcal{H}}_{\mathcal{V}}^r(\mathcal{P}) = \sum_{i=1}^{n}\kappa_i(C_{V_i\cap\overline{B}(p_i,r)} - p_i)^T\,(C_{V_i^r} - p_i). \quad (18)$$

Consider now the results of Lemma 1, we obtain the following:

$$C_{V_i^r} = C_{V_i\cap\overline{B}(p_i,r)}, \quad \forall i = 1,\ldots,n. \quad (19)$$

Therefore, from (18) and (19), we can state that

$$\dot{\mathcal{H}}_{\mathcal{V}}^r(\mathcal{P}) = \sum_{i=1}^{n}\kappa_i\left\|C_{V_i\cap\overline{B}(p_i,r)} - p_i\right\|^2 \geq 0, \quad (20)$$

which proves the statement. ■

According to Theorem 1, we can conclude that the proposed control law (11), which can be computed by each robot based only on directly sensed information, leads to optimally covering the environment.

During the execution of the control algorithm, every robot continuously updates the limited Voronoi diagram and moves towards the centroid of the respective Voronoi cell. The centroid computation and, hence, the control law (11), depend on the neighbors detected and the environment sensed. In other words, given a constant $\phi(q)$, every robot will tend to move far from other robots and from the environment boundary, while maximizing the coverage of the area.

VII. Experimental Validation

In this section we report the simulations and the real experiments we carried out to verify the proposed algorithm.

A. Simulations

The simulations were carried out in MATLAB® and aimed at investigating the effectiveness of the proposed method, also in comparison to the standard method introduced in Section IV. When the density function ϕ is constant, the objective of the robots is to spread uniformly in the environment. To understand the difference between the proposed method and the standard one, we tested both in the same environment (Q is a square $6\text{m} \times 6\text{m}$).

In order to evaluate the performance of the system with the proposed controller from a global point of view, Fig. 2 reports the value of the standard optimization function $\mathcal{H}_{\mathcal{V}}(\mathcal{P})$, as introduced in (4), considering four robots controlled with (11), varying the sensing range, and with the standard method (7). We use the standard optimization function (4) as a benchmark for both our method and the standard one to obtain comparable measures. We take as reference the performance of the standard method (in *blue* in Fig. 2), which is not affected by the sensing range of the robots. Intuitively, robots with a small sensing range are not able to optimally cover a large environment, when compared to an ideal condition of unlimited sensing. Thus, with the proposed control strategy, the robots converge to a sub-optimal solution, which represents a tradeoff achieved considering limited sensing capabilities. Thus, the coverage performance with small sensing range robots (e.g., $R = 1\text{m}$) controlled

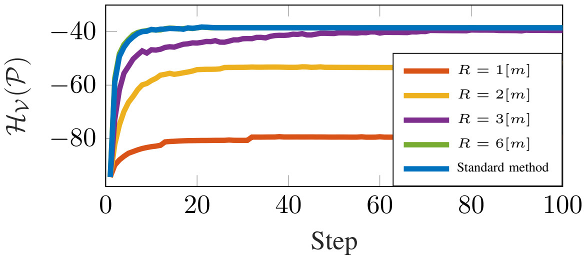


Fig. 2. Optimization function $\mathcal{H}_{\mathcal{V}}(\mathcal{P})$, as introduced in (4), achieved by a 4 robots controlled with the proposed control law (11), considering increasing sensing range, and compared with the standard method. The proposed method converges to the standard one for a sufficiently high sensing range.

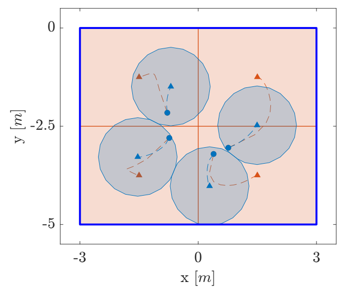


Fig. 3. Difference between Voronoi partitioning with the standard method (in *red*) and the proposed method with $R = 2\text{m}$ (in *blue*). Initial and final positions are represented with *circles* and *triangles*, respectively.

by our methodology is worse with respect to the standard one. However, when the sensing range is sufficiently big (e.g., $R = 3\text{m}$) the performance is comparable with the one obtained with the standard method and, with $R = 6\text{m}$, the performance obtained is the same as with the standard method. To visually show the difference, at steady-state, when the sensing range is small, we report, in Fig. 3, the initial (circles) and final positions of the robots (triangles), and the Voronoi partitions for both our case (in *blue*) and the standard method (in *red*). With our method it is clear how, with a sufficiently large environment, the robots converge to positions in which the neighbors $\mathcal{N}_{\mathcal{G}_{disk}}(p_i, \mathcal{P})$ are on the boundary of $\overline{B}(p_i, R)$, and the balls $\overline{B}(p_i, r)$ are tangent to each others, $\forall i$ (see Fig. 3).

Fig. 4 reports, again, the optimization function $\mathcal{H}_{\mathcal{V}}(\mathcal{P})$ defined in (4) as a benchmark for different numbers of robots controlled with the standard method (*solid* lines) and with our method with sensing range $R = 2\text{m}$ (*dashed* lines). If a group of robots with small sensing range is not able to optimally cover a large environment, increasing the number of robots, intuitively, will improve the optimization function consequently. Thus, in the proposed control strategy the low number of robots converge to a sub-optimal solution of the coverage problem when compared to the standard one, due to the limited sensing capabilities. In fact, with 5 robots (in *blue* in Fig. 4) our method does not perform as the standard one. Instead, increasing the number of robots the performance is comparable. The only difference is the transient, where the convergence is reached slowly with our method as the

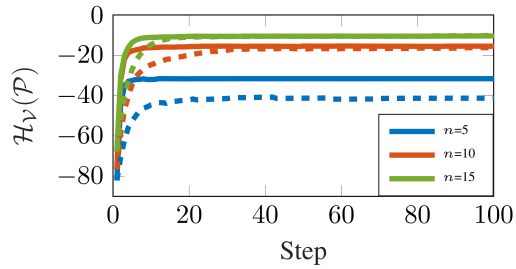


Fig. 4. Optimization function $\mathcal{H}_{\mathcal{V}}(\mathcal{P})$, as introduced in (4), achieved by a different number of robots controlled with the proposed control law (11). The *solid* lines represent the standard method, instead the *dashed* lines represent our method.

centroids are closer to the robots' position. In our method the centroid is in $\overline{B}(p_i, r)$, while in the standard method it is inside the Voronoi cell: hence, if the cell is big, the centroid can be further away from the position of the robot. It is worth noting that the steady-state value is higher with a higher number of robots given the chosen optimization function. In fact, if the number of robots is larger, then the area is better allocated to the robots, namely each point $q \in Q$ is closer to a robot than in the case with fewer robots.

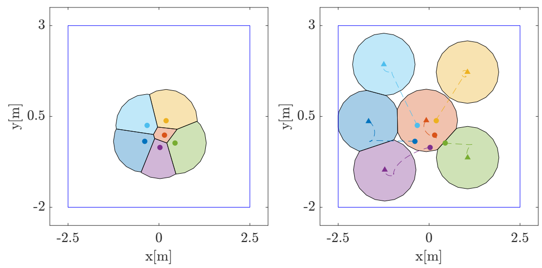
It is worth noting that these results corroborate what was demonstrated in Lemma 2 and Lemma 3, respectively. In fact, increasing the sensing range or the number of robots we obtain the same value of the optimization function because the underlying limited Voronoi diagram is equal to the exact one.

Moreover, we tested the proposed approach in a more complex environment, as shown in the accompanying video. The simulation, implemented in Matlab® , shows the motion of 6 robots deployed in a non-convex environment. Each robot (indicated with a colored dot) is directed towards the centroid (colored cross) of the respective limited Voronoi cell within the sensing range R (dotted colored circle). We remind that the standard coverage control is not suitable for non-convex environments because of the Voronoi partitioning and it needs to be combined with further control strategies [21], [22]. We remark that the simulation has the aim to show an experiment in a more complex situation and that the control strategy proposed seems to be able to deal with non-convex environments. However, further studies need to be conducted on this field.

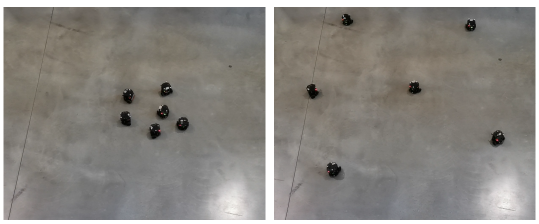
B. Implementation on the Robots

Some preliminary tests were carried out on real robotic platforms. The hardware set-up was made up of a group of TurtleBot 3 Burger robots, equipped with a Raspberry Pi 4 mounting Ubuntu 20.04, and of an OptiTrack tracking system. The proposed algorithm was implemented in C++ and, for the calculation of the limited Voronoi, we adapted a library[2] developed with the Fortune Algorithm [23]. The interface with the robots was implemented in ROS2, which managed also the communication with the OptiTrack system.

[2]https://github.com/pvigier/FortuneAlgorithm



(a) Initial positions and limited Voronoi cells. (b) Final positions and limited Voronoi cells.



(c) Initial position of the real robots. (d) Final position of the real robots.

Fig. 5. Experiments with uniform density function. In 5b and 5a every colored shape is the limited Voronoi cell of the respective robot. In 5b the *circles* are the starting positions, the *triangles* are the final positions.

It is worth noting that the proposed method does not need any communication between the robots and that the external tracking system can be replaced by on board sensors. In fact, the OptiTrack system was used to emulate a limited range sensor that is able only to measure neighbors' position. This information was used to create the limited Voronoi diagram with the C++ library, which calculated also the centroid of the cell based on the density function. Hence, each robot was able to calculate on-board the limited Voronoi cell V_i^r, and the centroid $C_{V_i^r}$. To obtain a faster implementation we used the line integration method proposed in [24]. The input calculated in terms of $\dot{p}_i$ from (11) was modified according to the input-output state feedback linearization [25] to be correctly applied to the robot. The accompanying video reports a representative trial of the experiments.

In the first experiment, the density function $\phi(q)$ was constant, hence, the aim of the robots was to spread into the environment. Fig. 5 reports the successful result of the experiment: initial and final positions of the robots are reported together with their respective limited Voronoi cell.

The second experiment aimed at testing the algorithm with a non uniform density function. We assumed $\phi(q)$ had the form of a bivariate unimodal Gaussian distribution with $\mu = \begin{bmatrix} 0 & 0 \end{bmatrix}$ and $\Sigma \in \mathbb{R}^{2\times2}$, $\Sigma = \begin{bmatrix} \sigma & 0 \\ 0 & \sigma \end{bmatrix}$, where $\sigma = 0.3$. μ and Σ are respectively the mean and the covariance matrix of the Gaussian distribution. Fig. 6 reports the trajectories of the robots from the initial positions (circles) to the final positions (triangles). As expected, the robots converged to the center and they rounded up proportionally to the variance. The Gaussian distribution is represented with the *black* cross, which is the mean value, and the *black* concentric circles,

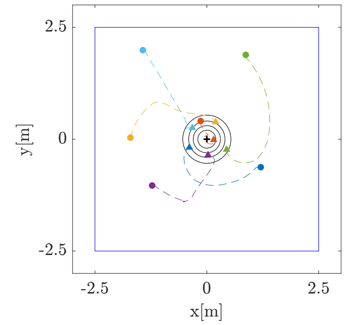


Fig. 6. Experiments with not uniform density function $\phi(q)$. Initial and final positions with *circles* and *triangles*, respectively. The *black* cross is the center of $\phi(q)$ and the *black* concentric circles are different values of variance.

which represent equidistant levels of the variance.

VIII. Conclusion and Future Works

In this paper, we presented a distributed coverage based control to coordinate a group of autonomous robots with limited sensing capabilities in an unknown environment. The proposed novel control strategy, based on a limited Voronoi partitioning, uses only the information directly detected by the robot. We formally show how the control law drives the group of robots to the respective centroid of the limited Voronoi cell. A series of experiments and simulations were performed on a group of mobile robots to validate the performance of the proposed control method. As a future work, we aim to further investigate how the proposed strategy can deal with non-convex environments, since promising results have been obtained in a few preliminary simulations. Moreover, we aim to extend the proposed control strategy to be effective for the exploration of unknown complex environments. Finally, we aim to improve the control method in order to consider a group of robots characterized by heterogeneous limited sensing capabilities.

References

[1] J. Cortes, S. Martinez, 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.

[2] J. Cortes, S. Martinez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 11, no. 4, pp. 691–719, 2005.

[3] Q. Du, M. Emelianenko, and L. Ju, "Convergence of the lloyd algorithm for computing centroidal voronoi tessellations," *SIAM journal on numerical analysis*, vol. 44, no. 1, pp. 102–119, 2006.

[4] F. Bullo, J. Cortés, and S. Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009.

[5] E. Teruel, R. Aragues, and G. López-Nicolás, "A practical method to cover evenly a dynamic region with a swarm," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1359–1366, 2021.

[6] M. Schwager, J. McLurkin, and D. Rus, "Distributed coverage control with sensory feedback for networked robots." in *robotics: science and systems*, 2006, pp. 49–56.

[7] K. Guruprasad and P. Dasgupta, "Distributed voronoi partitioning for multi-robot systems with limited range sensors," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 3546–3552.

[8] B. A. Bash and P. J. Desnoyers, "Exact distributed voronoi cell computation in sensor networks," in *Proceedings of the 6th international conference on Information processing in sensor networks*, 2007, pp. 236–243.

[9] W. Alsalih, K. Islam, Y. N. Rodríguez, and H. Xiao, "Distributed voronoi diagram computation in wireless sensor networks." in *SPAA*, 2008, p. 364.

[10] Q. Li and D. Rus, "Navigation protocols in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 1, no. 1, pp. 3–35, 2005.

[11] M. Cao and C. Hadjicostis, "Distributed algorithms for voronoi diagrams and application in ad-hoc networks," *UIUC Coordinated Science Laboratory, Tech. Rep. UILU-ENG-03-2222, DC-210*, 2003.

[12] S. Kemna, J. G. Rogers, C. Nieto-Granda, S. Young, and G. S. Sukhatme, "Multi-robot coordination through dynamic voronoi partitioning for informative adaptive sampling in communication-constrained environments," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2124–2130.

[13] C. He, Z. Feng, and Z. Ren, "Distributed algorithm for voronoi partition of wireless sensor networks with a limited sensing range," *Sensors*, vol. 18, no. 2, p. 446, 2018.

[14] M. Santos, Y. Diaz-Mercado, and M. Egerstedt, "Coverage control for multirobot teams with heterogeneous sensing capabilities," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 919–925, 2018.

[15] F. Farzadpour, X. Zhang, X. Chen, and T. Zhang, "On performance measurement for a heterogeneous planar field sensor network," in *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2017, pp. 166–171.

[16] M. Rudolph, S. Wilson, and M. Egerstedt, "Range limited coverage control using air-ground multi-robot teams," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[17] K. Laventall and J. Cortés, "Coverage control by multi-robot networks with limited-range anisotropic sensory," *International Journal of Control*, vol. 82, no. 6, pp. 1113–1121, 2009.

[18] A. Okabe, "Spatial tessellations," *International Encyclopedia of Geography: People, the Earth, Environment and Technology: People, the Earth, Environment and Technology*, pp. 1–11, 2016.

[19] R. Soukieh, I. Shames, and B. Fidan, "Obstacle avoidance of nonholonomic unicycle robots based on fluid mechanical modeling," in *Proceedings of the European Control Conference*, 2009.

[20] D. Lee, A. Franchi, H. Son, C. Ha, H. Bulthoff, and P. Robuffo Giordano, "Semiautonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 4, pp. 1334–1345, Aug 2013.

[21] S. Bhattacharya, N. Michael, and V. Kumar, "Distributed coverage and exploration in unknown non-convex environments," in *Distributed autonomous robotic systems*. Springer, 2013, pp. 61–75.

[22] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, and D. Rus, "Voronoi coverage of non-convex environments with a group of networked robots," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 4982–4989.

[23] S. Fortune, "A sweepline algorithm for voronoi diagrams," *Algorithmica*, vol. 2, no. 1, pp. 153–174, 1987.

[24] N. Hayashi, K. Segawa, and S. Takai, "2d voronoi coverage control with gaussian density functions by line integration," *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 2, pp. 110–116, 2017.

[25] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.