Distributed control of mobile robots in an environment with static obstacles

Konstantinos Giannousakis | Anthony Tzes

1Electrical & Computer Engineering Department, University of Patras, Rio, Greece
2Electrical & Computer Engineering Program, New York University Abu Dhabi, Abu Dhabi, United Arab Emirates

Abstract
This study addresses the problem of deploying a group of mobile robots over a non-convex region with obstacles. Assuming that the robots are equipped with omnidirectional range sensors of common radius, disjoint subsets of the sensed area are assigned to the robots. These proximity-based subsets are calculated using the visibility notion, where the cell of each robot is treated as an opaque obstacle for the other robots. Based on that, optimal spatially distributed coordination algorithms are derived for the area coverage problem and for the homing problem, where the swarm needs to move to specific locations. Experimental studies demonstrate the results.

1 | INTRODUCTION

The coordination of autonomous vehicles is a field that has known a huge growth over the last few years. This is of no surprise as the idea of deploying a decentralised group of robots, working in collaboration by distributing information, is very enticing and has many applications. Some of the most popular tasks for swarm robotics are: moving while preserving a formation [1–3], exploration and mapping of an unknown region [4–7], and foraging [8–10], where the robots have to find and transport objects to specific locations.

One of the most basic and interesting tasks in collaborative robotic networks is the area coverage. In area coverage, the robots are equipped with sensors and are assigned to sweep across a region of interest in a way that maximises their monitoring ability. Extensive work exists in the literature considering convex regions [11–13] as well as non-convex [14–17] regions. The definitive step in most of these methods is to determine what parts of the region are assigned to each robot, while there are also methods such as the Virtual Forces [18] that rely only on virtual interactions between the nodes to evenly distribute them. In convex domains, the commonly used method for determining the territory of each robot is the Voronoi tessellation [11, 12], whereas for non-convex domains, various methods have emerged.

One of these methods is the geodesic diagram, which uses the geodesic distance to create cells that follow the Voronoi or power diagram notion. The authors in [17, 19] use the limited geodesic diagram based on the positions of the robots, to build a distributed control law that maximises the total assigned area. On the other hand, the authors in [20] use static points to generate the geodesic power diagram of the whole region and modify the weights depending on the capabilities of the robots to make the partitions equitable.

Another way to deal with the non-convexity of the region is to discretise it and use graph representation to denote the connections between the neighbouring discrete points. The authors in [21] use this concept to create a generalised partitioning method, while ensuring that the partitions are connected. They determine that the optimal partitioning is also a centroidal Voronoi partition. Another method of partitioning the whole discretised region is presented in [22], where the main concept is to divide the region into cells of equal area.

The method presented in this article uses the visibility notion to assign cells to the robots. The visibility-based approach is more realistic as the responsibility of the robots is determined by the area that they can sense. Additionally, the a priori knowledge of the region is not necessary, making it suitable for exploration as well. Naturally, using only the visible area means that the assigned cells are not a partition of the...
total region and depending on the requirements, they may not even utilise the total sensed area. The authors in [23], tried to exploit the whole sensed area using a strictly proximity-based assignment of the visible region, with the drawback that the resulting cells consist of disjoint parts. Apart from violating the conditions of the extended Leibniz integral rule [24], which is required to determine the distributed control law, allowing a robot to move towards these disjoint parts means that the robot may step into the territory of another robot. This does not sit well with the notion of cell assignment and does not guarantee collision avoidance of the agents. The authors in [16], to avoid that, added the restriction of connected cells by completely discarding the disjoint parts, without trying to assign them to a different node. Here, a novel visibility-based approach is utilised, which ensures connected cells, while trying to make use of as much of the sensed area as possible.

Apart from area coverage, another useful task is the homing of the swarm, where the team of robots should acquire specific locations, and possibly orientations, in the region of interest. Homing is one of the subroutines in foraging [10], and is also a task that has been examined for single robots [25, 26], but not as an explicit task that considers the issues that surface in a multi-robot environment. The issues that the homing problem has to face concern: the optimal target assignment, how to prevent robot collisions and how to guarantee that no robot will block another. The proposed scheme in this article addresses these issues in a simple collaborative manner.

The focus of this article lies into the realistic application of the existing theory in practice. The region of interest is a general, non-convex, connected polygon with obstacles, thus a visibility-based approach has been utilised to determine which parts correspond to the vicinity of each robot. This proximity-based cell assignment maintains visibility accurate connected cells by treating the neighbouring cells as obstacles, while trying to maximise the total assigned area. The advantage of these cells is that every location of their interior is reachable in a straight line from the respective robot. This allows the robots to move freely in their cell and at the same time prevents collisions with other robots. Additionally, while most of the works in the literature consider dimensionless robots with simplified dynamics [13–15, 17, 19], this cell assignment enables us to directly utilise the motion controller of our previous work [23], to take into account the dimensions and the full model of the differential drive robot.

Furthermore, the notion of objective functions is employed to determine the optimal control inputs for the robots. An aggregate objective function [11] is defined for each problem, in a way that maximising it implies optimal placement of the robotic agents. The distributed nature of the problem lies on dividing this aggregate objective function into individual subproblems over each robot, where after iteratively optimising them, the aggregate objective function will converge on a local maximum. Additionally, the distributed concept lies on requiring only local knowledge, such as the locations of spatial neighbours, to form the individual objective functions, so that the robots can move autonomously with minimal communication requirements.

For the area coverage problem, the same procedure found in the literature is followed, while utilising the aforementioned visibility-based cell assignment and relying on our previous work [27] for obstacle avoidance. The resulting optimal control scheme is then evaluated by experimental studies. Subsequently, the homing problem is examined for a swarm of robots, where a simple target assignment method is devised and used in conjunction with the visibility cells. After assigning the targets, the optimal movement direction for each robot is determined by minimising the distance to its target. This scheme is tested in experimental studies as well.

The rest of the article has the following structure. In Section 2, the main assumptions are defined along with the model of the robots and their motion controller. Additionally, the problems of area coverage and homing are introduced. Section 3 defines how the region of interest is split into visibility-based cells, which are then assigned to the robots. In Section 4, the area coverage problem is analysed and the optimal movement direction for the robots is formulated for the general case and then for some specific cases. In Section 5, the homing problem is examined, where the target assignment and movement direction of the robots are determined. Finally, experimental results are presented in Section 6, followed by some concluding remarks in Section 7. The most important symbols of the article are summarised in the Appendix.

2 PROBLEM FORMULATION

Let $\Omega \subset \mathbb{R}^2$ be a simple compact and connected non-convex polygonal region with holes, which will be our domain of interest. In the region there are $n \in \mathbb{Z}$ robotic agents, whose positions will be denoted as $p_i \in \Omega$, $i \in I_n$, with $I_n = \{1, \ldots, n\}$. Additionally, we define $P = \{p_1, \ldots, p_n\}$ as the set of all the distinct robot positions.

The agents are assumed to be non-holonomic mobile robots with differential drive, as shown in Figure 1. Concerning their dimensions, it is presumed that they are of a circular shape of radius $R \in \mathbb{R}^+$, centred at their rotation centre, and that they are small enough to be able to traverse the narrowest parts of the region. Moreover, the robots are equipped with visibility-based omnidirectional range sensors, such as cameras or LIDARs, and their range is limited in a radius $r_d > 2R$ around the centre of the robots. If we define the unlimited visibility space of each robot as

$$V_i = \{q \in \Omega : p_i + \lambda(q - p_i) \in \Omega, \forall \lambda \in [0, 1]\},$$

then the limited range visibility space simply takes the form

$$V_{i,r} = V_i \cap B(p_i, r_d),$$

where $B(p_i, r_d)$ is the ball of radius $r_d$, centred at $p_i$.

Apart from the visibility space, we also need to define the distance of the visible obstacles around the robot, in terms of.
the visibility angle $\theta \in (-\pi, \pi]$. For that to end, we define the visibility field as

$$
d_{f,i}(\theta; W_i) = \begin{cases} 
\|q_0 - p_i\|, & \text{if } \exists q_0 \in \partial W_i : \angle(q_0 - p_i) = \theta \\
\infty, & \text{if } \nexists q_0
\end{cases}
$$

where $W_i \subseteq V_i$ is the part of the visibility space that the robot is allowed to operate and $\partial W_i$ is its boundary. The case $\nexists q_0$ refers to the unobstructed visibility angles, where $W_i$ is unbounded. Note that in the presence of obstacles, this distance is not continuous with respect to $\theta$. If we denote

$$
d_{f,i}(\theta; W_i) = \lim_{\theta \to \theta_0^+} d_{f,i}(\theta'; W_i),
$$

and $d_{f,i}(\theta_0^+; W_i) \neq d_{f,i}(\theta_b^+; W_i)$ if an obstacle appears at the visibility angle $\theta_b \in (-\pi, \pi]$. Even though the kinodynamic model of the robots is not required for the derivation of the control schemes, the obstacle avoidance is employed implicitly by the motion controller of our previous work [27], which in turn, relies on the model of the differential drive robot. Therefore, the position and the orientation of each robot are related to its linear and angular velocities through the kinodynamic equations

$$
\begin{bmatrix}
\dot{p}_{i}(t) \\
\dot{\theta}_{i}(t)
\end{bmatrix} = \begin{bmatrix}
\cos(\theta_{i}(t)) & 0 \\
\sin(\theta_{i}(t)) & 0
\end{bmatrix} \begin{bmatrix}
\nu_{i}(t) \\
\omega_{i}(t)\end{bmatrix},
\tag{1}
$$

where $\tau \in \mathbb{R}_+$ is the time instance, $p_{i}(t)$ is the location of its rotation centre, $\theta_{i}(t) \in (-\pi, \pi]$ is its orientation and $\nu_{i}(t), \omega_{i}(t) \in \mathbb{R}$ are its linear and angular velocities, respectively. The kinematic constraints can be expressed as follows:

$$
\begin{align*}
\nu_{\text{min}} \leq & \nu_{i}(t) \leq \nu_{\text{max}}, \\
\omega_{\text{min}} \leq & |\omega_{i}(t)| \leq \omega_{\text{max}}, \\
|\nu_{i}(t)| \leq & a_{\nu}, \text{ for } \nu_{i}(t) \nu_{i}(t) \geq 0, \\
|\omega_{i}(t)| \leq & a_{\omega}, \text{ for } \omega_{i}(t) \omega_{i}(t) < 0, \\
|\nu_{i}(t)| \leq & a_{\nu}, \text{ for } \nu_{i}(t) \nu_{i}(t) \geq 0, \\
|\omega_{i}(t)| \leq & a_{\omega}, \text{ for } \omega_{i}(t) \omega_{i}(t) < 0,
\end{align*}
$$

where $\nu_{\text{min}} \in \mathbb{R}_{\text{min}}^+, \nu_{\text{max}} \in \mathbb{R}_{\text{max}}^+$ and $\omega_{\text{min}} \in \mathbb{R}_{\text{max}}^+$ are the linear and angular velocity bounds, $a_{\nu}, a_{\omega}, a_{\nu} \in \mathbb{R}_{\text{max}}^+$ are the linear and angular accelerations of each robot, while $a_{\nu}, a_{\omega}, a_{\nu} \in \mathbb{R}_{\text{max}}^+$ are the linear and angular deceleration and acceleration limits. Finally, if $\nu_{\text{min}} \in \mathbb{R}_{\text{min}}^+$ is the maximum wheel velocity supported by the motors and $L \in \mathbb{R}_{\text{max}}^+$ is the wheel to wheel distance, then the differential drive robot has the additional constraint $|\omega_{i}| \leq 2(\nu_{\text{min}} - |\nu_{i}|)/L$ between its velocities.

The control input of each robot is assumed to be its reference linear and angular velocities. Thus, each robot has an internal velocity controller

$$
\begin{bmatrix}
\nu_{i}(t) \\
\omega_{i}(t)
\end{bmatrix} = C_{\nu}(\nu_{i}(t), \omega_{i}(t))
$$

to drive the motors, which naturally depends on all the robot constraints. The only elements that concern us about the internal velocity controller $C_{\nu}$ are, firstly, that our motion controller should provide the reference velocities $\nu_{i}(t), \omega_{i}(t) \in \mathbb{R}$, and secondly, that the actual velocities are part of the state vector of the robot. Consequently, our motion controller, which has a time interval $T_i \in \mathbb{R}_{\text{max}}^+$, is assumed to have access to the full state of the robot at each iteration $k \in \mathbb{N}_{\text{max}}$. In combination with Eq. (1), the full state of the robot is defined as $x_i = [p_{i}^T \theta_{i} \nu_{i} \omega_{i}]^T$, where the time parameter $t = kT_i$ is omitted for simplicity, and will also be omitted wherever it is not necessary. Here, a position error is allowed, but should be bounded in a disk of radius $r_a \in \mathbb{R}_{\text{max}}^+$.

As aforementioned, the obstacle avoidance of the robots is handled by the motion controller of our previous work [27]. We use this controller, with slight modifications, to perform direction-based control of the robots, while considering their dimensions and the obstacles around them, along with their dynamics and constraints. The controller here takes the form

$$
\begin{bmatrix}
\nu_{i}(t) \\
\omega_{i}(t)
\end{bmatrix} = M(u; x_i, W_i),
\tag{2}
$$

where $W_i$ is the region that the robot is allowed to move and $u_i$ is the desired movement direction. Treating the bounds of $W_i$ as the actual visible obstacles, the controller transforms the desired movement direction into the proper velocity inputs for the robot, $\nu_{i}, \omega_{i}$. The implementation modifications are to select $p_{i} = p_{i} \pm u_i$ as the final target, and completely skip the computation of the shortest path and select
\[ p_{i,1} = p_{i,1} = \arg \max_{\lambda \in [0,1]} \lambda \]

which is the furthest allowed point in the given direction.

Both the area coverage problem and the homing problem will be expressed in the form of an aggregate objective function. This function will be denoted by \( H(P(t)) \), as its value depends on the positions of the swarm. Its maximisation implies the optimal placement of the robotic agents; thus the optimal control law for the robots is to simply \( u_i = \partial H(P)/\partial p_i \). Therefore, the aim of the following sections is to determine this quantity for each robot independently.

In the area coverage problem, the objective function is formulated based on two components [12]. A density function \( \phi : \Omega \to \mathbb{R}_+ \), which is a measurable function representing the importance of the points in \( \Omega \). A performance function \( f : \mathbb{R}_+ \to \mathbb{R} \), which is a non-increasing and piecewise differentiable function that describes the sensing performance of the sensors. Naturally, the latter goes hand in hand with the selection of the distance function \( d : \Omega \to \mathbb{R}_+ \), which expresses the distance between the points of the region and the robots, with respect to the capabilities of the sensors. Thus, the aggregate objective function to be maximised, takes the form

\[
H(P) = \int_{\Omega} \max f(d(q,p_i)) \phi(q) dA,
\]

where \( \Omega \ni q = [x \ y]^T \) are the points of the region under integration and \( dA = dxdy \) is the infinitesimal area at \( q \).

A problem that arose in the experimentation with multiple robots is the initialisation of their position and orientation in the region of interest. To evaluate the behaviour of a distributed control scheme, such as the area coverage problem, multiple experiments should be performed using the same initial configuration. Even though we examined position control of a single robot inside a region with obstacles in our previous work [27], the simultaneous positioning of multiple robots proved to be challenging.

We define the homing problem for a group of robots, where the robots need to acquire specific locations in the region, but with no predefined assignment. Let \( Z_i \in \Omega \) with \( j \in I_\neq \) be the target locations and \( Z = \{Z_1, \ldots, Z_n\} \) be the set of all the distinct target points. Even though it is not required, for simplicity, we assume that these points are reachable by the robots, that is, their distance from the boundary of the region is at least \( R \). Let \( g : P \to Z \) be a bijective function expressing the robot-to-target assignment, then the objective function under maximisation is chosen as follows

\[
H(P) = \frac{1}{2} \sum_{i=1}^{n} d^2(P_i, g(P_i; P, Z)),
\]

where \( d_i : \Omega \times \Omega \to \mathbb{R}_+ \) is the length of the shortest path between two points in the region.

Even though in the area coverage problem, the knowledge of the sensed area of the robots is sufficient for them to move, in the homing problem, it is not enough. The computation of the shortest path [28] between the robots and their targets assumes a more comprehensive knowledge of the region. Clearly, a solution to that matter would be to use an Simultaneous Localization And Mapping algorithm [29] to map the region and assume that all unexplored area is free, but this is out of the scope of this article, which emphasises on the assignment of the targets and the optimal control law to solve the aggregate problem. For that reason, the region is presumed to be known and, additionally, the visibility of the robots is not limited, or equivalently, \( r_d = \infty \).

3 | VISIBILITY-BASED CELL ASSIGNMENT

A visibility-based approach is taken to properly deal with the proximity of the region points. First of all, we define the visibility distance function as

\[
d_V(q, p_i) = \begin{cases} 
\|q - p_i\| & \text{when } q \in V_i \\
\infty & \text{otherwise}.
\end{cases}
\]

Based on that, we define the relative visibility distance function as

\[
d_{RV}(q, p_i, P) = \begin{cases} 
\|q - p_i\| & \text{if } p_i \in P \text{ and } q \in V_{R,i}(P') \\
\infty & \text{otherwise},
\end{cases}
\]

where \( P' \subseteq P \) and

\[
V_{R,i}(P') = \left\{ q \in \Omega : \Omega \ni q \in P', d_V(q, p_i) \leq d_{RV}(q, p_i, P') \wedge \lambda \in [0,1], \forall p_j \in P \setminus \{p_i\} \right\}
\]

will be called the relative visibility cell of robot \( i \). Finally, the \( r \)-limited relative visibility cell and the respective \( r \)-limited relative visibility distance can be simply expressed as

\[
V_{R,i,r}(P) = V_{R,i}(P) \cap B(p_i, r),
\]

\[
d_{LRV}(q, p_i, r, P) = \begin{cases} 
\|q - p_i\| & \text{when } q \in V_{R,i,r}(P) \\
\infty & \text{otherwise}.
\end{cases}
\]

The distance \( d_{RV}(q, p_i, P) \) and the cell \( V_{R,i}(P) \) are inseparable by definition and they are characterised by a recursion. For a point \( q \in \Omega \) to be part of \( V_{R,i}(P) \), firstly it has to be part of \( V_n \) or equivalently, \( q_i \in \Omega \), \( \forall \lambda \in [0,1] \). Moreover, \( d_{RV}(q_i, p_i, P) \leq d_{RV}(q, p_i, P \setminus \{p_i\}) \) requires that all points in the line segment from \( p_i \) to \( q \) are closer to robot \( i \). This will filter out any disjoint parts ensuring that the visibility property is
satisfied for the cell of each robot, where a line segment from the robot to the boundary of the cell should be part of its interior. Furthermore, the recursion \( d_{RV}(q; p_i, p_j \setminus \{p_j\}) \), can be perceived as allowing the other nodes to form their cells in the absence of node \( p_j \) and then determining which parts of these cells actually belong to node \( p_i \). This allows the selection of points filtered out from other nodes, where the simple inequality \( d_V(q, p_i) \leq d_V(q, p_j) \) does not hold, as the cell of node \( p_j \) may be already limited by the presence of another node.

An example of the relative visibility cell assignment is shown in Figure 2b. By comparing it with the visibility shown in Figure 2a and the simple Euclidean Voronoi shown in Figure 2c, one can see the existence of points in \( V_{R,1} \) where \( d_V(q, p_i) > d_V(q, p_j) \). Additionally, it is clear that, in general, this cell assignment is not a partition of the total visible space. All the parts that would violate the visibility property have been discarded, forming a cell where all points in the line segment from its boundary to the robot are part of its interior. This cell is ideal for our motion controller, \( M \), which utilises the visibility property to incorporate obstacle avoidance. Moreover, it is worth mentioning that for convex regions, this cell assignment is identical to the Voronoi partitioning. Regarding the region, an interesting peculiarity that arises from the existence of holes is that two nodes may be neighbours from more than one path. This imposes a difficulty in expressing the neighbouring information with graph theory.

4 | AREA COVERAGE

To exploit the relative visibility cells in the coverage problem, in the objective function (3), we will utilise the respective distance function \( d_{LRV} \). Given that \( f \) is non-increasing and by the definition of \( V_{R,i}(P) \) and thus \( V_{R,i,j}(P) \), the aggregate objective function can be decoupled [11] and assumes the form

\[
\mathcal{H}(P) = \sum_{i \in I_i} \iint_{V_{R,i,j}(P)} f(||q - p_i||) \phi(q) \, dA.
\]

Since \( V_{R,i,j}(P) \) is a connected region, we can apply the extended Leibniz integral rule [24, 30] to find out how the objective function is modified with respect to the changes of \( p_i \) coordinates

\[
\frac{\partial \mathcal{H}}{\partial p_i}(P) = \iint_{V_{R,i,j}(P)} \left( \frac{\partial}{\partial p_i} f(||q - p_i||) \right) \phi(q) \, dA + \int_{\partial V_{R,i,j}(P)} f(||q - p_i||) \phi(q) \left( \frac{\partial q}{\partial p_i} \cdot \hat{n}(q) \right) \, dl + \sum_{j \in \mathcal{I}_i \setminus \{i\}} \int_{C_{i,j}} f(||q - p_i||) \phi(q) \left( \frac{\partial q}{\partial p_i} \cdot \hat{n}(q) \right) \, dl.
\]

The notation \( \partial V_{R,i,j} \) refers to the boundary of the cell, \( dl \) is the infinitesimal length and \( \hat{n}(q) \) is the outward normal vector to the curve at the point under integration \( q \), while \( \left( \frac{\partial q}{\partial p_i} \cdot \hat{n}(q) \right) \) is the velocity of the curve at that point in direction \( \hat{n}(q) \). Finally, \( C_{i,j} = \partial V_{R,i}(P) \cap \partial V_{R,j}(P) \) are the points where the cell of node \( j \) meets the cell of node \( i \).

To compute the aforementioned equation, it is required to express the velocity of the curves at their various parts. For that purpose, the boundary of each cell is broken down into four groups of disjoint curves

\[
\partial V_{R,i,j}(P) = C_{i,j}^0 \cup C_{i,j}^c \cup C_{i,j}^0 \cup C_{i,j}^c,
\]

which are analysed separately.

- \( C_{i,j}^0 = \partial V_{R,i} \cap \partial \Omega \): Visible boundary of the actual obstacles. These points are static, thus

\[
\frac{\partial q}{\partial p_i} \bigg|_{q \in C_{i,j}^0} = \emptyset \times 2.
\]

- \( C_{i,j}^c = \sum_{e \in \mathcal{E}_{i,j}} C_{i,j}^e \): The common line segments between the cells of nodes \( i, j \in I_n, i \neq j \). Obviously, \( C_{i,j}^e = \emptyset \) when the nodes are not neighbours. Also note that two nodes may share more than one bounding line segment, as the presence of holes

\[
\frac{\partial q}{\partial p_i} \bigg|_{q \in C_{i,j}^c} = \emptyset \times 2.
\]

Figure 2 Comparison for the node cell assignment. (a) Visibility, (b) Relative visibility, (c) Euclidean Voronoi

| (a) | (b) | (c) |
allows more than one neighbouring paths. The important thing here [30] is that $C_{ij} = C_{ji}^{g}$ and

$$\hat{n}_i^g(q) = -\hat{n}_i^{g*}(q), \forall q \in C_{ij}^{g}.$$ (6)

- $C_i^f = \bigcup_{k=1}^{n_{r,k}=n_{e,k}} C_{i,k}^f$: Line segments of obscured areas. The best way to describe these segments is as a discontinuity in the field of view. The $n_{r,k} \in \mathbb{Z}_+$ critical field-of-view angles are the $\theta_k \in (-\pi, \pi)$: $d_{ij}^f(\theta_k, V_{r,i,j}(P)) \neq d_{ij}^f(\theta_k, V_{r,j,i}(P))$, with $k \in \{1, \ldots, n_{r,k}\}$. In that way we can define these segments as

$$C_{i,k}^f = \{q_{i,k}(\lambda) = a_{i,k} + \lambda(b_{i,k} - a_{i,k}), \forall \lambda \in [0, 1]\},$$

with

$$a_{i,k} = p_i + \min \left( d_{ij}^f(\theta_k, V_{r,i,j}), d_{ij}^f(\theta_k, V_{r,j,i}) \right) \subset \partial \mathcal{B}(p_i, r_d)$$

$$b_{i,k} = p_i + \max \left( d_{ij}^f(\theta_k, V_{r,i,j}), d_{ij}^f(\theta_k, V_{r,j,i}) \right) \subset \partial \mathcal{B}(p_i, r_d).$$

In this case [15] we have $dl = d\lambda ||b_{i,k} - a_{i,k}||$ and

$$\frac{\partial q}{\partial p_i} \cdot \hat{n}_{i,k}^f = -\lambda \frac{||b_{i,k} - a_{i,k}||}{||b_{i,k} - a_{i,k}||} \cdot \hat{n}_{i,k}^f, \forall q \in C_{i,k}^f.$$

- $C_i^c = \partial V_{r,i} \cap \partial \mathcal{B}(p_i, r_d) = \bigcup_{k=1}^{n_{c,k}=n_{e,k}} C_{i,k}^c$: The arcs of the range limit, where

$$C_{i,k}^c = \{q_{c,k}(\theta) = p_i + r_d \hat{n}_{i,k}(\theta), \forall \theta \in \Theta_k\},$$

with $\hat{n}_{i,k}^c(\theta) = \left[ \cos \theta \sin \theta \right]^T$, are the $n_{c,k} \in \mathbb{Z}_+$ parts of continuous visibility angles $\Theta_k \subset (-\pi, \pi]$. These points have $dl = r_d d\theta$ and move along with $p_i$, so

$$\frac{\partial q}{\partial p_i} \bigg|_{q \in C_{i}^{c}} = 0_{2 \times 2}$$

and we have

$$\frac{\partial q_{c,k}(\theta)}{\partial p_i} \cdot \hat{n}_{i,k}(\theta) = \hat{n}_{i,k}(\theta), \forall \theta \in \Theta_k, \forall \lambda \in [1, n_{c,k}].$$

If we define as $G_{i}^{A}$ the area integral and as $G_{i}^{c}$, $G_{i}^{c*}$ and $G_{i}^{c*}$ the line integrals of (5) at the respective parts, the equation becomes

$$\frac{\partial H}{\partial p_i}(P) = G_{i}^{A} + (G_{i}^{c} + G_{i}^{c*} + G_{i}^{c*}) + \sum_{j \notin \lambda_{i}(i)} G_{ij}^{g}$$

$$= G_{i}^{A} + G_{i}^{c} + G_{i}^{c*} + \sum_{j \notin \lambda_{i}(i)} \left( G_{ij}^{g} + G_{ij}^{g*} \right).$$

Using Eq. (6) we deduce that $G_{ij}^{g} = -G_{ji}^{g}$, thus

$$\frac{\partial H}{\partial p_i}(P) = G_{i}^{A} + G_{i}^{c} + G_{i}^{c*},$$ (7)

where

$$G_{i}^{A} = \int_{V_{r,i}(P)} \frac{\partial}{\partial p_i} (f(`\| q - p_i `\|`) \phi(q) dA$$

$$G_{i}^{c} = r_d \sum_{k=0}^{n_{c,k}} \int_{q \in \mathcal{F}(q_{c,k})} \phi(q_{c,k}) d\theta$$

$$G_{i}^{c*} = -\sum_{k=0}^{n_{c,k}} \int_{q \in \mathcal{F}(q_{c,k})} \frac{\partial f}{\partial \phi(q_{c,k})} \phi(q_{c,k}) d\theta.$$ (8)

Now it is possible to examine specific cases based on the selection of the performance function $f$.

- **Area Problem:** $f_a(x) = \begin{cases} 1, & \text{if } x < \infty \text{ and } x \in [0, r_d] \\ 0, & \text{otherwise} \end{cases}$

In this case, the objective function is maximised when the area of the cells is maximised. Using Eqs. (7) and (8), we have

$$\frac{\partial H_{a}}{\partial p_i}(P) = G_{a,i}^{c} - G_{a,i}^{c*} = r_d \sum_{k=0}^{n_{c,k}} \int_{q \in \mathcal{F}(q_{c,k})} \phi(q_{c,k}) d\theta$$

$$-\sum_{k=0}^{n_{c,k}} \frac{\partial f}{\partial \phi(q_{c,k})} \frac{\partial q_{c,k}}{\partial p_i} \phi(q_{c,k}) d\theta.$$ (9)

- **Distortion Problem:** $f_d(x) = -x^2 f_a(x)$

This case normally goes along with $r_d = \infty$ and the resulting control law is based on the Lloyds algorithm [31], where the agents are inclined to become evenly spaced by moving towards the centroid of their cell. Here we assume that $f_d(x)$ is not defined for $x = \infty$, thus, the robot cannot assume responsibility for these points, where

$$\frac{\partial H_{d}}{\partial p_i}(P) = G_{d,i}^{A} - \sum_{k=1}^{n_{c,k}} \frac{\partial f}{\partial \phi(q_{c,k})} \frac{\partial q_{c,k}}{\partial p_i} \phi(q_{c,k}) d\theta$$

$$-\sum_{k=1}^{n_{c,k}} \frac{\partial f}{\partial \phi(q_{c,k})} \frac{\partial q_{c,k}}{\partial p_i} \phi(q_{c,k}) d\theta.$$ (10)

- **Mixed Problem:** $f_m(x) = x f_d(x) + f_a(x)$

Combining the above results [11] we have

$$\frac{\partial H_{m}}{\partial p_i}(P) = G_{d,i}^{A} + G_{d,i}^{c} - \gamma_d G_{a,i}^{c}.$$
In the experimental studies, we will examine both the area and the mixed problems. While the results are very close in terms of the covered area, using $f_a$ alone may result in a swarm that is not uniformly swept across the region. This may seem like an unbalanced monitoring; thus it is not preferable. In the mixed problem, this issue is countered by the $f_a$ term, while at the same time, the $f_a$ term ensures that enough importance is given in the increase of the covered area.

5 | HOMING

The objective function for the homing problem Eq. (4) seems already decoupled as long as the assignment function $g$ is defined. But before jumping to the definition, there are some things to be considered. First of all, the robots cannot blindly head to their assigned target, their movements should be restricted to avoid collisions with other robots. Additionally, the target assignment cannot be random, especially in regions with narrow paths. As shown in Figure 3a, a robot which has gloriously reached its target may block the path of another.

To deal with the collision avoidance, we will use the relative visibility cell assignment. Limiting the robot movements in its respective cell, $V_{R,i}$ eliminates any possibility of collisions. Considering the assignment of the targets, to avoid the aforementioned problem of a robot blocking another, it should be carried out in favour of the robots that are farther from the targets. Additionally, targets that are already in the respective cell of a robot should be considered better candidates to avoid decision fluctuations, where the target assignment would change from a time instance to the next.

Firstly, we define the shortest path [28] connecting the points $q_o, q_e \in \Omega$ as the series of points $S(q_o, q_e) = [q_{o,1}, \ldots, q_{o,n}]$, $n_s \geq 1$. The individual points of the path will be referred as $S_k(q_o, q_e)$, with $k \in [0, n_s]$. Considering these points, we have $S_0(q_o, q_e) = q_o$ and $S_n(q_o, q_e) = q_e$, whereas all the other points $S_k(q_o, q_e) \in eO$, $\forall k \in [1, n_s - 1]$, are the reflex vertices. Subsequently, the distance of the shortest path can be written as

$$d_i(q_o, q_e) = \sum_{k=1}^{n_s} ||S_k(q_o, q_e) - S_{k-1}(q_o, q_e)||.$$

Considering the requirements explained above, the distance of the shortest path does not suffice for an assignment criterion; we need a distance that also expresses the proximity of the target to the cell of the robot. Thus, we choose the following distance function

$$d_k(p, z) = \begin{cases} 0, & \text{if } z \in V_{R,i}(p) \\ d_i(p, z) - d_{\min}^{f,i,b}(p, z), & \text{otherwise}, \end{cases}$$

which simply expresses the length of the shortest path minus the part of the path that is in the cell of the robot. The latter can be expressed by $d_{\min}^{f,i,b}(p, z) = \min(d_{\min}^{f,i}(\theta; V_{R,i}), d_{\min}^{f,i}(\theta; V_{R,i}))$.

The proposed assignment function is given in Algorithm 1. As it is also shown in the respective experiment, the assignment may change later, depending on the actual movement of the robots. This is due to the fact that the physical robots move independently and do not have a constant velocity, additionally, they should keep a distance from the obstacles. Thus, the distances from their targets change at different rates. Either case, the fluctuations are minimised by the subtraction of $d_{\min}^{f,i,b}(p, z)$ in $d_k$. In our experimentation, the robots always reached the desired configuration, with no significant changes from the initial assignment.

To explain the methodology, we will use the simple example of Figure 3b. After the cell assignment, the distance function $d_g$ is evaluated for all combinations of robots and targets, as summarised in the Table 1, which is expressed in arbitrary units.

Since robot $p_1$ is the furthest from all the targets, it takes priority in the assignment and is the first to choose its closest target in terms of $d_g$, so $g(p_1) = z_1$. In the next iteration, $p_1$ and $z_1$ are not considered at all and the same procedure is followed with the remaining robots and targets. In our case, the
iteration simply terminates and $g(p_i) = z_j$. Another example is given in Figure 3c, where the assigned targets are marked with the colour of the respective robot.

Having solved the target assignment problem, it is now possible to determine the control inputs for the motion controller. Even though each robot should reevaluate its assigned target at each iteration; for simplicity, we assume that $g(p_i) = z_j$, therefore using Eq. (4) we get

$$ u_i = \frac{\partial \mathcal{H}(P)}{\partial \mathbf{p}_i} = d_i(p_i, z_j) \frac{S_i(p_i, z_j) - p_i}{\|S_i(p_i, z_j) - p_i\|}. \quad (9) $$

Hence, as expected, the optimal way for a robot to minimise the distance to its target is to head towards the next point of the shortest path. An important aspect achieved by the selection of the objective function (4) is that $\|u_i\| = d_i(p_i, z_j)$. This is desirable, because it informs the controller, $M$, that the robot should not stop until this distance is minimised. Here we exploit the capability of the utilised motion controller, where the desired movement direction will be adapted to achieve obstacle avoidance, depending on how much further the robot must go. This results in a smoother and faster trajectory than instructing the robot to explicitly approach the next point of the path by using $(S_i(p_i, z_j) - p_i)$ as the desired direction. This way the robot will properly traverse around the intermediate points of the path and will not stop until the final target has been reached.

For completeness and without any changes to the methodology in our experiments, we will incorporate a desirable orientation $\theta_i$ that accompanies each target point $z_j$. This is a feature of the utilised motion controller, and can be expressed as an extra parameter in Eq. (2). As a result, when the controller assumes the robot reached its destination, or equivalently $\|u_i\| \leq 2r_i$, then it switches to orientation control trying to match the desired orientation $\theta_i$.

**Algorithm 1 Assignment function for the homing problem.**

1: function $g(p; P, Z)$
2: assert $p \in P$
3: $\overline{P} \leftarrow P$
4: $\overline{Z} \leftarrow Z$
5: $\overline{\theta} \leftarrow \emptyset$
6: while $\overline{P} \neq p$ do
7:     $\overline{P} \leftarrow \arg\max_{\overline{P} \in \overline{P}} \left( \min_{\overline{Z} \in \overline{Z}} (d_g(\overline{P}, \overline{Z})) \right)$
8:     $\overline{Z} \leftarrow \arg\min_{\overline{Z} \in \overline{Z}} (d_g(\overline{P}, \overline{Z}))$
9:     $\overline{P} \leftarrow \overline{P} \setminus \overline{P}$
10:    $\overline{Z} \leftarrow \overline{Z} \setminus \overline{Z}$
11: end while
12: return $\overline{Z}$
13: end function

### 6 | EXPERIMENTAL RESULTS

The configuration and the region of our experiments is the same as in [27]. All experiments are performed by deploying $n = 6$ mobile robots; the robots are the AmiGoBot from ActiveMedia, as the one shown in Figure 4. Considering their dimensions, they can be bounded in a radius $R = 0.17$ m around their rotation centre and have a wheel-to-wheel distance $L = 0.279$ m. The maximum supported wheel velocity is

![Figure 4](image_url)  
**Figure 4** AmiGoBot from ActiveMedia

![Figure 5](image_url)  
**Figure 5** Experimental results of area coverage, for $r_d = 0.8$ m, $f_o$. (a) Initial configuration, (b) Trajectories, (c) Final configuration
\( v_{\text{min}} = 1 \text{ m/s}, \) while the velocity limits in our experiments are \( v_{\text{min}} = -0.5 \text{ m/s}, \) \( v_{\text{max}} = 0.5 \text{ m/s} \) for the linear velocity and \( \omega_{\text{max}} = 50 \text{ deg/s} \) for the angular velocity. Correspondingly, the deceleration and acceleration limits are \( a_d = a_a = 0.3 \text{ m/s}^2 \) for the linear and \( a_d = 100 \text{ deg/s}^2, a_a = 50 \text{ deg/s}^2 \) for the angular, respectively.

Considering localisation, it was implemented by utilising both an external camera system and the encoders at the wheels of the robots. The external camera system consists of two cameras on the ceiling, whose image frames are processed by the augmented reality library ArUco [32], to retrieve the position and orientation of the robots in the known region of interest. Additionally, the robots calculate their wheel velocities and their displacement using their equipped encoders, which are then combined with the camera measurements for a more complete and accurate localisation. The localisation error is bounded in a radius \( r_u \approx 0.015 \text{ m}, \) while the sampling time interval in our experiments is \( T_s \approx 160 \text{ ms}. \) Furthermore, we selected \( l_s = R/4; \) as the safe distance that the robots should maintain from the boundary when possible, which is an extra parameter of the utilised motion controller.

### 6.1 Case studies on the area coverage problem

The area coverage problem was examined in three cases. In all cases, a uniform importance density \( \phi = 1 \) was assumed for the region, and the initial position and orientation of the robots was the same. The first case utilises the simplest performance function, \( f_a, \) which is the one expressing the area problem. Additionally, a sensing radius of \( r_d = 0.8 \text{ m} \) was assumed. The results of the experiment are depicted in Figure 5, whereas Figure 6 shows the motion of the robots throughout the experiment. The total area that the swarm can theoretically cover is \( n \pi r_d^2 \approx 12 \text{ m}^2, \) which is larger than the area of the region, where area \( (\Omega) \approx 8.7 \text{ m}^2. \) Naturally, it is impossible to cover the entire region, due to its non-convexity. Therefore, the swarm reached a local maximum of \( \mathcal{H}_a, \) where the assigned area was \( 5.4 \text{ m}^2, \) approximately 62.14% of the total region in almost 40 s, as shown in the coverage progress in Figure 7.

![Figure 6](image6.png)  
**Figure 6** Robot motion of area coverage, for \( r_d = 0.8 \text{ m}, f_a \)

![Figure 7](image7.png)  
**Figure 7** Covered area throughout time

![Figure 8](image8.png)  
**Figure 8** Experimental results of area coverage, for \( r_d = 0.8 \text{ m}, f_a. \) (a) Initial configuration, (b) Trajectories, (c) Final configuration

![Figure 9](image9.png)  
**Figure 9** Robot motion of area coverage, for \( r_d = 0.8 \text{ m}, f_a \)
The second case demonstrates the mixed problem, using the performance function $f_m$, and again a sensing radius of $r_d = 0.8$ m. The results of the experiment are shown in Figure 8 and the robot motion in Figure 9. The final configuration, the attained local maximum of $H_m$ was also reached in almost 40 s, and the assigned area was again 5.4 m², or 62.14% of the region. As it can be seen in Figure 7, the behaviour is very close to that of the previous case, but a little slower. On the other hand, comparing the final configurations, the robots in this case are better swept across the region. It is also interesting to notice that there is a discrepancy on which robot reached each part of the region. This happened because even robots with restricted cells are inclined to move towards the centroid of their cell, while in the previous case, the robots are reluctant to move until they have unobstructed visibility parts in their assigned cell, given that the gradient of the free arcs, $G_{k,i}$ is the decisive factor of the control law.

Finally, the last case shows the mixed problem again, but this time with a smaller sensing radius of $r_d = 0.6$ m. The results are illustrated in Figure 10 and the robot motion in Figure 11. In this case, the total area that the swarm can theoretically cover is $n \pi r_d^2 \approx 6.8$ m², which is approximately 78% of the area of the region. The team achieved a coverage of 40% in 40 s and stabilised at almost 50% coverage of the region in another 30 s. Comparing the final configuration with the ones of the previous cases, it is interesting to notice that it is more similar to the case of the area problem, $f_a$, rather than the one with the same problem. The reason for this is because smaller sensing radius means smaller cells, which in turn, lead to slower movements and behaviour similar to the restricted robots in the area problem.

6.2 Case study on the homing problem

In the homing problem, the knowledge of the region is taken for granted; thus the notion of a sensing radius has no significant meaning and so we assume $r_d = \infty$. In all the previous experiments, the homing algorithm was performed for the swarm to acquire its initial configuration. Therefore, the experiment presented here uses as target positions and orientations for the swarm, the initial configuration of the coverage experiments, and as a starting point the final configuration of the one experiment with $f_m$ and $r_d = 0.8$.

The results are shown in Figure 12 and the motion of the robots throughout the experiment in Figure 13. The swarm reached the desired configuration in 39 s. The trajectories were, naturally, much smoother than the coverage experiments, as the robots had a specific final location to reach and not just a varying direction. This fact is also the reason why certain robots reached different positions than the ones of the initial configuration of the coverage scheme. An interesting turn of events that can be barely observed in the trajectories is that robot 1 initially settled on target $z_2$, but when robot 2 approached, avoiding the obstacles to target $z_1$, the targets swapped leading to the presented final configuration.

7 CONCLUSIONS

This article studied the collaborative control of mobile robots in a non-convex region with obstacles. In contrast to other authors, the dimensions and the full model of the differential drive robot were taken into account by utilising the motion controller of our previous work. Assuming line-of-sight sensors equipped on the robots, a visibility-based approach, slightly different from what is found in the literature, was devised to assign parts of the region to the robots. The problems of area coverage and homing were properly expressed by aggregate objective functions and the optimal movement direction for each robot was successfully
demonstrated. The effectiveness of the proposed schemes was demonstrated by experimental studies.

**REFERENCES**

1. Wang, J., et al.: Time-varying formation of second-order discrete-time multi-agent systems under non-uniform communication delays and switching topology with application to UAV formation flying. IET Control Theory & Appl. 14(14), 1947–1956 (2020)

2. Xu, P., et al.: Distributed event-triggered circular formation control for multiple anonymous mobile robots with order preservation and obstacle avoidance. IEEE Access. 8, 167288–167299 (2020)

3. Miao, Z., et al.: Distributed estimation and control for leader-following formations of nonholonomic mobile robots. IEEE Trans. Autom. Sci. Eng. 15, 1946–1954 (2018)

4. Albina, K., Lee, S.G.: Hybrid stochastic exploration using grey wolf optimiser and coordinated multi-robot exploration algorithms. IEEE Access. 7, 14246–14255 (2019)

5. Fox, D., et al.: Distributed multirobot exploration and mapping. Proc. IEEE. 94, 1325–1339 (2006)

6. Arvanitakis, I., Tzes, A., Giannousakis, K.: Synergistic exploration and navigation of mobile robots under pose uncertainty in unknown environments. Int J Adv Rob Syst. 15(1) (2018). https://journals.sagepub.com/doi/10.1177/1729881417750785

7. Bravo, L., et al.: A distributed exploration Algorithm for unknown environments multiple obstacles by multiple robots. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4460–4466. Vancouver (2017)

8. Lee, W., Vaughan, N., Kim, D.: Task allocation into a foraging task with a series of subtasks in swarm robotic system. IEEE Access. 8, 107549–107561 (2020)

9. Liu, Y., Passino, K.M.: Stable social foraging swarms in a noisy environment. IEEE Trans. Automat. Contr. 49, 30–44 (2004)

10. Zeddada, O., et al.: Multi-agent foraging: state-of-the-art and research challenges. Complex Adapt. Syst. Model. 5(1), 3 (2017)

11. Martínez, J., Martínez, S., Bullo, F.: Spatially-distributed coverage optimization and control with limited-range interactions. ESAIM: COCV. 11(4), 691–719 (2005)

12. Martínez, S., Cortes, J., Bullo, F.: Motion coordination with distributed information. IEEE Contr. Syst. Mag. 27, 75–88 (2007)

13. Mohseni, F., Doustmohammadi, A., Menhaj, M.B.: Distributed reeding horizon coverage control for multiple mobile robots. IEEE Syst. J. 10, 198–207 (2016)

14. Breitenmoser A., et al.: Voronoi coverage of non-convex environments with a group of networked robots. In: 2010 IEEE International Conference on Robotics and Automation, pp. 4982–4989. Anchorage (2010)

15. Kantaros, Y., Thanou, M., Tzes, A.: Distributed coverage control for concave areas by a heterogeneous Robot-Swarm with visibility sensing constraints. Automatica. 53, 195–207 (2015)

16. Lu, L., Choi, Y., Wang, W.: Visibility-based coverage of mobile sensors in non-convex domains. In: 2011 Eighth International Symposium on Voronoi Diagrams in Science and Engineering, pp. 105–111. QingDao (2011)

17. Stergiopoulos, Y., Thanou, M., Tzes, A.: Distributed collaborative coverage-control schemes for non-convex domains. IEEE Trans. Automat. Contr. 60, 2422–2427 (2015)

18. Liu, S., Zhang, R., Shi, Y.: Design of coverage algorithm for mobile sensor networks based on virtual molecular force. Comput. Commun. 150, 260–277 (2020)

19. Thanou, M., Stergiopoulos Y., Tzes, A.: Distributed coverage using geodesic metric for non-convex environments. In: 2013 IEEE International Conference on Robotics and Automation, 6-10 May 2013, Karlsruhe, Germany, Piscataway, NJ, USA: IEEE. pp. 933–938, (2013)

20. Palacios-Gasós, J.M., et al.: Equitable persistent coverage of non-convex environments with graph-based planning. Int. J. Robot. Res. 38(14), 1674–1694 (2019a)

21. Durham, J.W., et al.: Discrete partitioning and coverage control for gossipping robots. IEEE Trans. Robot. 28, 364–378 (2012)

22. Kaposutis, A.C., et al.: DARP: divide areas algorithm for optimal multi-robot coverage path planning. J. Intell. Robot. Syst. 86(3), 663–680 (2017)

23. Kantaros, Y., Thanou, M., Tzes, A.: Visibility-oriented coverage control of mobile robotic networks on non-convex regions. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 1126–1131. Hong Kong (2014)

24. Slattery, J.C.: Advanced transport Phenomena. In: Cambridge Series in Chemical Engineering. Cambridge University Press (1999). https://doi.org/10.1017/CBO9780511800238

25. Floreano, D., Mondada, F.: Evolution of homing navigation in a real mobile robot. IEEE Trans. Syst., Man. Cybern. B. 26, 396–407 (1996)

26. Kim, P., Nakamura, S., Kurabayashi, D.: Robotic homing through a noisy potential field using information entropy. In: 2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 23–26 Nov. 2011, Incheon, Republic of Korea. Piscataway, NJ, USA: IEEE. pp. 594–597, (2011)
27. Giannousakis, K., Tzes, A.: Mobile robot position control in environment with static obstacles. IET Cyber-Syst. Robot. 2(9), 78–87 (2020)
28. Guibas, L.J., Hershberger, J.: Optimal shortest path queries in a simple polygon. J. Comput. Syst. Sci. 39(2), 126–152 (1989)
29. Mur-Arual, R., et al.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Robot. 31, 1147–1163 (2015)
30. Stergiopoulos, Y., Tzes, A.: Autonomous deployment of heterogeneous mobile agents with arbitrarily anisotropic sensing patterns. In: 2012 20th Mediterranean Conference on Control Automation (MED), 3-6 July 2012, pp. 1585–1590. IEEE, Barcelona, (2012)
31. Cortes, J., et al., Coverage control for mobile sensing networks. IEEE Trans. Robot. Automat. 20, 243–255 (2004)
32. Garrido-Jurado, S., et al.: Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern. Recogn. 47(6), 2280–2292 (2014)

**How to cite this article:** Giannousakis K, Tzes A. Distributed control of mobile robots in an environment with static obstacles. *IET Cyber-Syst. Robot*. 2021;3:128–139. https://doi.org/10.1049/csy2.12018

---

**APPENDIX**

In this section we summarise the most important variables that are used throughout the article. The symbols are given in Table 2.

| Symbol | Description |
|--------|-------------|
| Ω      | The region of interest |
| pi     | The position of robot i |
| θi     | The orientation of robot i |
| P      | The set of distinct robot positions |
| rd     | The radius of the omnidirectional range sensor on each robot |
| dV(q, pi) | The distance of visible points from robot i |
| dRV(q, pi) | Relative visibility distance from robot i |
| dLRV(q, pi) | The limited relative visibility distance from robot i |
| Vi     | The unlimited visibility space of robot i |
| Vi′    | The range-limited visibility space of robot i |
| VR,i(P) | The relative visibility cell of robot i |
| VR,i,r(P) | The range-limited relative visibility cell of robot i |
| df(θ, Wi) | The distance of the obstacles around robot i |
| d(qi, qj) | The length of the shortest path between points qi, qj ∈ Ω |
| zj     | The position of target j |
| Z      | The set of distinct target positions |
| g(pi; P, Z) | The bijective function expressing the robot-to-target assignment |

The symbols in bold italics represent vectors and are consistent with the rest of the article.