# Distributed Coverage Estimation and Control for Multirobot Persistent Tasks

José Manuel Palacios-Gasós, *Student Member, IEEE*, Eduardo Montijano, *Member, IEEE*, Carlos Sagüés, *Senior Member, IEEE*, and Sergio Llorente

***Abstract*—In this paper, we address the problem of persistently covering an environment with a group of mobile robots. In contrast to traditional coverage, in our scenario the coverage level of the environment is always changing. For this reason, the robots have to continually move to maintain a desired coverage level. In this context, our contribution is a complete approach to the problem, including distributed estimation of the coverage and control of the motion of the robots. First, we present an algorithm that allows every robot to estimate the global coverage function only with local information. We pay special attention to the characterization of the algorithm, establishing bounds on the estimation error, and we demonstrate that the algorithm guarantees a perfect estimation in particular areas. Second, we introduce a new function to determine the possible improvement of the coverage at each point of the environment. Upon this metric, we build a motion control strategy that drives the robots to the points of the highest improvement while following the direction of the gradient of the function. Finally, we simulate the proposal to test its correctness and performance.**

***Index Terms*—Distributed systems, multi-robot systems, persistent coverage.**

## I. Introduction

RECENT advances in mobile robotics and an increasing development of affordable autonomous robots have motivated an extensive research in multirobot systems. A particularly interesting problem is the coverage problem, that aims to cover a given environment with a team of robotic agents. This problem is of special interest in many applications such as vacuuming [1], cleaning a place where dust is continuously settling [2], lawn mowing [3], or environmental monitoring [4], [5]. More recently, the apparition of useful unmanned aerial vehicles (UAVs) has encouraged the application of the coverage problem to surveillance and monitoring [6].

The approaches to the coverage problem can be classified into three different categories, *static*, *dynamic,* and *persistent coverage*. Nonetheless, they are closely related. In many cases, the same problem can be stated as the three of them depending on the objective, solutions can be extended from one approach to the others with slight adaptations, and most of the tools are useful for all of them. In the first place, the *static coverage* or *deployment problem* seeks to determine the optimal positions of a group of robots to cover an environment. This approach [7], [8] is based on the locational-optimization theory, whose objective is to allocate resources in an area according to a fixed criterion, e.g., mail boxes in a city [9]. The second approach is *dynamic coverage*, in which the robots must cover the entire environment at least once or until a desired coverage level is reached to consider the task finished. Typically, the goal of this approach is to minimize the time and energy consumption needed to complete the task [10], [11].

The last approach is the *persistent coverage problem*, in which this paper focuses. The aim of persistent coverage is to maintain a desired coverage level over the environment. Persistent coverage differs from static and dynamic coverage in that the coverage of the environment persistently decays and the robots have to continually move to maintain the desired level, i.e., it requires repetition and redundant actions. Therefore, in this case, the task can generally never be completed. Note that static and dynamic coverage may need to be repeated or maintained if the environment changes and, thus, become persistent static coverage and persistent coverage, respectively.

From a practical point of view, the coverage level can be seen as a physical quantity such as the temperature while heating a place or the level of water while watering crops. These magnitudes deteriorate over time due to physical phenomena such as cooling or evaporation. It can also be seen as the quality of a measurement and, in this sense, persistent coverage is often addressed as *persistent surveillance* or *environmental monitoring*, especially with UAVs [6]. The problem is also strictly related to the *patrolling* problem [12]–[14], although patrolling may require awareness of the behavior of adversaries as in [15].

The solutions to the persistent coverage problem intend to derive results which are applicable for infinite time. Hokayem *et al.* [16] have the objective of revisiting every point periodically. However, they do not consider explicitly a decay of the coverage. The approach from [17] considers a decay, called awareness, and only requires full awareness in some points while revisiting all points periodically. An optimal control solution is presented in [18] to minimize the uncertainty in a one-dimensional field. This uncertainty increases constantly with time and is reduced over the range of the sensors. An optimal solution is also found for the discretized problem in [19] using a Branch-and-Bound algorithm. In [20], the objective of a team of UAVs is to minimize the maximum time between visits to all cells of the environment. This type of solutions are also derived for the patrolling problem. In [14], a Voronoi partition is used to

Manuscript received December 23, 2015; revised May 15, 2016; accepted July 15, 2016. Date of publication October 20, 2016; date of current version December 2, 2016. This paper was recommended for publication by Associate Editor P. Robuffo Giordano and Editor T. Murphey upon evaluation of the reviewers' comments. This work was supported in part by the projects RTC-2014-1847-6 of Retos-Colaboración and DPI2015-69376-R (MINECO/FEDER), CUD2013-05 and in part by a Gobierno de Aragón Scholarship C076/2014, partially funded by European Social Fund.

J. M. Palacios-Gasós and C. Sagüés are with Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Zaragoza 50018, Spain (e-mail: jmpala@unizar.es; csagues@unizar.es).

E. Montijano is with Centro Universitario de la Defensa and Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Zaragoza 50018, Spain (e-mail: emonti@unizar.es).

S. Llorente is with the Department of Research and Development, Induction Technology, Product Division Cookers, BSH Home Appliances Group, Zaragoza 50016, Spain (e-mail: sergio.llorente@bshg.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2016.2602383

partition the environment and each agent follows goals to cover its own region.

Other solutions are focused on computing the trajectories which robots follow [12], [13]. These paths can be used to compute a speed controller to keep the coverage level within desired limits [21]. With the objective of finding optimal closed paths, Rapidly-exploring Random Trees are used in [22] to sense a dynamic Gaussian random field of the environment that evolves over time. Soltero *et al.* in [23] develop an online adaptive path planning algorithm based on gradient descent of a Voronoi-based cost function that simultaneously allows the robots to explore the regions of interest in an unknown environment and to calculate their paths. In [24], Lin *et al.* address the problem of their previous work [18] in a two-dimensional space and formulate a parametric optimization to determine a set of elliptical trajectories for the robots. Alamdari *et al.* [25] plan paths inside the graph of regions of interest to minimize the time between visits.

Since planning optimal trajectories usually requires the solution of an intractable dynamic program (DP), other works compute suboptimal paths. In [26], a gradient descent method together with a centralized assignment of objectives is proposed and a variable coverage action is introduced. Hubel *et al.* [27] tackle a similar problem in which the coverage reaches an upper bound on the objective and the interest of the environment varies with time. In [28], they solve approximately over a finite-time horizon the DP, whose optimization criterion is the minimum uncertainty of a field estimate.

A typical assumption in the approaches to the coverage problem is the exact knowledge of the coverage levels of the whole environment that permits the calculation of the motion of the robots. One alternative to collect this information is to communicate each production or measurement of every robot through the network labeled with a robot identifier and a time stamp. Then each robot is capable of retrieving the actual coverage map using all the received information. However, this information exchange can be challenging when dealing with large environments, large and changing robotic networks or when the team of robots has limited communication capabilities. Therefore, distributed estimation strategies where the information is synthesized instead of accumulated become more appropriate for this problem. In terms of estimation, in [29], the environment is parameterized by a set of basic functions, which are estimated by using a continuous-time PI-consensus algorithm [30]. A distributed interpolation scheme is used in [31] with a Kalman-like formulation and compression of the data. Although their performance is good, these approaches do not take into account how the actions of the robots modify the coverage level of the different areas of the environment.

### A. Contributions

In this paper, we address the persistent coverage problem in which the coverage level of the environment decreases over time and the robots are capable of increasing it as in heating or watering applications. We propose a solution with a distributed robotic network that presents the following main contributions:

Our first contribution is an algorithm to estimate the coverage of the environment using only local information. The algorithm is based on distributed max-consensus combined with additive inputs given by the actions of the robots. Although we do not consider delays, noise or uncertainties in the parameters, we use the term estimation as in deterministic problems of control theory such as state observers [32]. We provide bounds on the estimation error of the robots for any point of the environment and characterize the set of allowable actions that ensure that each robot maintains a perfect estimation within a certain area, that we call reachable area. Moreover, we characterize the area in which there is no estimation error for any possible action. The main contributions and additions with respect to the preliminary version of this part of the work that appeared in [33] are the following:

1) A deeper revision of the state of the art is presented.
2) Regarding the estimation algorithm, we extend the results to the reachable area of the robots that is larger than the zero-error area.
3) In this paper, we include several proofs that do not appear in [33], specifically the proofs of Lemmas IV.1 to IV.3.
4) In the simulation part, we report more rigorous and detailed results, including the influence of the parameters.

The second contribution is a distributed strategy to control the motion of the robots that combine a goal-oriented term and a gradient term. Both terms are based on a new metric, called *Improvement Function*. This metric characterizes how profitable it is in terms of coverage to locate a robot at a specific position. Upon this metric, we build an algorithm that allows each robot to locally determine the goals that it has to follow as the points with the highest improvement value. The goal-oriented term that we propose drives the robots to these goals while following the direction of the gradient of the *Improvement Function* from the other term. This avoids revisiting well-covered areas and drives the trajectories through uncovered areas. The proposed motion control law, even though it is based on heuristics, addresses an NP-hard problem, is fully distributed, accounts for the quality of the estimation and shows good performance.

The remainder of the paper is structured as follows: Section II presents a discrete-time formulation of the persistent coverage problem. The description of the algorithm for the local map update is in Section III. The behavior of the algorithm is characterized in Section IV. Section V introduces the motion law to control the movement of the robots. The performance of the whole approach is analyzed in simulations in Section VI. Section VII gathers the conclusion of this paper. Finally, in order to facilitate the reading, the proofs of most theoretical results are in the appendices of the paper.

## II. Problem Formulation

To formulate the problem of persistently covering an environment with a team of robotic agents, we develop the formulation in discrete time due to the fact that a distributed system requires discrete communications.

Let $\mathbf{Q} \subset \mathbb{R}^2$ be a bounded environment which a team of robots has to persistently cover. The team consists of $N \in \mathbb{N}$ mobile robots. We assume that they are holonomic:

$$\mathbf{p}_i(k) = \mathbf{p}_i(k-1) + \mathbf{u}_i(k-1)$$

where $\mathbf{p}_i(k) \in \mathbf{Q}$ is the position of robot $i$ at time $k \geq 1$ with $i \in \{1, \ldots, N\}$. The maximum distance that a robot can move in one step is $u^{\max}$, i.e., $\|\mathbf{u}_i(k)\| \leq u^{\max}$.

The robots form a network defined by a communication graph $G_{\text{com}}(k) = \big(V(k), E(k)\big)$. The vertices $V(k)$ of the graph are the positions $\mathbf{p}_i(k)$ of the robots. To define the edges $E(k)$,

we let $r^{\text{com}} > 0$ be the *communication radius*, the maximum distance between two robots at which they can communicate. We assume it constant and equal for all the robots. With this radius, an edge $(i,j) \in E(k)$ if $\|\mathbf{p}_i(k) - \mathbf{p}_j(k)\| \leq r^{\text{com}}$. Additionally, $N_i(k) = \{j \in \{1,\dots,N\} \mid (i,j) \in E(k)\}$ are the *neighbors* of robot $i$ at instant $k$.

Depending on the motion of the robots, the communication graph could end up being disconnected. In this paper, we assume that this is not the case, and we assume that it remains connected at all times even though the graph topology can change over time. This could be achieved, for example, by including in the motion of the robots connectivity constraints such as [34] and [35]. Nevertheless, a more relaxed assumption such as periodic joint connectivity [36], [37] could be made, where the union of the disconnected communication graphs would become connected at most every T instants.

The communication in the network is considered synchronous and not affected by delays or noise, assuming this can be provided by a synchronizer [38]. Moreover, the presence of obstacles does not affect the communication capabilities of the network, i.e., two robots keep in communication even if there is an obstacle between them.

The coverage of the environment is modeled with a time-varying field, $Z(\mathbf{q},k)$, which we call coverage function or global map indistinctly. This field represents a physical quantity in persistent coverage applications, such as temperature while heating a building or level of water in a field or in flowerpots while watering. The aim of the robotic team is to maintain a desired coverage level, $Z^*(\mathbf{q}) > 0,\ \forall\, \mathbf{q} \in \mathbf{Q}$. We will alternatively refer to this function as an objective map.

To reach this aim, the robots are able to generate an increase on the coverage level at each time instant. Depending on their position, $\mathbf{p}_i(k)$, each robot has a coverage area, $\mathbf{\Omega}_i\big(\mathbf{p}_i(k)\big) \subseteq \mathbf{Q}$, which is bounded by a circle of radius $r_i^{\text{cov}}$ centered at $\mathbf{p}_i(k)$, although it is not necessarily circular, convex, or equal for all the robots. In this area, a robot increases the value of the coverage by $\alpha_i\big(\mathbf{q}, \mathbf{p}_i(k)\big) > 0$, which we call production. For the rest of the points of the environment, i.e., the ones that are not covered by robot $i$, we consider that $\alpha_i\big(\mathbf{q}, \mathbf{p}_i(k)\big) = 0$. From now on, we will denote $\mathbf{\Omega}_i(k) \equiv \mathbf{\Omega}_i\big(\mathbf{p}_i(k)\big)$, $\alpha_i(k) \equiv \alpha_i\big(\mathbf{q}, \mathbf{p}_i(k)\big)$ and $\alpha(k) \equiv \sum_{i\in\{1,\dots,N\}} \alpha_i(k)$.

On the other hand, as time goes by, the coverage level of a given point decreases according to a constant *decay* gain $d(\mathbf{q})$, with $0 < d(\mathbf{q}) < 1$. Thus, the coverage function at time $k$ is

$$Z(\mathbf{q},k) = d(\mathbf{q})Z(\mathbf{q},k-1) + \alpha(k). \tag{1}$$

For clarity, in the rest of the paper, we omit the dependency of $\mathbf{q}$ unless it is strictly necessary, i.e., $Z(\mathbf{q},k) \equiv Z(k)$ or $Z^*(\mathbf{q}) \equiv Z^*$. Note that the coverage level of the environment is bounded by $0 \leq Z(k) \leq \max \alpha(k)/(1-d)$. This definition of the evolution of the coverage is suitable for applications such as heating a place or watering crops. It differs from alternative formulations as in [21], where the coverage level increases over time and is decreased by the robots.

It is clear that any control policy that aims to make $Z(k)$ equal to $Z^*$ will consider the coverage level in the design of the inputs of the robots. However, in (1), we observe that the value of $Z(k)$ depends on the coverage actions of all the robots. This represents an obstacle in a distributed scenario, where a robot may not be in direct communication with all the others and may only have partial information of the environment. For this reason, a distributed approach to the persistent coverage problem requires every robot to have an estimation of the coverage function of the entire environment as well as a control policy aware of it.

**Algorithm 1:** Local Map Update.

1: – *Step 1:*
2: Calculate map-to-communicate $Z_i^{\text{com}}(k)$, (2).
3: Communicate map to neighbors.
4: Update local map $Z_i^-(k)$, (3).
5: – *Step 2:*
6: Extract overlapped production $\beta_i\big(\mathbf{q}, \mathbf{p}_i(k)\big)$, (4).
7: Communicate regions to neighbors.
8: Update local map $Z_i(k)$, (5).

## III. Local Estimation of the Coverage

In order to allow each robot to have an accurate estimate of the coverage field we introduce a distributed estimation algorithm that only uses information provided by direct neighbors in the communication graph.

*Assumption III.1:* We assume that robots with overlapping coverage areas can communicate, i.e., $r^{\text{com}} > 2r^{\text{cov}}_{\max}$, where $r^{\text{cov}}_{\max} = \max_{i\in\{1,\dots,N\}} r_i^{\text{cov}}$.

The local estimation of the coverage function of each robot is $Z_i(\mathbf{q},k)$, that we call local map. Each robot updates its local map using the information received from its current neighbors, which may include the local map and, in some cases, the coverage function of the neighboring robots. Before presenting our updating algorithm, let us introduce an assumption referring to the initial estimation.

*Assumption III.2:* We assume that at the initial time, the robots know the actual coverage level, i.e., the estimation of every robot is correct: $Z_i(0) = Z(0),\ \forall\, \mathbf{q} \in \mathbf{Q}$.

Now we present in Algorithm 1 our two-step updating strategy along with an illustrating example in Fig. 1. In this example, we have the global coverage map $Z(k)$ in Fig. 1(a) and we refer to the estimation of robot 1.

The intuitive idea of this algorithm is the following: we extract in Step 1 the additions that the neighbors have already gathered, both from their productions and from other robots, and we take into account the overlappings to correct the estimation in Step 2. In fact, the objective of this update is to keep the estimation as close to the actual value as possible without overestimating it.

In the first step, at each communication time, $k$, each robot generates its *map-to-communicate*, $Z_i^{\text{com}}(k)$ (Fig. 1(b) in example), as the map of the previous time instant with its decay plus its current production

$$Z_i^{\text{com}}(k) = dZ_i(k-1) + \alpha_i(k) \tag{2}$$

where we assume that the decay rate is known by the robots. Note that at time $k$ each robot includes the production of the same time instant $k$. Therefore, there is no delay or offset in the estimation.

Each robot sends its map-to-communicate to its neighbors and receives their maps. With this information, the first step of the update is performed dividing the map into two parts: the coverage area, $\mathbf{\Omega}_i(k)$, and the rest of the map. The robot updates

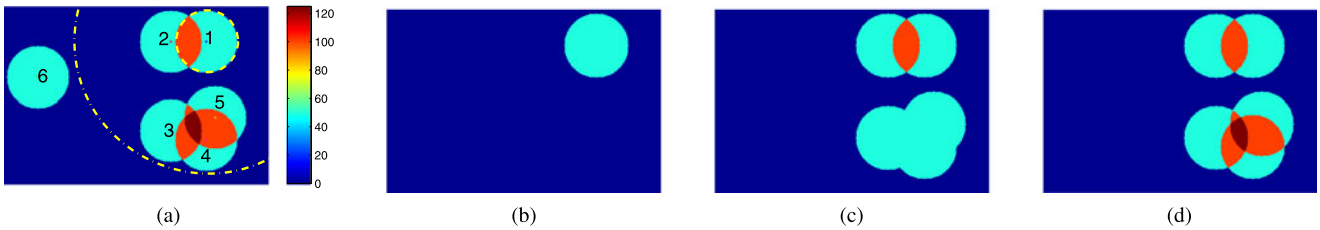



Fig. 1. Example of the estimation algorithm at time k. (a) Global coverage map, $Z(k)$. The coverage area of robot 1 is represented with a dashed yellow circumference and its communication area with a dash-dotted one. (b) Map-to-communicate of robot 1, $Z_1^{\text{com}}(k)$. (c) Map calculated by robot 1 in the first update step, $Z_1^-(k)$. (d) Estimation of the coverage map of robot 1, $Z_1(k)$. The orange areas coincide with the overlapped productions of the robots, $\beta_i(k)$.

each region separately according to

$$Z_i^-(k) = Z_i^{\text{com}}(k) + \sum_{j\in N_i(k)} \max\big(Z_j^{\text{com}}(k) - dZ_i(k-1), 0\big), \quad \forall \mathbf{q}\in\boldsymbol{\Omega}_i(k) \quad (3a)$$

$$Z_i^-(k) = \max_{j\in N_i(k)} \big(dZ_i(k-1), Z_j^{\text{com}}(k)\big), \quad \forall \mathbf{q}\notin\boldsymbol{\Omega}_i(k). \quad (3b)$$

For the update (3a) robot $i$ adds its map-to-communicate (2) to the contributions of its neighbors. The contribution of a neighbor $j$ can be calculated as the difference $Z_j^{\text{com}}(k) - dZ_i(k-1)$, as the orange area of Fig. 1(c). To prevent this difference from being negative, we use the maximum function. This may happen when the estimation of robot $j$ is smaller than the estimation of robot $i$ as in the following situation. Let robot $i$ be overlapped with a robot $\ell$ that is not a neighbor of robot $j$, i.e., $j, \ell \in N_i(k)$ and $j \notin N_\ell(k) \Leftrightarrow \ell \notin N_j(k)$. The estimation $Z_j^{\text{com}}(k)$ may be lower than $dZ_i(k-1)$ in the overlapping region $\boldsymbol{\Omega}_i(k) \cap \boldsymbol{\Omega}_\ell(k)$ since the production of robot $\ell$ at time $k$ is not available to robot $j$. This would happen to the estimation of robot 2 in Fig. 1(a) at the next iteration. It would be overlapped with robot 1 and in direct communication with robot 6, while 1 and 6 would not be neighbors.

In the second part of the local map update (3b), each robot updates its local map outside its coverage area, $\mathbf{q}\notin\boldsymbol{\Omega}_i(k)$, as the maximum of the received values and its own. Such value corresponds to the latest information of this robot or its neighbors. This update underestimates the coverage level when two or more neighbors are overlapping outside the coverage area of a robot as in Fig. 1(c). In that situation only the highest contribution is considered and not the addition of all of them.

To counteract this error, a second updating step is executed. At first, each robot extracts the region of its coverage area that is overlapped with the one of another robot, and then sends its coverage function in this region to its neighbors

$$\beta_i(k) \equiv \beta_i\big(\mathbf{q}, \mathbf{p}_i(k)\big) = \alpha_i\big(\mathbf{q}, \mathbf{p}_i(k)\big), \quad \forall \mathbf{q}\in\boldsymbol{\Omega}_i^o(k) \quad (4)$$

where $\boldsymbol{\Omega}_i^o(k) = \big\{\mathbf{q}\in\boldsymbol{\Omega}_i(k)\cap\boldsymbol{\Omega}_j(k) \,|\, j\in N_i(k)\big\}$ is the overlapped area of robot $i$ with its neighbors. Robot $i$ determines its own overlapped area as $\boldsymbol{\Omega}_i(k)\cap\boldsymbol{\Omega}_j(k) = \big\{\mathbf{q}\in\boldsymbol{\Omega}_i(k) \,|\, Z_j^{\text{com}}(k) - dZ_i(k-1) > 0\big\}$, $\forall j\in N_i(k)$.

The robots exchange the overlapped productions with their neighbors and, with the received ones, they perform the final update

$$Z_i(k) = \begin{cases} Z_i^-(k), & \forall \mathbf{q}\in\boldsymbol{\Omega}_i(k) \\ Z_i^-(k) - \max\limits_{j\in N_i(k)} \beta_j(k) + \sum\limits_{j\in N_i(k)} \beta_j(k), & \forall \mathbf{q}\notin\boldsymbol{\Omega}_i(k). \end{cases} \quad (5)$$

This final step adds the contributions that are not considered in the first and ends the estimation, as shown in Fig. 1(d).

Note that our local map updating strategy from Algorithm 1 satisfies the following property:

$$Z_i(k) \geq dZ_i(k-1), \quad \forall \mathbf{q}\in\mathbf{Q}. \quad (6)$$

Also note that the presence of obstacles or a nonconvex environment would not affect the estimation algorithm according to the communication policy.

Our algorithm permits that each robot locally decide which points of its own production are of interest and discards the rest in a communication-effective solution. Nevertheless, slightly different rules could be applied to estimate the coverage but, in that case, different information should be exchanged. For instance, another estimation strategy could exchange the coverage $\alpha_i$ and forward the received $\alpha_j$. However, this alternative would require a trace of which productions have been included and forwarded, several communication rounds at each iteration and an increased communication expense.

## IV. Characterization of the Estimation

We characterize now the accuracy of the estimation using our algorithm. In the first place, we determine the allowable actions of each robot in order to guarantee that the estimation is equal to the global map inside what we call the zero-error area and the reachable area of the robots. Next, we delimit the areas in which a correct estimation is guaranteed regardless of the movement of the robots. Finally, we establish bounds to the estimation error in all the points of the environment.

Let us begin defining the zero-error area of robot $i$, $\boldsymbol{\Omega}_i^z(k) = \big\{\mathbf{q}\in\mathbf{Q} \,|\, \|\mathbf{q}-\mathbf{p}_i(k)\| < r^{\text{com}} - r_{\max}^{\text{cov}}\big\}$, that contains the points in which our algorithm guarantees a correct estimation. These are the points of the environment that can be covered by another robot only if it is a neighbor of $i$ at each time. In Fig. 2, an illustrative example of this area is shown.

We also define the reachable area of robot $i$, $\boldsymbol{\Omega}_i^r(k)$, that comprises the points of the environment that can be reached by the zero-error area with the action calculated at time $k$, $\boldsymbol{\Omega}_i^r(k) = \big\{\mathbf{q}\in\mathbf{Q} \,|\, \|\mathbf{q}-\mathbf{p}_i(k)\| < r^{\text{com}} - r_{\max}^{\text{cov}} + u_i^{\max}(k)\big\}$, where $u_i^{\max}(k)$ is the maximum action of robot $i$.

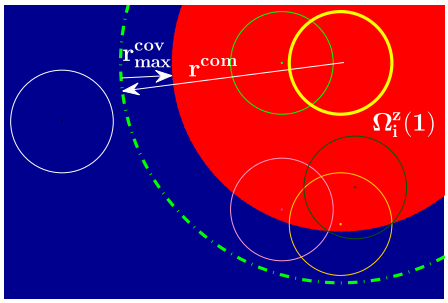



Fig. 2. Example of zero-error area (red) for the robot depicted in yellow.

Let us introduce now some useful results in the following lemmas to demonstrate the correctness of the estimation in these points in Theorem IV.4. First, we prove that a correct estimation at a time $k-1$ in $\mathbf{\Omega}_i^r(k-1)$ leads to a bounded estimation in all the environment at time $k$ and also to a correct estimation in $\mathbf{\Omega}_i^z(k)$ so that the update of the algorithm does not introduce an error in the current zero-error area.

*Lemma IV.1:* For any iteration $k$, consider that at the previous iteration the estimation of the coverage map of a robot $i$ is equal to the global coverage map inside its reachable area

$$Z_i(k-1) = Z(k-1) \quad \forall \mathbf{q} \in \mathbf{\Omega}_i^r(k-1). \tag{7}$$

Then the execution of Algorithm 1 leads to a bounded estimation in all the environment, $Z_i(k) \leq Z(k) \quad \forall \mathbf{q} \in \mathbf{Q}$, and does not introduce an error in the estimation inside the zero-error area, $Z_i(k) = Z(k) \quad \forall \mathbf{q} \in \mathbf{\Omega}_i^z(k)$.

*Proof:* See Appendix A. ■

Next, we prove that the estimation at a particular time takes into account all the productions from at least $N-1$ instants before. To do so, we denote the diameter of the communication graph as $\operatorname{diam}\left(G_{\text{com}}(k)\right)$ that is bounded by $1 \leq \operatorname{diam}\left(G_{\text{com}}(k)\right) \leq N-1$. Thus, the maximum time that the production of a robot needs to propagate through the entire network is equal to the number of robots minus one.

*Lemma IV.2:* Consider that for any iteration $k-N$, the estimation of the coverage map of all robots is equal to the global coverage map inside their reachable area, $Z_i(k-N) = Z(k-N) \quad \forall \mathbf{q} \in \mathbf{\Omega}_i^r(k-N)$. Then, according to Algorithm 1, the production of the robots developed at time instant $k-N+1$, i.e., $\alpha(k-N+1)$, does not induce an error in the estimation of the coverage at time $k$, i.e., in $Z_i(k), i \in \{1, \ldots, N\}$.

*Proof:* See Appendix B. ■

The third result establishes bounds in the motion of each robot in order to assure that its zero-error area does not reach the production of other robots which were not its neighbors at the previous time.

*Lemma IV.3:* Let $\mathbf{\Omega}_i^\rho(k)$ be a circle of center $\mathbf{p}_i(k)$ and radius $\rho_i(k) - r^{\text{cov}}_{\max}$. $\rho_i(k) = \min_{j,\ell} \|\mathbf{p}_i(k) - \mathbf{p}_j(\ell)\|$, with $j \notin N_i(k)$ and $\ell > 0, \ell \in \{k-N+1, \ldots, k\}$, is the minimum distance from the current position of robot $i$ to any of the last $N-1$ positions of other robots $j$ that are not neighbors of $i$ at iteration $k$. If the motion of robot $i$ at time $k$ satisfies that

$$\|\mathbf{u}_i(k)\| \leq u_i^{\max}(k) = \min\left(u^{\max}, \rho_i(k) - r^{\text{com}}\right) \tag{8}$$

then $\alpha_j(\ell) = 0 \quad \forall \mathbf{q} \in \mathbf{\Omega}_i^r(k)$, with $j \notin N_i(k)$ and $\ell > 0, \ell \in \{k-N+1, \ldots, k\}$.

*Proof:* See Appendix C. ■

In practice, the restriction that the previous lemma establishes on $\|\mathbf{u}_i(k)\|$ may be stronger than needed, since the direction of the movement may direct the robot to a place with no possible error, while the module of the action restricts such movement. Therefore, this restriction should only be applied depending on the direction of the motion.

These results allow us to introduce the theorem that guarantees a perfect estimation of the coverage in the reachable area of the robots at every time. This theorem proves the hypothesis of Lemmas IV.2 and IV.1.

*Theorem IV.4:* If Assumptions III.1 and III.2 hold and the motion of the robots satisfies (8), then, according to Algorithm 1, the local map is equal to the global coverage map at time instant $k$ inside the reachable area of the robot, $Z_i(k) = Z(k), \quad \forall \mathbf{q} \in \mathbf{\Omega}_i^r(k)$.

*Proof:* See Appendix D. ■

We can now extend the area in which a robot has no estimation error at each time depending on the positions of the non-neighbors.

*Corollary IV.5:* Considering the same as in Theorem IV.4, the local estimation of the coverage map of any robot $i$ is equal to the global map inside $\mathbf{\Omega}_i^\rho(k)$, $Z_i(k) = Z(k), \quad \forall \mathbf{q} \in \mathbf{\Omega}_i^\rho(k)$.

*Proof:* The same proof as for Theorem IV.4 can be applied here using $\mathbf{\Omega}_i^\rho(k)$ instead of $\mathbf{\Omega}_i^r(k)$. ■

The previous results guarantee a zero estimation error in a region that depends on the positions of the robots that are not neighbors, $m \notin N_i(k)$, at each time $k$. However, in a distributed approach, a robot does not know such positions. Regardless of the motion of the robot and the positions of the other robots there is a region, presented in the following theorem, in which zero error is guaranteed.

*Theorem IV.6:* Let $r_z^* = r^{\text{com}} - r^{\text{cov}}_{\max} - (N-1)u^{\max}$. Then, if Assumptions III.1 and III.2 hold, Algorithm 1 guarantees that $Z_i(k) = Z(k), \forall \mathbf{q} \in \mathbf{Q} \,|\, \|\mathbf{p}_i(k) - \mathbf{q}\| < r_z^*$.

*Proof:* See Appendix E. ■

Finally, we set bounds to the estimation error in all points.

*Theorem IV.7:* Consider that Assumptions III.1 and III.2 hold. Also consider that the maximum value of the production function of every robot is $P = \max_{i \in \{1\ldots,N\}} \alpha_i(k)$. Then, according to Algorithm 1, the error of the local estimation of the coverage map for all $k > 0$ is bounded by

$$0 \leq Z(k) - Z_i(k) \leq P(N-1)\frac{1-d^N}{1-d}. \tag{9}$$

*Proof:* See Appendix F. ■

The bound in (9) can also be modified to take into account disconnected networks under the standard assumption of periodic joint connectivity [39], which supposes the union of the disconnected communication graphs from a connected one at most every T instants.

*Remark IV.8:* Consider the same as in Theorem IV.7 and also consider periodic joint connectivity of the network. Then, according to Algorithm 1, the error of the local estimation of the coverage map for all $k > 0$ is bounded by

$$0 \leq Z(k) - Z_i(k) \leq P(N-1)\frac{1-d^{TN}}{1-d}.$$

## V. Motion Control

Finding the optimal actions in a coverage problem is NP-hard [21] and, thus, we propose a heuristic strategy to control the motion of the robots that makes use of our distributed estimation algorithm.

To build this solution, we introduce a new metric that characterizes how profitable it is to locate a robot at a specific position in terms of the coverage level. We call this metric *Improvement Function*. Then we divide the environment using a distributed Voronoi partition that provides each robot with a safe region where there is no interference with other robots. Inside this region, each robot can safely determine the goal that leads its movement as the maximum of the *Improvement Function*. Finally, their motion is guided by a control law combining a gradient-based and a goal-oriented action.

### A. Improvement Function

An approach to control the motion of the robots is to define goals to which the robots must move. These goals could be selected randomly according to a sampling-based motion planning algorithm [40]. However, for the distributed coverage problem, these goals must be points that provide the maximum improvement of the coverage according to the local information of the robots. To find them, we introduce a new metric that characterizes how profitable it is to locate a robot at a specific position, $\mathbf{p} \in \mathbf{Q}$, in terms of the coverage estimated by the robots. We call this metric *Improvement Function*

$$M_i(\mathbf{p},k) = \frac{\displaystyle\int_{\mathbf{\Omega}_i(\mathbf{p})} \frac{Z^* - Z_i(k)}{Z^*}\,\Phi(\mathbf{q})\,\alpha_i(\mathbf{q},\mathbf{p})\,\mathrm{d}\mathbf{q}}{\displaystyle\int_{\mathbf{\Omega}_i(\mathbf{p})} \Phi(\mathbf{q})\,\alpha_i(\mathbf{q},\mathbf{p})\,\mathrm{d}\mathbf{q}} \tag{10}$$

where $\Phi(\mathbf{q}) \in (0,1]$ is a weighting function to adjust the importance of covering each point. This metric computes the integral of the normalized coverage error $\big(Z^* - Z_i(k)\big)/Z^*$ weighted with the production of a robot located in $\mathbf{p}$, $\alpha_i(\mathbf{q},\mathbf{p})$, and normalizes it to obtain a value independent of the production function. Note that $M_i(\mathbf{p},k) \leq 1$ by definition. Its value is between 0 and 1 when the coverage area around the point $\mathbf{p}$ is slightly covered and below zero when it is overcovered. According to this definition, the goal of a robot at each time instant is the point where this function reaches its maximum.

### B. Distributed Partition of the Environment

The local *Improvement Function* (10) may not allow the robots to find the actual maximum improvement since it only uses a local estimation, $Z_i(k)$. Additionally, the selection of goals in any part of the environment may lead to long and unnecessary shifts, redundancy, or even collisions. The solution to these problems is to assign particular regions to the robots in which they are responsible for the coverage. This can be achieved partitioning the environment in a distributed fashion. In particular, we choose a distributed Voronoi tesellation. It is a well-known technique that associates to each robot its closest points of the environment.

The estimation of the coverage is always guaranteed to be exact within the zero-error area of the robots when their motion satisfies (8) and, since the size and shape of this region is constant, we reduce the Voronoi region to the intersection with this area, resulting in the r-disk Voronoi partition [41]

$$\mathbf{V}_i(k) = \{\mathbf{q} \in \mathbf{\Omega}_i^z(k) \,|\, \|\mathbf{p}_i(k) - \mathbf{q}\| \leq \|\mathbf{p}_j(k) - \mathbf{q}\| \;\; \forall j \neq i\}.$$

This region, though little conservative in some cases, presents following advantages:

1) It has been successfully applied in solutions to the static and dynamic coverage problems [7].
2) It prevents the robots from selecting as goals points in which the estimation is not exact, since inside $\mathbf{\Omega}_i^z(k)$, the estimation is always correct.
3) It overcomes the problem of long shifts, since each robot chooses goals only in its proximity.
4) It inherently prevents collisions, since each robot has an exclusive region.

When the motion is unrestricted a region of radius $\mathbf{r}_\mathbf{z}^*$ should be used to guarantee a correct estimation inside the r-disk Voronoi cell but depending on the parameters of the system, $\mathbf{r}_\mathbf{z}^*$ and the partition might be small.

It should be noted that any other partition method such as power diagrams [42] or K-Means clustering [43] could be used. The partitions could also be combined with some task assignment procedure [44] to consider in the problem the optimization of some global metric, e.g., total distance traveled by the robots, at the expense of some additional coordination. In fact, the selection of the distributed partition method is independent of the rest of the elements considered in the distributed strategy to coordinate the robots.

### C. Goal-Oriented Control

The goal-oriented control aims to direct the robots to areas where the *Improvement Function* is maximized. The direction in which each robot moves toward its goal is

$$\mathbf{u}_i^{\text{goal}}(k) = \mathbf{g}_i^*(k) - \mathbf{p}_i(k) \tag{11}$$

where $\mathbf{g}_i^*(k)$ is the goal of robot $i$ at time $k$.

In accordance with the *Improvement Function* (10), the goal should be the point $\mathbf{q}$, where it reaches its maximum value inside the Voronoi region

$$\mathbf{g}_i^*(k) = \arg \max_{\mathbf{q} \in \mathbf{V}_i(k)} M_i(\mathbf{q},k).$$

However, this point may not exist or may not be unique, for instance, when there are several points uncovered, or may change every iteration, leading to undesirable oscillations or cyclic behaviors. To cope with these problems, we maintain a list of possible goals, $\mathbf{L}_i(k)$, that is updated every iteration and from which the current goal is selected.

In our solution, any goal $\mathbf{g}_i \in \mathbf{L}_i(k)$ must be inside the Voronoi partition, $\mathbf{g}_i \in \mathbf{V}_i(k)$, and must be undercovered, $M_i(\mathbf{g}_i,k) > 0$. Therefore, the first step is to check if the goals from the list of the previous iteration, $\mathbf{g}_i \in \mathbf{L}_i(k-1)$, are still feasible, including if they have been reached

$$\mathbf{g}_i \text{ is feasible} \iff$$
$$M_i(\mathbf{g}_i,k) > 0 \wedge \mathbf{g}_i \in \mathbf{V}_i(k) \wedge \|\mathbf{g}_i - \mathbf{p}_i(k)\| > \mathcal{E} \tag{12}$$

where $\mathcal{E}$ is the distance at which a goal is considered reached. Whenever a goal stops satisfying these criteria, it is removed from the current list $\mathbf{L}_i(k)$.

Next, we define a region in which new possible goals are searched as

$$\mathbf{R}_i(k) = \mathbf{V}_i(k) \setminus \boldsymbol{\Omega}_i\big(\mathbf{p}_i(k)\big) \setminus \boldsymbol{\Omega}_i(\mathbf{g}_i) \tag{13}$$

that includes all points of the Voronoi region that are not in the coverage area of the robot centered at $\mathbf{p}_i(k)$ or at any $\mathbf{g}_i \in \mathbf{L}_i(k)$. Note that in the initialization step for $\mathbf{L}_i(1)$, the search region is $\mathbf{R}_i(1) = \mathbf{V}_i(1)$, and that each new possible goal found in the search region meets the distance requirement

$$\mathbf{g}_i^2 \notin \boldsymbol{\Omega}_i(\mathbf{g}_i^1) \tag{14}$$

with $\mathbf{g}_i^1$, $\mathbf{g}_i^2$ any pair of possible goals in $\mathbf{L}_i(k)$.

Each robot iteratively looks for new possible goals inside $\mathbf{R}_i(k)$, that are global or local maxima or quasi-maxima of the *Improvement Function*. The set of these points is

$$\mathrm{P}_i = \big\{\mathbf{p} \in \mathbf{R}_i(k) \,|\, M_i(\mathbf{p}, k) \geq M_i(\mathbf{p} + \epsilon, k)\big\} \tag{15}$$

which can be found using state-of-the-art methods such as pattern search. Any $\mathbf{p} \in \mathrm{P}_i$ must also satisfy $M_i(\mathbf{p}, k) > 0$. Nevertheless, from this set only the nearest point to the robot is selected as a new possible goal

$$\mathbf{g}_i = \arg\min_{\mathbf{p} \in \mathrm{P}_i} (\|\mathbf{p} - \mathbf{p}_i(k)\|). \tag{16}$$

This point, $\mathbf{g}_i$, is the most interesting of the current search region both in terms of the *Improvement Function* and in terms of the time until it can be covered. Thus, it is appended to the current list of goals, $\mathbf{L}_i(k)$, and the points close to it are eliminated from the search region to keep satisfying (14)

$$\mathbf{R}_i(k) = \mathbf{R}_i(k) \setminus \boldsymbol{\Omega}_i(\mathbf{g}_i). \tag{17}$$

The search is repeated until $\mathbf{R}_i(k)$ is empty or there are no more points with a positive value of the *Improvement Function* and eventually, each robot has an updated list of possible goals, $\mathbf{L}_i(k)$.

The selection of the goal to follow from this list can be done in accordance with several criteria such as the value of the *Improvement Function*, $M_i(\mathbf{g}_i, k)$, the distance to the current position of the robot or the distance to the neighbors. This means that, for instance, the most profitable goal may be the one with the highest value of $M_i(\mathbf{q}, k)$ or the nearest to the robot and it is clear then that the definition of profitability of a goal depends on the application. Therefore, the selection of the current goal from the list can be generalized as

$$\mathbf{g}_i^*(k) = \arg\min_{\mathbf{g}_i \in \mathbf{L}_i(k)} f(M_i(\mathbf{g}_i, k), \|\mathbf{g}_i - \mathbf{p}_i\|, \|\mathbf{g}_i - \mathbf{p}_j\|) \tag{18}$$

where $j \in N_i(k)$ and the *Profitability Function* $f(\cdot)$ represents the profitability of a goal from the list. In particular, in this paper, we use $f(\mathbf{g}_i, k) = -M_i(\mathbf{g}_i, k)$. However, to give a more general idea of this function in the simulation section, we provide a comparison of several different alternatives.

Finally, in Algorithm 2, we summarize this method to search for goals.

**Algorithm 2** *Selection of Goals*

**Require:**
– Previous list of goals, $\mathbf{L}_i(k-1)$.
– Voronoi region of the robot, $\mathbf{V}_i(k)$.
– *Improvement Function*, $M_i(\mathbf{q}, k) \quad \forall\, \mathbf{q} \in \mathbf{V}_i(k)$.

1: $\mathbf{L}_i(k) = \mathbf{L}_i(k-1)$
2: For all $\mathbf{g}_i \in \mathbf{L}_i(k-1)$
3: If $\mathbf{g}_i$ is not feasible, (12),
4: Remove $\mathbf{g}_i$ from $\mathbf{L}_i(k)$.
5: Define search region $\mathbf{R}_i(k)$, (13).
6: While $\mathbf{R}_i(k) \neq \emptyset$ and $\exists\, \mathbf{p} \in \mathbf{R}_i(k) \,|\, M_i(\mathbf{p}, k) > 0$
7: Search for a new possible goal $\mathbf{g}_i \in \mathbf{R}_i(k)$, (15)–(16).
8: Append $\mathbf{g}_i$ to $\mathbf{L}_i(k)$.
9: Eliminate points near $\mathbf{g}_i$ from $\mathbf{R}_i(k)$, (17).
10: Select current goal $\mathbf{g}_i^* \in \mathbf{L}_i(k)$, (18).

### D. Local, Gradient-Based Control

The motion toward the goal aims to direct the robot to poorly covered areas that maximize the improvement of the coverage. However, the path to the goal may go through well- or overcovered areas, that can deteriorate the overall performance. To avoid it and move across areas where the coverage level is under the objective, we propose an additional action to maximize the improvement of the coverage in the trajectory to the goal. For this reason, we use the gradient of the *Improvement Function* with respect to the robot position

$$\mathbf{u}_i^{\text{grad}}(k) = \nabla_{\mathbf{p}_i} M_i\big(\mathbf{p}_i(k), k\big) = \frac{\displaystyle\int_{\boldsymbol{\Omega}_i} \frac{\Phi}{Z^*} \nabla_{\mathbf{p}_i} \alpha_i(k) A(k)\, \mathrm{d}\mathbf{q}}{\displaystyle\int_{\boldsymbol{\Omega}_i} \Phi \alpha_i(k)\, \mathrm{d}\mathbf{q}} - \frac{\displaystyle\int_{\boldsymbol{\Omega}_i} \Phi \frac{Z^* - Z_i(k)}{Z^*} \alpha_i(k)\, \mathrm{d}\mathbf{q} \int_{\boldsymbol{\Omega}_i} \Phi \nabla_{\mathbf{p}_i} \alpha_i(k)\, \mathrm{d}\mathbf{q}}{\left(\displaystyle\int_{\boldsymbol{\Omega}_i} \Phi \alpha_i(k)\, \mathrm{d}\mathbf{q}\right)^2} \tag{19}$$

with

$$A(k) = Z^* - dZ_i(k-1) - \sum_{j \in N_i(k)} \alpha_j^i(k) - 2\alpha_i(k) \tag{20}$$

$$\alpha_j^i(k) = \max\big(Z_j^{\text{com}}(k) - dZ_i(k-1), 0\big) \quad \forall\, \mathbf{q} \in \boldsymbol{\Omega}_i(k). \tag{21}$$

This gradient represents how the *Improvement Function* changes inside the coverage area of robot $i$ due to small variations of its position and it gives the direction of the motion of the robot that maximizes the increase of the improvement, i.e., the direction in which the coverage would be most improved. A more detailed formulation of how this gradient is obtained can be found in Appendix G.

The use of this kind of gradient descent solution is usually applied to centralized systems [26]. Fortunately, our distributed estimation algorithm of Section III allows us to use the same gradient-based controller as a centralized system to obtain the same solution because each robot has an accurate estimation of the global map in the area required for the computation of the gradient.

*Proposition V.1:* The response of the gradient-based motion controller from (19) is the same for our distributed system as for a centralized system with

$$B(k) = Z^* - dZ(k-1) - \sum_{j \in \{1,\ldots,N\}} \alpha_j(k) - \alpha_i(k) \tag{22}$$

instead of $A(k)$ and $Z(k)$ instead of $Z_i(k)$.

*Proof:* In the first place, we demonstrate that $\sum_{j\in N_i(k)} \alpha_j(k) = \sum_{j\in N_i(k)} \alpha_j^i(k) \quad \forall \mathbf{q} \in \mathbf{\Omega}_i(k)$. According to the definition of the map-to-communicate from (2), $\alpha_j(k) = Z_j^{\text{com}}(k) - dZ_j(k-1)$. Inside $\mathbf{\Omega}_i(k) \cap \mathbf{\Omega}_j^r(k)$, Theorem IV.4 states that $Z_j(k) = Z(k) = Z_i(k)$. Then

$$\begin{aligned}\alpha_j^i(k) &= Z_j^{\text{com}}(k) - dZ_i(k-1)\\ &= \alpha_j(k) \quad \forall \mathbf{q} \in \mathbf{\Omega}_i(k) \cap \mathbf{\Omega}_j^r(k).\end{aligned}$$

In $\mathbf{\Omega}_i(k) \setminus \mathbf{\Omega}_j^z(k)$, according to Theorem IV.7, $Z_j(k-1) \leq Z(k-1) = Z_i(k-1)$. Then, we have $Z_j^{\text{com}}(k) - dZ_i(k-1) \leq 0$ that implies $\alpha_j^i(k) = 0 = \alpha_j(k)$, completing the first part of the proof.

If we now introduce this result in (22) we have

$$\begin{aligned}B(k) = Z^* - dZ(k-1) - \sum_{j\in N_i(k)} \alpha_j^i(k)\\ - \sum_{j\notin N_i(k)} \alpha_j(k) - \alpha_i(k).\end{aligned}$$

Since the integrals in (19) are only computed inside $\mathbf{\Omega}_i(k)$ and in this region $\alpha_j(k) = 0 \quad \forall j \notin N_i(k),\ j \neq i$, then we have $\sum_{j\notin N_i(k)} \alpha_j(k) = \alpha_i(k)$. Therefore

$$B(k) = Z^* - dZ(k-1) - \sum_{j\in N_i(k)} \alpha_j^i(k) - 2\alpha_i(k) = A(k).$$

Recalling that in $\mathbf{\Omega}_i(k)$, Theorem IV.4 proves that $Z_i(k) = Z(k)$ since $\mathbf{\Omega}_i(k) \subset \mathbf{\Omega}_i^r(k)$, the proof is complete. ■

This proposition demonstrates that with our proposal any robot can calculate its own action using only local information and obtain the same results as a centralized system.

### E. Motion Control Law

The overall motion control law that we propose gathers both previous actions, the local, gradient-based, $\mathbf{u}_i^{\text{grad}}(k)$, and the goal-oriented, $\mathbf{u}_i^{\text{goal}}(k)$:

$$\mathbf{u}_i(k) = W_L(k) u_i^{\max}(k) \frac{W_L(k)\mathbf{u}_i^{\text{grad}}(k) + W_G(k)\mathbf{u}_i^{\text{goal}}(k)}{\|W_L(k)\mathbf{u}_i^{\text{grad}}(k) + W_G(k)\mathbf{u}_i^{\text{goal}}(k)\|} \tag{23}$$

where $u_i^{\max}(k)$ is bounded by Lemma IV.3 or by $u^{\max}$ and $\mathbf{u}_i^{\text{grad}}(k)$ and $\mathbf{u}_i^{\text{goal}}(k)$ are the normalized directions introduced in (19) and (11), respectively. The weight $W_L(k)$ of the gradient-based action is also used as a gain of the maximum velocity of the robot and is defined according to the *Improvement Function* from (10)

$$W_L(k) = 1 - \max\left(M_i\left(\mathbf{p}_i(k), k\right), 0\right)$$

where $0 \leq W_L(k) \leq 1$. To avoid revisiting areas already well covered, the importance of the gradient action tends to 1 when the coverage level is near to the objective in the coverage area of the robot. When the coverage level is poor, the local direction becomes less important since coverage is still needed. This weight also acts as a velocity gain. When the area has a low coverage level, it has the effect of slowing the robot down to keep covering it and of speeding it up to quickly move to less covered areas when its coverage level is near the objective.

The importance of the goal-oriented action is

$$W_G(k) = M_i\left(\mathbf{g}_i(k), k\right)$$

where $0 \leq W_G(k) \leq 1$ and $\mathbf{g}_i(k)$ is the goal of robot $i$, described in the Section V-C. This weight increases the importance of the goal when the coverage area of the robot virtually located there is poorly covered and, vice versa, the goal becomes less important when it is well covered.

### F. Additional Considerations

We devote this final part to present several aspects of the motion control and the persistent coverage approach that are worth mentioning.

First, there is no guarantee that the trajectories obtained with the proposed motion control law are optimal and, therefore, the coverage strategy may also be suboptimal. However, in the simulations sections, we evaluate the performance of the system and show the kindness of our approach.

In the second place, it is known that gradient methods may lead to local optima and, therefore, the robots could end up stuck in one of these minima. However, there are several features of our approach that prevent this situation, as follow:

1) we follow goals as well as the gradient and include the variable weights in the motion control law (23);
2) the coverage level outside the coverage area of the robots decreases when it is not covered and, even if a robot reaches a local minimum, the area around it will decay and will require the robot to cover it;
3) if both weighted actions cancel out, the robot does not move but keeps covering, making the weight of the gradient $W_G$ change and, thus, breaking the equality.

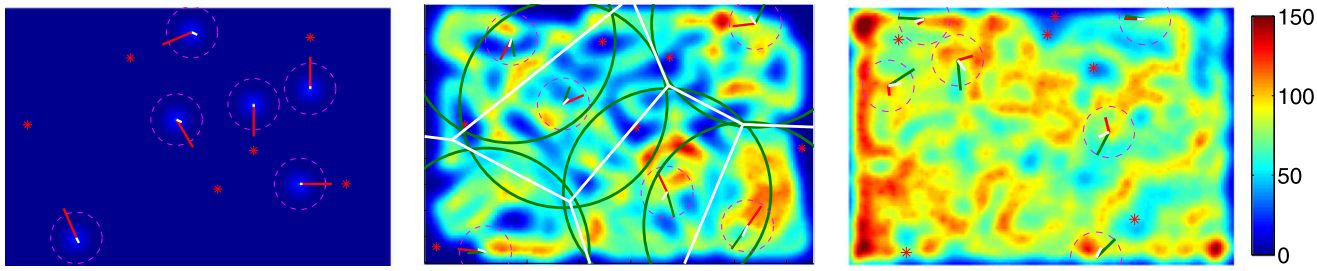
This does not imply that all the points are theoretically guaranteed to reach the objective even though it should happen in practice.

Additionally, although the control law does not guarantee collision-free motion, the Voronoi partition prevents collisions, as mentioned earlier. Also, state-of-the-art collision avoidance methods such as potential fields can be incorporated into the system and used if the gradient term dominates the motion.

Finally, the discrete-time formulation of the problem requires the frequency of the estimation to be sufficiently high w.r.t. the maximum velocity and the coverage radius of the robots to avoid discretization problems. Since the actual production is continuous, the estimation must be run fast enough not to miss sensible information between two consecutive estimation instants. The goal selection and motion control do not need to be run as fast as the estimation. In fact, there must be a tradeoff between the quality of the solution and the computational cost in the sense that the more frequent the calculation is carried out, the better the final coverage is obtained but the higher computational cost is incurred. Regarding the order, the goal selection and motion step should follow the estimation step in order to calculate the actions with the latest information.

## VI. Simulations

Eventually, we present simulation results for the proposed solution to the persistent coverage problem. At the beginning, we show the behavior of the entire system using an illustrative example. Afterward, we provide an exhaustive evaluation of




(a) k = 1 (b) k = 150 (c) k = 300

Fig. 3. Example of simulation: coverage map with the coverage areas of the robots (magenta circumferences) and their motion control actions for three different instants. Red lines represent the goal-oriented term, $W_G(k)\mathbf{u}_i^{\text{goal}}(k)$, magnified by 2.5 and point to the goals that are represent by red asterisks. Green lines represent the gradient-based term, $W_L(k)\mathbf{u}_i^{\text{grad}}(k)$, magnified by 2.5. The resulting action is represented in white. Green circumferences in (b) represent the zero-error areas of the robots and the thickest white lines, the Voronoi partition of the environment.

the local map updating strategy, the proposed alternatives to the *Profitability Function*, the control of the motion of the robots and the whole approach to the problem using a Monte Carlo analysis.

The reference settings that we have used in these simulations are the following. The environment $\mathbf{Q}$ is a rectangle of $100 \times 150$ units with a decay rate $d = 0.995$ and the desired coverage level is $Z^* = 100$ for every point. The team consists of $N = 6$ robots, whose maximum motion is $u^{\max} = 5$. The production function is equal to

$$\alpha_i(k) = \frac{P}{r_i^{\text{cov}2}}(r - r_i^{\text{cov}})^2, \qquad \text{if } r = \|\mathbf{p}_i(k) - \mathbf{q}\| \leq r_i^{\text{cov}}$$

and $\alpha_i(k) = 0$, otherwise. The maximum value is $P = 25$ and the coverage radius, $r_i^{\text{cov}} = 10$ units. The communication radius is $r^{\text{com}} = 120$, that clearly satisfies Assumption III.1. Note that in a real-world environment, these parameters are fixed by the application and by the specific robots used. Thus, for the simulation, the selection is arbitrary. In particular, we select the number of robots that is able to produce more coverage than the total decay of the environment when it is at the desired level, i.e., we use the minimum number of robots that satisfy

$$\int_{\mathbf{Q}} \sum_{i\in\{1,\ldots,N\}} \alpha_i(k)\mathrm{d}\mathbf{q} > \int_{\mathbf{Q}} (1-d)Z^*\mathrm{d}\mathbf{q}.$$

Although this is a sufficient condition for the robots to be able to reach the desired coverage level, it is not a requirement of the algorithm. If less robots were deployed, they would try to keep the average coverage level as high as possible. Also note that, without loss of generality, we use the same coverage radius and the same coverage function for all the robots although they could be different [26] without modifying the global behavior of the system. In addition, we define the quadratic coverage error

$$\tilde{\varepsilon}(k) = \int_{\mathbf{Q}} \left(Z^* - Z(k)\right)^2 \mathrm{d}\mathbf{q}$$

that is an interesting metric to evaluate the performance.

The simulations are implemented in MATLAB, run on a laptop with an Intel Core i7 and the average iteration time for these settings is around 80 ms.

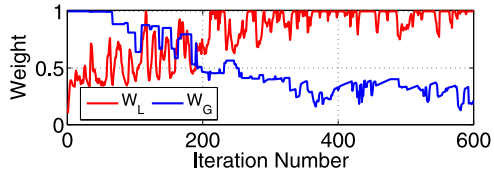



Fig. 4. Evolution of the weights of the actions $W_L$ and $W_G$ for one of the robots.

### A. Illustrative Example

This section presents an illustrative example of the simulations with the reference settings. In Fig. 3, we depict the coverage map with the coverage areas of the robots and their motion control actions (23) for three different instants. The average coverage level reached is 96.4 with a standard deviation of 21.3 that is, at most, 30 in the transient state. This supports that, although it is not theoretically guaranteed, most of the points reach the desired level at some time and also an average level near the objective is maintained over time.

Fig. 3(a) shows that, in the beginning, there is almost no movement since the environment is uncovered and the weight $W_L$, that is the velocity gain, is small. At this time, the goal-oriented term, $W_G(k)\mathbf{u}_i^{\text{goal}}(k)$, prevails. When the environment starts to be covered, the combination of both terms drives the robots [see Fig. 3(b)]. In the end, when all the environment is covered, the term that has more influence is the gradient-based, $W_L(k)\mathbf{u}_i^{\text{grad}}(k)$, since there are few uncovered areas that are worth visiting [see Fig. 3(c)]. In Fig. 4, the weight of each action is depicted for one of the robots. It confirms that the goal-oriented term is more important in the transient, while it is the gradient in the steady state and that the velocity gain $W_L$ progressively increases with the coverage.

The trajectories for the first 200 instants of this simulation are shown in Fig. 5. At start, the robots focus on their particular regions for three reasons:

1) they follow the goals from the initial list, that belong to the same region and all are poorly covered;
2) the goal term is more important in the control law; and
3) they move slowly since the velocity gain, $W_L(k)$, is small due to the low coverage of the environment.

As the algorithm proceed, the gradient term dominates the motion and the velocity is higher making the covered regions of the robots overlap more and more. It is worth mentioning that

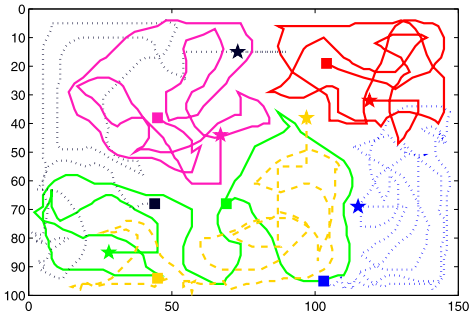


Fig. 5. Trajectories followed by the six robots during the first 200 iterations. Initial positions are represented with stars and final positions with squares.

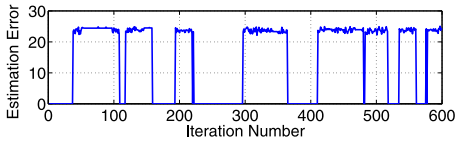



Fig. 6. Evolution of the maximum estimation error of the network.

for these settings, and in particular for this number of robots and communication radius, the network did not become disconnected in any of the carried-out simulations.

In Theorem IV.7, we state that the estimation error is bounded by (9). If we calculate the maximum estimation error of the network for this example, the maximum value reached is 25.1, as can be seen in Fig. 6. This error is lower than the value of our upper bound that is 740.7. Although the theoretical bound is loose, it shows that the estimation error does not go to infinity and remains bounded. In practice, it can be seen that the estimation error is much lower and that the algorithm works even better than theoretically predicted. The minimum value of the estimation error is always 0 which also confirm the lower bound in (9).

Now, we focus in the characterization of the estimation. In Theorem IV.4, we state that the estimation is perfect inside the reachable area of each robot when the motion action satisfies (8) and in Theorem IV.6, the area in which there is no estimation error when no restriction is applied to the motion.

To check that these statements hold, we use the minimum distance to an estimation error between all the robots (MDE). This distance is the shortest Euclidean distance from a robot to the points in which it has an estimation error, i.e., where $Z_i(k) \neq Z(k)$

$$\text{MDE}(k) = \min_{i,\mathbf{q}} \left( \|\mathbf{q} - \mathbf{p}_i\| \mid Z_i(\mathbf{q}, k) \neq Z(\mathbf{q}, k) \right).$$

In Fig. 7, we depict the MDE in each iteration. When the motion satisfies (8) [see Fig. 7(a)], the MDE is always greater than the radius of the reachable area (red line) and, thus, comply with Theorem IV.4. It is also greater than the radius of the zero-error area (green dash-dotted line) that has a constant value. On the other hand, when there is no restriction to the motion, Fig. 7(b) shows that MDE is greater than $r_z^* = 85$ (red line) as asserted in Theorem IV.6. In fact, although it is not theoretically guaranteed, it can be seen that most of the time the estimation is correct in an area with the same radius as the zero-error area, i.e., MDE greater than 110, and even in a bigger region. This is an advantage since the velocity restriction from (8) cannot be

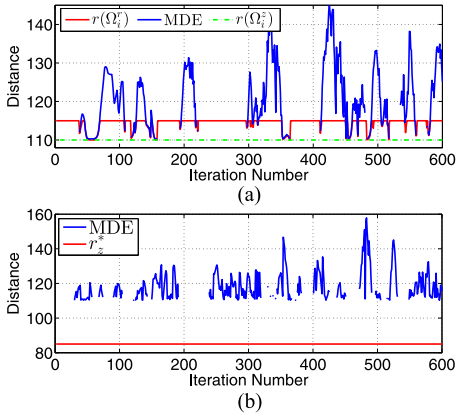



Fig. 7. Evolution of the MDE. (a) Motion satisfies Equation (8). (b) No restriction to the motion.

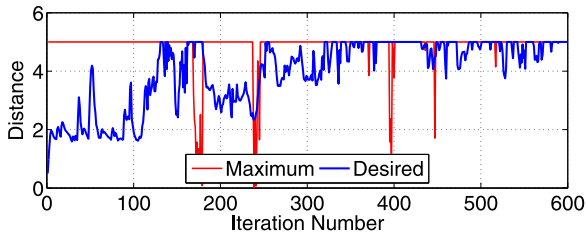



Fig. 8. Movement of a robot. The blue line represents the desired movement of the robot and the red line, the maximum movement allowed by Lemma IV.3.

calculated in a distributed scenario and allows us to omit it in a real-world application. Note that the MDE is not depicted in the iterations in which all the robots have a perfect estimation.

If we test the restriction in terms of the motion of the robots, this example shows that, in practice, it is not very limiting. Fig. 8 shows the maximum allowed movement in the calculated direction at each iteration for a robot. It can be seen that the desired movement is under the maximum allowed in most cases.

### *B. Local Estimation of the Coverage*

In order to perform a thorough validation, all the analyses that we present in the following sections have been developed through a Monte Carlo analysis. This analysis includes 20 runs of the simulations from 20 different initial positions.

First, we study the influence of the parameters of the system in our local map updating strategy from Algorithm 1. In particular, we evaluate the influence of varying the decay, $d$, the number of robots, $N$, the communication radius, $r^{\text{com}}$, the maximum value of the production, $P$, and the coverage radius, $r^{\text{cov}}$, with respect to the reference settings. To do so, we synthesize the two measurements introduced in the example: we calculate the maximum estimation error and the minimum MDE over all the runs and average both over time. We call them $\overline{\varepsilon}_{\max}$ and $\overline{\text{MDE}}$, respectively. In Table I, the results of this study are summarized and the conclusions from them are as follows:

1) The variation of the decay has a small influence on $\overline{\varepsilon}_{\max}$. This error is around the value of $P$ which means two things: few overlaps take place outside the zero-error area of the robot and the production of the non-neighbors need

TABLE I
MAXIMUM ESTIMATION ERROR, $\overline{\varepsilon}_{max}$, AND MINIMUM DISTANCE TO AN ESTIMATION ERROR, $\overline{\text{MDE}}$.

| **d** | **0.991** | **0.993** | **0.995** | **0.997** | |
|---|---|---|---|---|---|
| $\overline{\varepsilon}_{max}$ | 25.3 | 25.4 | 25.8 | 26.2 | |
| $\overline{\text{MDE}}$ | 111.3 | 111.2 | 111.2 | 111.3 | |
| **N** | **4** | **6** | **8** | **12** | **16** |
| $\overline{\varepsilon}_{max}$ | 25.2 | 25.8 | 26.3 | 29.5 | 31.8 |
| $\overline{\text{MDE}}$ | 112.6 | 111.2 | 110.8 | 110.5 | 110.4 |
| $\mathbf{r}^{com}$ | **80** | **100** | **120** | **140** | **160** |
| $\overline{\varepsilon}_{max}$ | 42.6 | 32.7 | 25.8 | 22.4 | 4.1 |
| $\overline{\text{MDE}}$ | 70.5 | 90.8 | 111.2 | 132.4 | 153.8 |
| **P** | **15** | **20** | **25** | **30** | **35** |
| $\overline{\varepsilon}_{max}$ | 15.1 | 20.3 | 25.8 | 31.7 | 36.6 |
| $\overline{\text{MDE}}$ | 111.2 | 111.3 | 111.2 | 111.1 | 111.2 |
| $\mathbf{r}^{cov}$ | **5** | **10** | **15** | **20** | |
| $\overline{\varepsilon}_{max}$ | 25.1 | 25.8 | 28.0 | 31.4 | |
| $\overline{\text{MDE}}$ | 116.6 | 111.2 | 106.1 | 101.1 | |

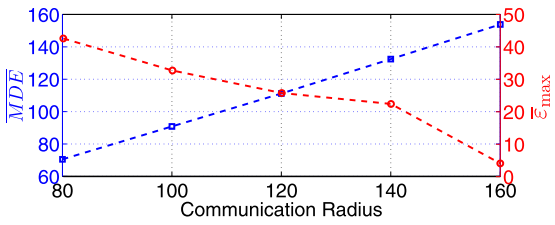



Fig. 9. Evolution of the estimation with the communication radius.

only one instant to reach the robot. In these conditions, the decay does not play an important role. In addition, the decay does not influence the $\overline{\text{MDE}}$ since it only modifies the coverage level. Further tests showed that both conclusions are valid for even larger variations of the decay rate.

2) An increase in the number of robots makes $\overline{\varepsilon}_{max}$ increase because the number of overlappings rises. On the contrary, it makes the $\overline{\text{MDE}}$ decrease because it is more likely that a robot appears on the limit of the zero-error area. Nevertheless, the influence of this parameter is small in both measurements.
3) The communication radius is the parameter with the biggest influence in the estimation. For this reason, we represent the evolution of the two measurements in Fig. 9. It can be seen that $\overline{\varepsilon}_{max}$ decreases when the communication radius increases. This happens because a larger radius implies that more robots are in direct communication, reducing the error drastically. On the contrary, the $\overline{\text{MDE}}$ increases with the communication radius, since the production of the robots that are in direct communication does not induce an estimation error.
4) The influence of the maximum production is very important since it directly modifies the estimation error. However, as well as the decay, it has no effect in the distance to the error.
5) The coverage radius makes $\overline{\varepsilon}_{max}$ increase since the bigger the radius is, the more overlappings occur. The $\overline{\text{MDE}}$ decreases with this radius. If two robots are not neighbors, a larger coverage radius makes the $\overline{\text{MDE}}$ that they produce on each other shorter.

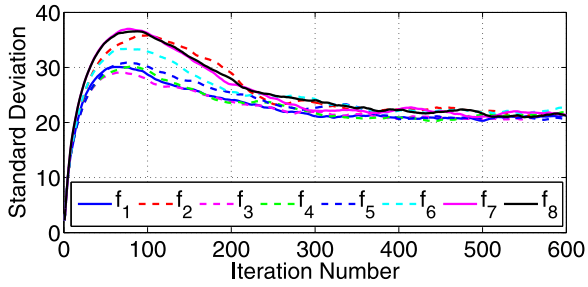



Fig. 10. Average standard deviation of the coverage level for the different *Profitability Functions* and additional selection methods.

Although it cannot be shown graphically for all the trials, it should be noted that both the boundedness of the estimation error and the correct estimation in the characterized areas held at all times for all the simulations that we have carried out.

### C. Profitability Function

As mentioned in Section V-C, the *Profitability Function*, $f(\cdot)$, that allows the selection of the current goal from the list can be defined in a different manner depending on the application. In this section, we compare several different alternatives to see the influence that they have in the global behavior of the system. The first one selects the point with the best value of the *Improvement Function*, $f_1(\mathbf{g}, k) = -M_i(\mathbf{g}, k)$. The second one is based on the minimum distance to the objectives, in which we want to reduce the overall motion, $f_2(\mathbf{g}, k) = \|\mathbf{g} - \mathbf{p}_i(k)\|$. In the third place, we weight the previous two, varying the importance of the *Improvement Function* and the distance

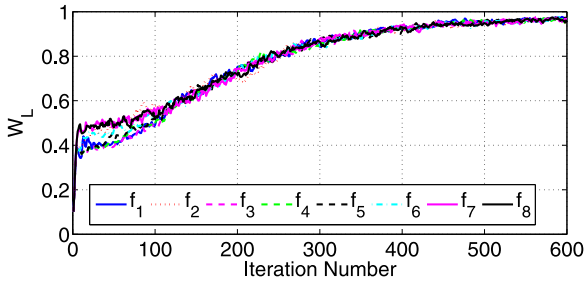
$$f_3(\mathbf{g}, k) = -0.7\, M_i(\mathbf{g}, k) + 0.3\, \|\mathbf{g} - \mathbf{p}_i(k)\|$$

$$f_4(\mathbf{g}, k) = -0.5\, M_i(\mathbf{g}, k) + 0.5\, \|\mathbf{g} - \mathbf{p}_i(k)\|$$

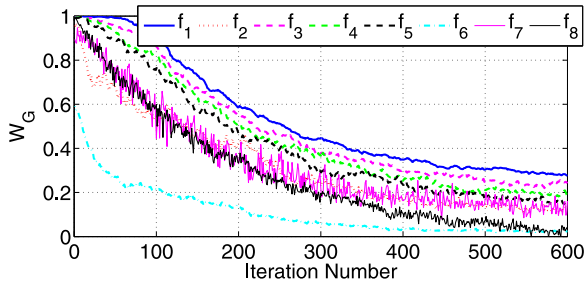
$$f_5(\mathbf{g}, k) = -0.3\, M_i(\mathbf{g}, k) + 0.7\, \|\mathbf{g} - \mathbf{p}_i(k)\|.$$

We also include in the comparison other selection methods that may not work properly, to have a better idea of the performance of all of them. The sixth alternative is to select the goal with the lowest value of the *Improvement Function*, i.e., the point of the list that causes the smallest improvement of the coverage, $f_6(\mathbf{g}, k) = M_i(\mathbf{g}, k)$. Additionally, we use a random selection of the goal from the list and we refer to it as $f_7(\mathbf{g}, k)$. Finally, we also compare these methods with a random selection of the goal in the Voronoi region of the robot. Instead of constructing a list of goals, we directly select a random point of the region as the goal to follow. We refer to this as $f_8(\mathbf{g}, k)$.

From the simulations of these eight alternatives with the reference settings, we obtained the following results. First, we find that the standard deviation of the coverage level averaged over the runs is very significant to compare the options as can be seen in Fig. 10. The average coverage level is the same in all the cases since the production of the robot does not change, but the standard deviation gives an idea of how homogeneous is the coverage of the environment. The figure shows that the lowest deviation is achieved with $f_1$, $f_3$, $f_4$, and $f_5$, i.e., with the *Profitability Functions* that look for the goal with the maximum value of the improvement. These selections lead to a more




(a) $W_L$




(b) $W_G$

Fig. 11. Evolution of the weights for the different *Profitability Functions*.

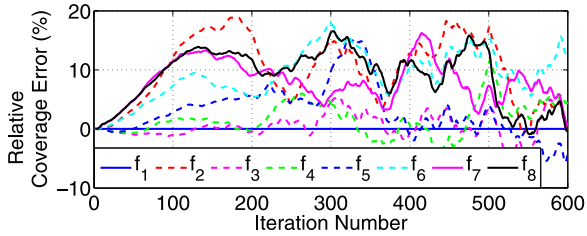



Fig. 12. Relative quadratic coverage error with respect to $f_1$ for the different *Profitability Functions* and additional selection methods.

TABLE II
RELATIVE COVERAGE ERROR WITH RESPECT TO $f_1$ AVERAGED OVER TIME

| $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ |
|---|---|---|---|---|---|---|---|
| 0% | 10.1% | 0.5% | 1.7% | 3.1% | 9.2% | 8.0% | 9.1% |

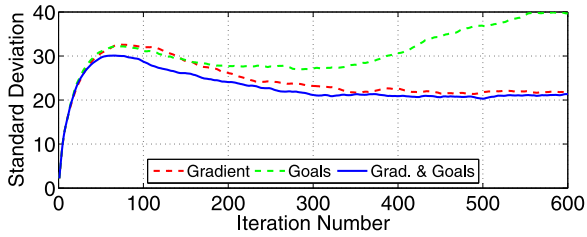



Fig. 13. Average standard deviation of the coverage level for the different motion control laws.

homogeneous coverage of the environment than the others in the transient state.

The *Profitability Functions* that look for the worst covered areas select goals that produce high values of $W_G$ and not only direct the robots to the most profitable areas but also give more importance to the goal-oriented term in the control law. On the contrary, if the improvement of the selected goals is really low, as for $f_6$ or $f_8$, $W_G$ tends to zero. In that case, the gradient term dominates the motion and the results tend to be similar to the ones obtained only with the gradient term. This is the reason why the performance differences between so different *Profitability Functions* are not larger and why all of them converge to the same standard deviation in the steady state, because eventually the gradient term dominates the motion.

Fig. 11 shows the average weights for the eight *Profitability Functions*. It can be seen in Fig. 11(a) that the gradient weight is very similar for all of them except for those based upon the *Improvement Function* in the first 100 iterations. In the end, the weight of the gradient is almost 1 in all cases, supporting that all the alternatives converge to the same value of the standard deviation. However, the most important result is the influence of the *Profitability Function* in $W_G$, shown in Fig. 11(b). The chosen function clearly determines which term is more important in the control action in the transient: the goal-oriented term dominates in the beginning for $f_1$, $f_3$, $f_4$, and $f_5$ but for $f_6$, the gradient rules the motion from the start. This is the reason why $f_1$ or $f_3$ give much better results than the other alternatives in the transient state.

Additionally, Fig. 11(a) gives an idea of the distances traveled by the robots since $W_L$ acts as a velocity and the total traveled distance is proportional to the integral of $W_L$ over time. Intuitively, $f_2$ should lead to the shortest distance but in practice it does not because the nearest goals may not be the most profitable and then the robots move faster and travel more distance.

Finally, we compare the performance of the alternatives using the quadratic coverage error of the system, $\tilde{\varepsilon}(k)$. Upon this metric, we compute the relative coverage error as the quadratic coverage error of each alternative relative to the best one. In this case, we calculate $\big(\tilde{\varepsilon}(k) - \tilde{\varepsilon}_{f_1}(k)\big)/\tilde{\varepsilon}_{f_1}(k)$, where $\tilde{\varepsilon}_{f_1}(k)$ is the quadratic coverage error obtained with $f_1$, so as to obtain a representative measure in percentage for each alternative (see Fig. 12). It shows that the *Profitability Functions* that are not built upon the *Improvement Function* perform up to 20% worse.

If we average these relative errors over time (see Table II), we can have a single measure of the differences that the selection methods cause on the performance of the system. The conclusion from these results is that, although the *Profitability Function* is important conceptually, using a criterion that is based on the maximum *Improvement Function* of the coverage, such as $f_1$ or $f_3$, leads to a coverage around a 10% better than using any other selection that does not take it into account.

### D. Motion Control

In the third place, we provide results regarding the control of the motion of the robots. To evaluate the influence of the restriction from (8) in terms of the motion of the robots, we ran a Monte Carlo simulation with the reference settings and concluded that, in practice, this restriction in the motion is not very limiting. In 20 runs of 600 iterations, it was only applied in the 6.4% of the cases and caused an average reduction of the motion of the 54.0%.

Next, we compare our motion control law with the performance of the gradient-term and the goal-oriented term, separately. In Fig. 13, it can be seen that the average standard

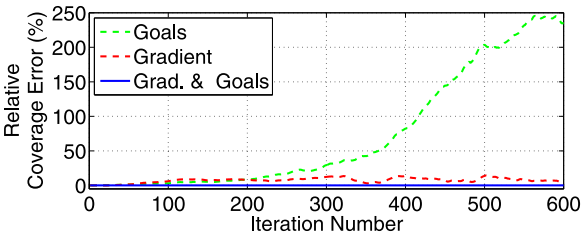



Fig. 14. Relative quadratic coverage error with respect to (23) for the different motion control laws.

TABLE III
RELATIVE COVERAGE ERROR WITH RESPECT TO (23) AVERAGED OVER TIME

| Gradient and Goals | Gradient | Goals |
|---|---|---|
| 0% | 7.5% | 73.4% |

deviation of the coverage level is greater when only one of the terms of the control law is used. This means that our motion control law produces a more homogeneous coverage in the environment. It is also remarkable that, in the steady state, the control using only the goal-oriented term has a much higher deviation. This happens when the environment is already well covered because the straight path to the objective produces overcovered regions. On the contrary, the behavior of the gradient-term separately is very similar to the entire law because this term prevails over the goal oriented.

We also represent the quadratic coverage error of each term separately relative to the quadratic coverage error obtained with our motion control law, $\left(\tilde{\varepsilon}(k) - \tilde{\varepsilon}_{u_i}(k)\right)/\tilde{\varepsilon}_{u_i}(k)$, in Fig. 14. It shows that the goal-oriented term alone also produces a great increment of the coverage error at the end of the simulations while the gradient term does not.

Table III demonstrates that, on average, the coverage with our motion control law performs 7.5% better than only the gradient-based term and 73.4% better than only the goal-oriented term.

### E. Global Performance Analysis

In this final part, we provide simulation results on the performance of our entire approach to the persistent coverage problem, which comprises of the estimation algorithm and the motion control. Specifically, we study the influence of the parameters of the system in the quadratic coverage error of the steady state, calculated for $k = 600$, and in the standard deviation of the coverage level.

In Fig. 15, we present the evolution of the quadratic coverage error of the steady state averaged over the runs for different values of the parameters. It can be seen that, in terms of this coverage error, there is an optimum value for each parameter when all the others are fixed and, when a different value is applied, the coverage error of the steady state is higher. The reason is that higher values of the parameters increase the production of the system or decrease the deterioration of the coverage, leading to a coverage level of the environment greater than the objective. Similarly, lower values lead to an undercovered environment. In both cases, the quadratic coverage error is higher than in a well-covered environment according to its definition.

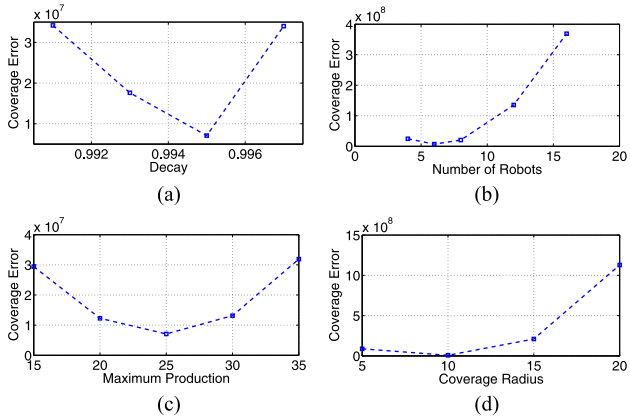



Fig. 15. Evolution of the average quadratic coverage error of the steady state for different values of the parameters. (a) d = 0.991, 0.993, 0.995, 0.997. (b) N = 4, 6, 8, 12, 16. (c) P = 15, 20, 25, 30, 35. (d) rcov = 5, 10, 15, 20.

TABLE IV
AVERAGE COVERAGE LEVEL OF THE STEADY STATE

| **d** | **0.991** | **0.993** | **0.995** | **0.997** | |
|---|---|---|---|---|---|
| $\overline{Z}$ | 56.9 | 72.0 | 96.4 | 140.5 | |
| **N** | **4** | **6** | **8** | **12** | **16** |
| $\overline{Z}$ | 65.1 | 96.4 | 127.8 | 190.7 | 253.6 |
| **P** | **15** | **20** | **25** | **30** | **35** |
| $\overline{Z}$ | 58.6 | 77.5 | 96.4 | 115.4 | 134.6 |
| $\mathbf{r}^{\mathbf{cov}}$ | **5** | **10** | **15** | **20** | |
| $\overline{Z}$ | 25.0 | 96.4 | 212.1 | 368.4 | |

These results are supported by the specific values of the average coverage level of the steady state, gathered in Table IV. Since the robots are capable of increasing the coverage level, the average level of the environment increases with the number of robots, the maximum value of the production, $P$, and the coverage radius. Moreover, since higher values of the $d$ produce lower decreases on the coverage level, the evolution with respect to this parameter is the same. An uncertainty on the value of the decay might lead to under or overcovering the points of the environment. In particular, it would lead to a certain value in Fig. 15(a) different from the actual value.

Note that we do not show the evolution with the communication radius since it is negligible. This happens due to the size of the Voronoi region with respect to the zero-error area and to the selection of the goals near the robot, as can be seen in Fig. 3(b).

We also evaluate the influence of all the parameters in the standard deviation of the coverage level. Fig. 16 shows the average standard deviation for each value of the parameters. We can draw the following conclusions:

1) Low values of the decay present a similar behavior, especially in the steady state. Only the greatest value shows a greater standard deviation for the transient and the stationary [see Fig. 16(a)].
2) The influence of the number of robot in the transient is rather complex [see Fig. 16(b)]. However, in the end, the more robots are used, the higher the standard deviation.
3) An increase in the maximum production makes the standard deviation of the coverage level also increase, since the increment of the coverage level at each time is higher

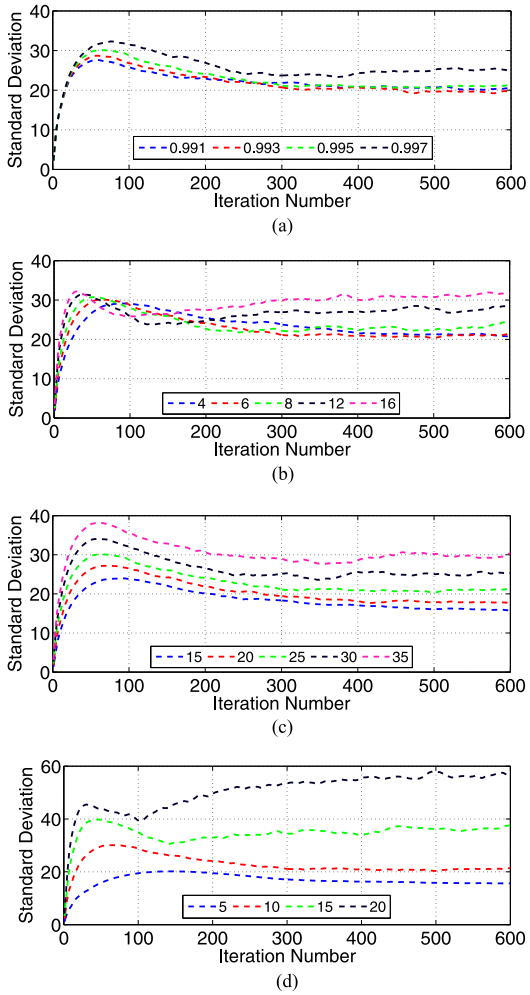



Fig. 16. Evolution of the average standard deviation of the coverage level for different values of the parameters. (a) d = 0.991, 0.993, 0.995, 0.997. (b) N = 4, 6, 8, 12, 16. (c) P = 15, 20, 25, 30, 35. (d) rcov = 5, 10, 15, 20.

and, therefore, the value of the most covered points with respect to the least covered ones [see Fig. 16(c)].

4) The deviation of the coverage level responds to the variation of the coverage radius in the same way as to $P$ [see Fig. 16(d)] and this parameter is again the one that produces the greatest variation.

Although, in most cases, the lowest value of each parameter leads to the lowest deviation, it is essential that the environment reaches the objective and, thus, the average coverage level of the environment from Table IV must be taken into account. It can be seen that the coverage level increases with all the parameters and that the optimal value according to this criterion is the same as for the coverage error.

## VII. Conclusion and Future Work

In this paper, we have presented a distributed approach to the persistent coverage problem with a group of mobile robots. We have formulated the problem in discrete time to deal with the discrete nature of the communications in a distributed system. We have presented an algorithm to obtain an estimation of the global coverage map with only local information received from neighboring robots. This algorithm has proven to be accurate when calculating the estimation. We have introduced a motion control law that drives the robots to poorly covered areas while moving in the direction of the maximum improvement. The first part is achieved by following goals that are selected inside the Voronoi regions to avoid conflicts between robots. The second part is achieved using the local gradient of the *Improvement Function*. Finally, we have carried out different simulations to validate the proposed contributions and to evaluate the performance of the system.

## Appendix A
## Proof of Lemma IV.1

In this proof, all the areas refer to time $k$ unless it is explicitly stated, for instance, $\mathbf{\Omega}_i^z \equiv \mathbf{\Omega}_i^z(k)$ and also $N_i \equiv N_i(k)$. In the first place, we prove the upper bound of the estimation. Following the principle of induction:

1) Consider $Z_i(0) \leq Z(0)$ from Assumption III.2.

2) Assume as an induction hypothesis that $Z_i(k-1) \leq Z(k-1)$, $\forall \mathbf{q} \in \mathbf{Q}, i \in \{1, \ldots, N\}$.

3) We split the environment into two areas in accordance to the estimation algorithm. If a point $\mathbf{q} \in \mathbf{\Omega}_i$, the update from (2), (3a), (5) and Assumption III.1 is

$$Z_i(k) = dZ_i(k-1) + \alpha_i(k) + \sum_{j \in N_i} \max\Big(dZ_j(k-1) + \alpha_j(k) - dZ_i(k-1), 0\Big)$$

where, according to the hypothesis of this lemma (7) and the induction hypothesis

$$0 \leq \max\Big(dZ_j(k-1) + \alpha_j(k) - dZ_i(k-1), 0\Big) \leq \alpha_j(k).$$

Therefore, we have

$$Z_i(k) \leq dZ_i(k-1) + \alpha_i(k) + \sum_{j \in N_i} \alpha_j(k) \leq Z(k).$$

In the second area, i.e., if $\mathbf{q} \notin \mathbf{\Omega}_i$, the update from (2), (3b), and (5) is

$$Z_i(k) = \max_{j \in N_i}\big(dZ_i(k-1), dZ_j(k-1) + \alpha_j(k)\big) - \max_{j \in N_i} \beta_j(k) + \sum_{j \in N_i} \beta_j(k) \tag{24}$$

where the first term

$$\max_{j \in N_i}\big(dZ_i(k-1), dZ_j(k-1) + \alpha_j(k)\big) \leq dZ(k-1) + \max_{j \in N_i} \alpha_j(k).$$

If such $\mathbf{q} \in \mathbf{\Omega}_j^o$, that is, $\beta_j(k) > 0$, then $\max_{j \in N_i} \alpha_j(k) = \max_{j \in N_i} \beta_j(k)$, $\sum_{j \in N_i} \beta_j(k) = \sum_{j \in N_i} \alpha_j(k)$ and (24) becomes

$$Z_i(k) \leq dZ(k-1) + \sum_{j \in N_i} \alpha_j(k) \leq Z(k). \tag{25}$$

Finally, if such $\mathbf{q} \notin \mathbf{\Omega}_j^o$, there exists no $\beta_j(k)$, (24) becomes

$$Z_i(k) \leq dZ(k-1) + \max_{j \in N_i} \alpha_j(k) \leq Z(k)$$

and the first result is proven.

For the second result of this lemma an analogous demonstration can be developed splitting the environment into $\mathbf{\Omega}_i$ and the rest of the zero-error area, $\mathbf{\Omega}_i^z \setminus \mathbf{\Omega}_i$, particularizing each region for $\mathbf{q} \in \mathbf{\Omega}_j$ and $\mathbf{q} \notin \mathbf{\Omega}_j$, and applying the hypothesis and the first result of this lemma. ■

## APPENDIX B
## PROOF OF LEMMA IV.2

Let $\mathbf{q} \in \mathbf{Q}$ be a point such that, for at least one robot, $j_1$, $\alpha_{j_1}(k-N+1) > 0$. In such case $\mathbf{q} \in \mathbf{\Omega}_{j_1}^r(k-N)$, by assumption of this lemma $Z_{j_1}(k-N) = Z(k-N)$ and from Lemma IV.1, $Z_{j_1}(k-N+1) = Z(k-N+1)$. In the following iterations, according to the property of the estimation from (6), it is clear that

$$\begin{aligned} Z_{j_1}(k-N+1+\ell) &\geq d^{\ell-1} Z(k-N+1) \\ &= d^{\ell} Z(k-N) + d^{\ell-1}\alpha(k-N+1) \end{aligned} \tag{26}$$

with $\ell \in \{1, \ldots, N-1\}$, implying that any possible error in the estimation of $j_1$ is not due to the productions at time $k-N+1$.

In the next iteration, $k-N+2$, for any neighbor of $j_1$, say $j_2$, it can happen that $\mathbf{q} \in \mathbf{\Omega}_{j_2}^r(k-N+1)$ or not. If $\mathbf{q} \notin \mathbf{\Omega}_{j_2}(k-N+2)$, from (3b) and (5)

$$\begin{aligned} &Z_{j_2}(k-N+2) \\ &\quad = \max_{m_2}(dZ_{j_2}(k-N+1), Z_{m_2}^{\text{com}}(k-N+2)) \\ &\qquad + \sum_{m_2} \beta_{m_2}(k-N+2) - \max_{m_2}\left(\beta_{m_2}(k-N+2)\right) \\ &\quad \geq dZ_{j_1}(k-N+1) \geq d^2 Z(k-N) + d\alpha(k-N+1) \end{aligned}$$

with $m_2 \in N_{j_2}(k-N+2)$. On the other hand, if $\mathbf{q} \in \mathbf{\Omega}_{j_2}(k-N+2)$, the update from (3a) and (5) results in

$$\begin{aligned} &Z_{j_2}(k-N+2) \\ &\geq dZ_{j_2}(k-N+1) + \alpha_{j_2}(k-N+2) \\ &\quad + \max\left(dZ_{j_1}(k-N+1) + \alpha_{j_1}(k-N+2)\right. \\ &\quad \left. - dZ_{j_2}(k-N+1), 0\right) \\ &\geq dZ_{j_1}(k-N+1) \geq d^2 Z(k-N) + d\alpha(k-N+1). \end{aligned}$$

Thus, $j_2$ has no estimation error at time $k-N+2$ caused by $\alpha(k-N+1)$. Additionally, using the same argument as in (26), it is clear that any error in $j_2$ in the following iterations is not due to $\alpha(k-N+1)$.

A similar reasoning can be developed at time $k-N+3$ for neighbors of $j_1$ or $j_2$ and, consequently, in at most $N-1$ iterations, the same result can be obtained for all the robots of the network. Therefore, the estimation of any robot of the network at time $k$ includes at least the productions of the robots at time $k-(N-1)$ and the proof is complete. ■

## APPENDIX C
## PROOF OF LEMMA IV.3

Since $\rho_i(k)$ is the minimum distance from the current position of robot $i$ to any of the last $N-1$ positions of other robots $j$ that are not neighbors of $i$ at iteration $k$, then $\alpha_j(\ell) = 0$ for all $\mathbf{q} \in \mathbf{\Omega}_i^{\rho}(k)$ and $\ell = \{k-N+1, \ldots, k\}$.

Therefore, we need to demonstrate that $\mathbf{\Omega}_i^r(k) \subseteq \mathbf{\Omega}_i^{\rho}(k)$. This happens if the radius of $\mathbf{\Omega}_i^r(k)$ is smaller than the radius of $\mathbf{\Omega}_i^{\rho}(k)$, that is, $r^{\text{com}} - r_{\max}^{\text{cov}} + u_i^{\max}(k) \leq \rho_i(k) - r_{\max}^{\text{cov}}$, which is true if (8) holds. ■

## APPENDIX D
## PROOF OF THEOREM IV.4

Following the principle of induction:

1) Consider $Z_i(0) = Z(0)$ from Assumption III.2.

2) Assume as an induction hypothesis that $Z_i(k-1) = Z(k-1) \quad \forall\, \mathbf{q} \in \mathbf{\Omega}_i^r(k-1)$.

3) Using Lemma IV.1 and the induction hypothesis, we have $Z_i(k) = Z(k)$ in $\mathbf{\Omega}_i^z(k)$. Then, for those areas of $\mathbf{\Omega}_i^r(k)$ that intersect with $\mathbf{\Omega}_i^z(k)$, the result is proven.

We now focus on demonstrating that the same holds for any point $\mathbf{q} \in \mathbf{\Omega}_i^r(k) \setminus \mathbf{\Omega}_i^z(k)$. Lemma IV.2 demonstrates that the production of the robots before or at time $k-N+1$ induces no error in $Z_i(k)$ and Lemma IV.3 guarantees that $\alpha_m(\ell) = 0$ in $\mathbf{\Omega}_i^r(k)$ for all $\ell \in \{k-N+1, \ldots, k\}$, $m \notin N_i(k)$. Therefore, only the production after $k-N+1$ of the current neighbors of robot $i$ can cause an error on $Z_i(k)$ within $\mathbf{\Omega}_i^r(k) \setminus \mathbf{\Omega}_i^z(k)$. For any point of this region, the updating law from (2), (3b), and (5) is (24).

If there exists some neighbor $j$ such that $\mathbf{q} \in \mathbf{\Omega}_j^z(k)$, then $Z_j(k) = Z(k)$ and the first term of (24) becomes $dZ(k-1) - \max_{j \in N_i(k)} \alpha_j(k)$. Since $\alpha_i(k) = 0$ and $\max_{j \in N_i(k)} \alpha_j(k) = \max_{j \in N_i(k)} \beta_j(k)$, then $Z_i(k) = Z(k)$ in $\left(\mathbf{\Omega}_i^r(k) \setminus \mathbf{\Omega}_i^z(k)\right) \cup \mathbf{\Omega}_j^z(k)$.

On the other hand, if there is no neighbor $j$ such that $\mathbf{q} \in \mathbf{\Omega}_j^z(k)$, i.e., $\alpha_j(k) = 0$ for all $j \in V$ in $\mathbf{q}$, then (24) results in $Z_i(k) = \max_{j \in N_i(k) \cup i} dZ_j(k-1)$.

Repeating the same procedure for time $k-1$, if $\mathbf{q} \in \mathbf{\Omega}_j^z(k-1)$ for any $j \in \{N_i(k), i\}$, then $Z_j(k-1) = Z(k-1)$ which implies that $Z_i(k) = dZ(k-1) = Z(k)$. If $\mathbf{q} \notin \mathbf{\Omega}_j^z(k-1)$, then we can repeat recursively until $k-(N-1)$. If $\mathbf{q} \notin \mathbf{\Omega}_j^z(\ell)$ with $\ell \in \{k-N+1, \ldots, k-1\}$, then, according to Lemma IV.2, $Z_i(k) = d^N Z(k-N) + d^{N-1}\alpha\big(k-(N-1)\big) = Z(k)$, completing the proof. ■

## APPENDIX E
## PROOF OF THEOREM IV.6

According to Lemma IV.2, whose hypothesis is confirmed by Theorem IV.4, the production of the robots from $N-1$ or more instants before does not induce an error. Therefore, it is guaranteed that the robot does not have an estimation error in an area around it if it is not capable of reaching the production of non-neighbors in less than $N-1$ iterations. According to Assumption III.1, in the worst case, the production of a non-neighbor is at a distance of $r^{\text{com}} - r_{\max}^{\text{cov}}$. Additionally, a robot can move at most $(N-1)u^{\max}$ units in $N-1$ instants. Therefore, the radius of the area around the

robot that cannot reach the previous productions of a non-neighbor is $r_z^* = r^{\text{com}} - r_{\max}^{\text{cov}} - (N-1)u^{\max}$ and the proof is complete. ■

## APPENDIX F
## PROOF OF THEOREM IV.7

The lower bound is proven in Lemma IV.1, whose hypothesis is confirmed by Theorem IV.4. We prove here the upper bound. As we mentioned, the error of the local estimation, $\varepsilon^{\text{est}} = Z(k) - Z_i(k)$, is caused by the lack of communication between all robots of the network. In fact, the more non-neighbors of robot $i$ overlap between them, the higher this error is for robot $i$ at each time instant. Consequently, we can formulate the maximum error produced at a particular time instant as $e = Pn_o$ where $0 < n_o \leq N-1$ is the number of overlapping robots outside the communication area of any other robot. In the worst case scenario, this error is repeated every time instant and added to the previous ones that have decayed over time. Therefore, we have $\varepsilon^{\text{est}} = e + de + d^2e + \cdots + d^{\delta-1}e = e(1-d^\delta)/(1-d)$, where $\delta - 1$ represents the number of iterations until the robot receives the information and its value depends on $n_o$. However, as introduced for Lemma IV.2, the maximum is $N-1$ instants, i.e., $\delta - 1 \leq N-1$, and only the errors of the last $N-1$ iterations have to be added. Thus

$$\varepsilon^{\text{est}} = Pn_o \frac{1-d^\delta}{1-d} \leq P(N-1)\frac{1-d^N}{1-d}$$

and the desired result is proven. ■

## APPENDIX G
## FORMULATION OF THE GRADIENT

In this appendix, we omit the dependencies with $\mathbf{q}$ for simplicity and we use $\mathbf{p}_i \equiv \mathbf{p}_i(k)$ and $\mathbf{\Omega}_i \equiv \mathbf{\Omega}_i(\mathbf{p}_i)$. Equation (19) comes from the derivation of the *Improvement Function* from (10) particularized for the position of the robot, $\mathbf{p}_i(k)$. To obtain (19), first we derive the quotient in (10)

$$\nabla_{\mathbf{p}_i} M_i(\mathbf{p}_i, k) = \frac{\nabla_{\mathbf{p}_i} B_1 \cdot B_2 - B_1 \cdot \nabla_{\mathbf{p}_i} B_2}{B_2^2} \tag{27}$$

with

$$B_1(k) = \int_{\mathbf{\Omega}_i} \frac{Z^* - Z_i}{Z^*}\, \Phi\, \alpha_i(k)\, \mathrm{d}\mathbf{q},\ B_2(k) = \int_{\mathbf{\Omega}_i} \Phi\, \alpha_i(k)\, \mathrm{d}\mathbf{q}.$$

From the estimation algorithm in (2), (3a), and (5) and (21), we have

$$Z_i(k) = dZ_i(k-1) + \alpha_i(k) + \sum_{j \in N_i} \alpha_j^i(k) \quad \forall\, \mathbf{q} \in \mathbf{\Omega}_i$$

and applying the Leibniz integral rule to the term $\nabla_{\mathbf{p}_i} B_1$, we obtain

$$\begin{aligned}\nabla_{\mathbf{p}_i} B_1 &= \int_{\mathbf{\Omega}_i} \frac{\Phi}{Z^*} \nabla_{\mathbf{p}_i} \Big( Z^* \alpha_i(\mathbf{p}_i) \\ &\quad - \big( dZ_i(k-1) + \alpha_i(\mathbf{p}_i) + \sum_{j \in N_i} \alpha_j^i(k) \big) \alpha_i(\mathbf{p}_i) \Big)\, \mathrm{d}\mathbf{q} \\ &= \int_{\mathbf{\Omega}_i} \frac{\Phi}{Z^*} \nabla_{\mathbf{p}_i} \alpha_i(\mathbf{p}_i) \big( Z^* - dZ_i(k-1) \\ &\quad - \sum_{j \in N_i} \alpha_j^i(k) - 2\alpha_i(\mathbf{p}_i) \big)\, \mathrm{d}\mathbf{q}. \end{aligned} \tag{28}$$

On the other hand, we have $\nabla_{\mathbf{p}_i} B_2 = \int_{\mathbf{\Omega}_i} \Phi \nabla_{\mathbf{p}_i} \alpha_i(\mathbf{p}_i)\, \mathrm{d}\mathbf{q}$. Introducing this equation and (28) in (27) and denoting by $A(k)$ the expression in (20), the gradient of the *Improvement Function* results in (19). ■

## REFERENCES


[1] D. Mackenzie and T. Balch, "Making a clean sweep: Behavior based vacuuming," in *Proc. AAAI Fall Symp.: Instantiating Real-world Agents*, 1993, pp. 93–98.

[2] N. M. Kakalis and Y. Ventikos, "Robotic swarm concept for efficient oil spill confrontation," *J. Hazardous Mater.*, vol. 154, no. 1, pp. 880–887, 2008.

[3] E. M. Arkin, S. P. Fekete, and J. S. Mitchell, "Approximation algorithms for lawn mowing and milling," *Comput. Geom.*, vol. 17, no. 1, pp. 25–50, 2000.

[4] R. N. Smith, M. Schwager, S. L. Smith, B. H. Jones, D. Rus, and G. S. Sukhatme, "Persistent ocean monitoring with underwater gliders: adapting sampling resolution," *J. Field Robot.*, vol. 28, no. 5, pp. 714–741, 2011.

[5] D. A. Paley, F. Zhang, and N. E. Leonard, "Cooperative control for ocean sampling: the glider coordinated control system," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 4, pp. 735–744, Jul. 2008.

[6] N. Nigam, "The multiple unmanned air vehicle persistent surveillance problem: A review," *Machines*, vol. 2, no. 1, pp. 13–72, 2014.

[7] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.

[8] M. Schwager, D. Rus, and J. Slotine, "Decentralized, adaptive coverage control for networked robots," *Int. J. Robot. Res.*, vol. 28, no. 3, pp. 357–375, 2009.

[9] H. W. Hamacher and Z. Drezner, *Facility Location: Applications and Theory*. New York, NY, USA: Springer, 2002.

[10] I. I. Hussein and D. M. Stipanovic, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 4, pp. 642–657, Jul. 2007.

[11] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 376–386, Jun. 2005.

[12] F. Pasqualetti, J. W. Durham, and F. Bullo, "Cooperative patrolling via weighted tours: Performance analysis and distributed algorithms," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1181–1188, Oct. 2012.

[13] D. Portugal, C. Pippin, R. P. Rocha, and H. Christensen, "Finding optimal routes for multi-robot patrolling in generic graphs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 363–369.

[14] A. Marino, G. Antonelli, A. P. Aguiar, and A. Pascoal, "A new approach to multi-robot harbour patrolling: theory and experiments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2012, pp. 1760–1765.

[15] N. Agmon, S. Kraus, and G. A. Kaminka, "Multi-robot perimeter patrol in adversarial settings," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 2339–2345.

[16] P. F. Hokayem, D. Stipanovic, and M. W. Spong, "On persistent coverage control," in *Proc. IEEE Conf. Dec. Control*, 2007, pp. 6130–6135.

[17] C. Song, L. Liu, G. Feng, Y. Wang, and Q. Gao, "Persistent awareness coverage control for mobile sensor networks," *Automatica*, vol. 49, no. 6, pp. 1867–1873, 2013.

[18] C. G. Cassandras, X. Lin, and X. Ding, "An optimal control approach to the multi-agent persistent monitoring problem," *IEEE Trans. Autom. Control*, vol. 58, no. 4, pp. 947–961, Apr. 2013.
[19] J. M. Palacios-Gasos, E. Montijano, C. Sagüés, and S. Llorente, "Multi-robot persistent coverage using branch and bound," in *Proc. Am. Control Conf.*, 2016, pp. 5697–5702.
[20] N. Nigam, S. Bieniawski, I. Kroo, and J. Vian, "Control of multiple UAVs for persistent surveillance: Algorithm and flight test results," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 5, pp. 1236–1251, Sep. 2012.
[21] S. L. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: monitoring and sweeping in changing environments," *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 410–426, Apr. 2012.
[22] X. Lan and M. Schwager, "Planning periodic persistent monitoring trajectories for sensing robots in gaussian random fields," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 2407–2412.
[23] D. E. Soltero, M. Schwager, and D. Rus, "Decentralized path planning for coverage tasks using gradient descent adaptive control," *Int. J. Robot. Res.*, vol. 33, no. 3, pp. 401–425, 2014.
[24] X. Lin and C. G. Cassandras, "An optimal control approach to the multi-agent persistent monitoring problem in two-dimensional spaces," in *Proc. IEEE Conf. Decis. Control*, 2013, pp. 6886–6891.
[25] S. Alamdari, E. Fata, and S. L. Smith, "Persistent monitoring in discrete environments: Minimizing the maximum weighted latency between observations," *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 138–154, 2013.
[26] C. Franco, G. Lopez-Nicolas, C. Sagues, and S. Llorente, "Persistent coverage control with variable coverage action in multi-robot environment," in *Proc. IEEE Int. Conf. Decis. Control*, 2013, pp. 6055–6060.
[27] N. Hubel, S. Hirche, A. Gusrialdi, T. Hatanaka, M. Fujita, and O. Sawodny, "Coverage control with information decay in dynamic environments," in *Proc. 17th IFAC World Congr.*, 2008, pp. 4180–4185.
[28] R. Graham and J. Cortés, "Adaptive information collection by robotic sensor networks for spatial estimation," *IEEE Trans. Autom. Control*, vol. 57, no. 6, pp. 1404–1419, Jun. 2012.
[29] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Trans. Robot.*, vol. 24, no. 3, pp. 710–724, Jun. 2008.
[30] R. A. Freeman, P. Yang, and K. M. Lynch, "Stability and convergence properties of dynamic average consensus estimators," in *Proc. IEEE Int. Conf. Decis. Control*, 2006, pp. 398–403.
[31] S. Martínez, "Distributed interpolation schemes for field estimation by mobile sensor networks," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 491–500, Mar. 2010.
[32] K. J. Aström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton, NJ, USA: Princeton Univ. Press, 2010.
[33] J. M. Palacios-Gasos, E. Montijano, C. Sagues, and S. Llorente, "Distributed coverage estimation for multi-robot persistent tasks," in *Proc. Eur. Control Conf.*, Jul. 2015, pp. 3681–3686.
[34] L. Sabattini, A. Gasparri, C. Secchi, and N. Chopra, "Enhanced connectivity maintenance for multi-robot systems," in *Proc. 10th Int. IFAC Symp. Robot Control*, 2012, pp. 319–324.
[35] M. Zavlanos and G. Pappas, "Distributed connectivity control of mobile networks," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1416–1428, Dec. 2008.
[36] M. M. Zavlanos, "Synchronous rendezvous of very-low-range wireless agents," in *Proc. Int. Conf. Decis. Control*, 2010, pp. 4740–4745.
[37] G. Hollinger and S. Singh, "Multi-robot coordination with periodic connectivity," in *Proc. Int. Conf. Robot. Autom.*, 2010, pp. 4457–4462.
[38] N. A. Lynch, *Distributed Algorithms*. San Mateo, CA, USA: Morgan Kaufmann, 1996.
[39] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.
[40] L. Jaillet, A. Yershova, S. La Valle, and T. Simeon, "Adaptive tuning of the sampling domain for dynamic-domain RRTs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2005, pp. 2851–2856.
[41] F. Bullo, J. Cortés, and S. Martinez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton, NJ, USA: Princeton Univ. Press, 2009.
[42] L. Pimenta, V. Kumar, R. Mesquita, and G. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *Proc. IEEE Conf. Decis. Control*, 2008, pp. 3947–3952.
[43] A. Solanas and M. Garcia, "Coordinated multi-robot exploration through unsupervised clustering of unknown space," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2004, vol. 1, pp. 717–721.
[44] N. Michael, M. Zavlanos, V. Kumar, and G. J. Pappas, "Distributed multi-robot task assignment and formation control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 128–133.

**José Manuel Palacios-Gasós** (S'16) received the B.S. degree in industrial engineering in 2012 and the M.S. degree in systems engineering and computer science in 2014 from University of Zaragoza, Zaragoza, Spain, where he is currently working toward the Ph.D. degree with the Instituto de Investigación en Ingeniería de Aragón, Zaragoza, Spain.

His research interests include multiagent systems, distributed and cooperative control, and computer vision.

**Eduardo Montijano** (M'12) received the M.Sc. and Ph.D. degrees from Universidad de Zaragoza, Zaragoza, Spain, in 2008 and 2012, respectively.

He has been a Visiting Scholar with University of California San Diego, University of California Berkeley, and Boston University in the United States and with the Royal Institute of Technology, Stockholm, Sweden. He is currently a Professor with Centro Universitario de la Defensa, Zaragoza, Spain. His main research interests include distributed algorithms, cooperative control, and computer vision.

Dr. Montijano's Ph.D. dissertation received the extraordinary award from Universidad de Zaragoza during the 2012–2013 academic year.

**Carlos Sagüés** (M'00–SM'11) received the M.Sc. and Ph.D. degrees from Universidad de Zaragoza, Zaragoza, Spain.

During the course of the Ph.D. degree, he worked on force and infrared sensors for robots. Since 1994, he has been an Associate Professor and, since 2009, a Full Professor with the Departamento de Informatica e Ingeniería de Sistemas, Universidad de Zaragoza, where he has also been a Head Teacher. His research interest includes control systems, computer vision, visual robot navigation, and multivehicle cooperative control.

**Sergio Llorente** received the M.S. and Ph.D. degrees in electronics engineering from University of Zaragoza, Zaragoza, Spain, in 2001 and 2016, respectively.

In 2001, he joined the BSH Home Appliances Group, Zaragoza, Spain, where he has held different positions in the Research and Development Department of Induction Cooktops. He is currently in charge of several research lines and is an Inventor in more than 150 patents. He has also been an Assistant Professor with University of Zaragoza since 2004. His research interests include power electronics, simulation and control algorithms for power electronics, and temperature control.