



Persistent coverage control for a team of agents with collision avoidance



Carlos Franco ^{a,*}, Dušan M. Stipanović ^b, Gonzalo López-Nicolás ^a, Carlos Sagüés ^a, Sergio Llorente ^c

^a Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, 50018 Zaragoza, Spain

^b Department of Industrial and Enterprise Systems Engineering, and the Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801, USA

^c Research and Development Department, Induction Technology, Product Division Cookers, BSH Home Appliances Group, 50016 Zaragoza, Spain

ARTICLE INFO

Article history:

Received 15 April 2014

Received in revised form

20 October 2014

Accepted 1 December 2014

Recommended by A. Aghdam

Available online 24 December 2014

Keywords:

Persistent coverage

Range sensing

Nonholonomic motion planning

Collision avoidance

ABSTRACT

In this paper, the idea of persistent coverage to be accomplished by multiple agents while avoiding collisions is considered and developed. The persistent coverage problem is formulated by assuming that the coverage degrades over time. In this framework, our contribution is a new distributed control law which is capable of carrying out the persistent coverage without computing agents' paths explicitly. The proposed setup considers agents with nonholonomic motion constraints and it is based on the combination of local and global strategies to achieve efficient coverage while avoiding bottlenecks such as local minima. The local strategy is based on the gradient of the coverage error in the neighborhood of an agent whereas the global strategy leads the agents to uncovered areas of the domain. Furthermore, we present a new bounded potential repulsion law and a proof of safe navigation is provided for the case of unicycle vehicles. We also propose a modification of the tangent-bug algorithm to deal with multiple non-point agents which allows the team to navigate in environments with non-convex obstacles in a reactive manner. Simulation results illustrate the performance of the proposed control law.

© 2014 European Control Association. Published by Elsevier Ltd. All rights reserved.

1. Introduction

The problem of area coverage by a team of agents is of interest in a wide variety of applications such as cleaning [23], lawn mowing [1] and monitoring [32]. In general, the use of multiple agents to solve this problem enhances the coverage performance by, for example, decreasing the coverage time. However, multiple agents introduce additional issues like collision avoidance and coordination of agents.

In the case of static coverage the agents are assumed to be able to cover an area by being placed at particular positions [9]. However, if the agents are mobile, it is possible to deploy the resources and adapt their positions to a variety of environments. Deployment has been solved through different approaches: with Voronoi partitions [7,15], by using potential fields [27,4,24], or with gradient methods [33]. In dynamic coverage problems agents are assumed to have a limited sensing range and can not cover the domain statically, that is, by any deployment. To accomplish the task, some approaches

compute a path explicitly [6,30,17], whereas others solve the problem without computing a path [18,13,14]. If the environment is invariable, the problem is solved by covering all the points once [6,18,13,14]. However, some tasks require to re-cover all the points over time because the environment evolves and the task is to monitor the area persistently [30,17,19]. Our work is focused on the latter scenario which we refer to as the persistent coverage.

A relevant issue that arises when dealing with multiple agents and environments with obstacles is the problem of collision avoidance. Obstacle avoidance for navigation purposes has been addressed by many different strategies: potentials [21], vector field histogram [2], dynamic window approach [11], elastic bands [28], nearness diagram navigation [25], and model predictive control [29], to name a few.

In the field of coverage, the work presented in [3] introduces the tangent-bug algorithm to avoid convex obstacles and develop deployment in multi-agent environment. However, they consider point agents and do not consider inter-agent avoidance. On the other hand inter-agent avoidance has been commonly treated with a repulsive force that grows relative to the proximity among the agents. In [5,26] authors introduce forces that modify the angular velocity whereas Liu et al. (2006) [22] and Dimarogonas et al. (2006) [8] introduce a repulsive force that modifies both module and direction of the motion toward the main objective. In the field of dynamic coverage with

* Corresponding author.

E-mail addresses: cfranco@unizar.es (C. Franco),
dusan@uiuc.edu (D.M. Stipanović), gonlopez@unizar.es (G. López-Nicolás),
csagues@unizar.es (C. Sagüés), sergio.llorente@bshg.com (S. Llorente).

sensors networks, Hussein and Stipanović [18] introduce a scheme based on the collision avoidance ideas from [31]. In those papers inter-agent avoidance is solved but obstacle avoidance is not addressed.

In this work, we focus on the problem of persistent coverage control by a team of nonholonomic agents in an environment with a coverage decay. We propose a coverage control law based on ideas introduced in [13,14] where a local strategy and a global strategy are combined. We begin by proposing a new model for the evolution of the coverage, based on a differential equation that evolves between zero and a maximum coverage level. The behavior of the model can be tuned with two gains, the sensing gain and the decay gain, to adjust the values for different scenarios.

To develop persistent coverage we propose reactive strategies that do not compute a path explicitly. We use a strategy based on the gradient of the coverage error to find the best direction to move instantaneously. As gradient strategies may get trapped in local minima, we combine the local strategy with a global strategy that leads agents to uncovered areas. Both strategies are continuously weighted in such a way that agents obey their local control laws if the error in their neighborhoods is high, and they move to new areas obeying the global control law when there is no benefit in covering the nearby areas. To reach uncovered areas by avoiding obstacles and other agents in a reactive fashion we use the tangent-bug algorithm [20] with a modification which allows the algorithm to work in environments with multiple non-point agents. Once the coverage action is obtained from the local and global strategy, it is combined with a new bounded repulsion law. The coverage control law and the repulsion law are weighted depending on the danger of collision to obtain the desired motion.

Finally, with the target motion we design a control law to govern the nonholonomic agents. The angular velocity input is proportional to the angular error, and the linear velocity input takes into account the maximal speed of the agent, the local coverage error, the angular error, the distance to global goals and the danger of collision. If the local coverage error is high, the speed is decreased to provide a better coverage of the neighborhood. If the local coverage error is low, the speed is increased for the agent to leave the covered area. The speed is also decreased as the angular error gets larger to avoid high linear speeds while turning, and as agents approach global targets and obstacles. In this paper, the coverage information and the global strategy are centralized but the motion is agent-based. This is done to reduce the communication costs and increase the flexibility to changing environments while keeping a good level of efficiency. In fact, each agent can compute the coverage map, and the global goal of each agent is achieved with only position information.

Specifically this paper provides the following items as the main contributions:

- An algorithm that develops persistent coverage without computing explicitly agents' paths.
- An adaptation of the tangent-bug algorithm to multi-agent environments, to allow multiple non-point agents to reach their global goals in unconnected environments or with non-convex obstacles.
- A new bounded potential repulsion law for agents that allows safe navigation for unicycles. Furthermore, proofs of collision avoidance in multi-agent environments with obstacles are provided.

The paper is organized as follows: Section 2 introduces the problem formulation and the model of the evolution of the coverage. Section 3 presents the coverage strategy and the repulsion law. Section 4 describes how coverage and collision avoidance objectives are combined, presents the nonholonomic motion control law, and provides collision avoidance proofs. Section 5 provides simulation results of the proposed algorithm in different environments. Finally, Section 6 presents the conclusions of the paper.

2. Problem formulation

In this section we introduce the problem formulation and a new evolution coverage model for a team of agents performing dynamic coverage tasks. We abuse notation by including the dependencies of the variables only when they are defined. One of the main objectives is to reach a desired coverage level $\Lambda^*(x) \in \mathbb{R}^+$ for all the points $x \in D_x$ over a bounded domain $D_x \subset \mathbb{R}^2$. We assume that mobile agents behave as differential drives, that is, each agent A_i of the team $A = \{A_1, \dots, A_N\}$ of N agents is governed by the following model:

$$\begin{aligned}\dot{p}_{i_1} &= v_i \cos(\theta_i), \\ \dot{p}_{i_2} &= v_i \sin(\theta_i), \\ \dot{\theta}_i &= \omega_i,\end{aligned}\tag{1}$$

where $\dot{p}_i(t) = [\dot{p}_{i_1}(t), \dot{p}_{i_2}(t)]^T$ is the motion vector of the i -th agent, $p_i(t) = [p_{i_1}(t), p_{i_2}(t)]^T$ is the position of the i -th agent in a domain $D_p \subset \mathbb{R}^2$, $\theta_i(t) \in (-\pi, \pi]$ is the orientation angle, $v_i(t) \in \mathbb{R}$ is the input velocity, and $\omega_i(t) \in \mathbb{R}$ is the input angular velocity. Let us define $r(x, p_i) \in \mathbb{R}$ as the Euclidean distance between a point x and the position of agent p_i , and $\alpha_i(r) \in \mathbb{R}$ as the coverage action that the i -th agent develops over points at distance r inside each agent's actuator domain (or footprint which will be used interchangeably) $\Omega_i(p_i, R) \subset \mathbb{R}^2$. In this work, we restrict to circular actuators with limited range $R \in \mathbb{R}^+$ so that their coverage function α_i is

$$\begin{cases} \alpha_i \geq 0 & \text{if } r < R (x \in \Omega_i), \\ \alpha_i = 0 & \text{if } r \geq R (x \notin \Omega_i). \end{cases}\tag{2}$$

In Section 5 we give an explicit example of a coverage actuator function. Then, the actuator domain of each agent Ω_i is the area where the agent is developing coverage action, and it varies over time with the position of the agent. Note that D_p , which is the domain of all positions reachable by the agents, can be different from D_x , which is the domain of the points to be covered. The points of D_x just need to be to R or closer to any point of D_p in order to be covered. The coverage action of the team of agents is defined as $\alpha = \sum_{i \in \{1, \dots, N\}} \alpha_i$. Furthermore we define $\Lambda(t, x) \in [0, \bar{\Lambda}(x))$ as the coverage level developed by a team of agents over a point x at time t , where $\bar{\Lambda}(x) \in \mathbb{R}^+$ is the maximum reachable coverage.

Here we propose a new model for the evolution of the coverage level. The coverage information is updated continuously at each point x with the following differential equation:

$$\frac{\partial \Lambda}{\partial t} = K_s(\bar{\Lambda} - \Lambda)\alpha - K_d\Lambda\tag{3}$$

where $K_s \in \mathbb{R}^+$ is the coverage gain, $K_d \in \mathbb{R}^+$ is a gain that represents the decay of the coverage, and the term $(\bar{\Lambda} - \Lambda)$ can be interpreted as the intensity of covering point x . The dynamic behavior of the model can be changed by modifying K_d and K_s to adapt to different applications. Notice that in the case that $(\bar{\Lambda} - \Lambda)$ is constant, (3) is equivalent to a first-order linear system. We assign $\Lambda(0, x) = 0$, $\forall x \in D_x$, which means that at the beginning all points are assumed as uncovered. With this initial value, the model solutions are bounded and evolve between no coverage, with $\Lambda = 0$ and the maximum reachable coverage $\Lambda = \bar{\Lambda}$. The main difference with respect to other models with decay as, for example, the one proposed in [16], is that it is possible to select independently the maximum reachable coverage, the decay rate and the coverage gain, whereas in other proposals these quantities are coupled. This provides the possibility of modeling additional scenarios as, for example, snow removal, lawn mowing or confidence in surveillance, since the decay is very slow compared to the coverage ability, and the maximum coverage is not necessarily related to these values. Fig. 1 shows an example of the values that Λ takes at one point with two different dynamics but the same coverage objective Λ^* , maximum coverage level $\bar{\Lambda}$ and coverage

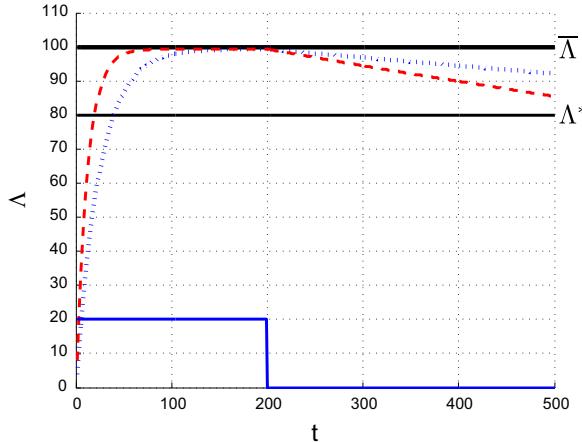


Fig. 1. Evolution of Λ at one point in two different scenarios with different parameters. Dotted line represents a system with $K_s = 1/500$ and $K_d = -1/4000$ whereas dashed line represents a system with $K_s = 1/250$ and $K_d = -1/2000$. Solid thin line represents the coverage action α which is 20 at the beginning and 0 after 200 units of time.

action α . The maximum coverage level is reached quickly when the coverage action is applied, and it decays slowly when the action is null. As the sensing gain increases the system reaches the maximum coverage faster and as the decay gain increases the coverage degrades faster. Notice that as the maximum coverage level is higher than the coverage objective, the objective can be fulfilled during some time after the applied action is null.

Here, we introduce the lack of coverage $Y(t, x) \in \mathbb{R}^+$ over a point x at time t as

$$Y = \max\left(0, \frac{\Lambda^* - \Lambda}{\Lambda^*}\right). \quad (4)$$

Note that Y is a measurement of the coverage error at each point scaled with the desired coverage level Λ^* , and that we only take positive lack of coverage. In this way, we consider coverage problems where the excess of coverage is not harmful. Moreover, we introduce $\Phi(x) \in (0, 1]$, $\forall x \in D_x$, as the priority to cover each point x . Φ is a map that weights the significance of the points in the domain to give more priority to particular zones of special interest.

At this point let us define the error function $e_{D_x}(t) \in [0, 1]$ over the whole domain as

$$e_{D_x} = \frac{\int_{D_x} Y \Phi dx}{\int_{D_x} \Phi dx}, \quad (5)$$

and the error function of the actuator domain of each agent $e_{\Omega_i}(t) \in [0, 1]$ as

$$e_{\Omega_i} = \frac{\int_{\Omega_i} Y \Phi dx}{\pi R^2}. \quad (6)$$

3. Dynamic coverage control laws

The crucial objective of our proposed distributed control law is to keep decreasing the error e_{D_x} . Let us now divide the domain into the points which have positive lack of coverage $D_x^+ (Y) = \{x \in D_x | Y > 0\}$, and the rest $D_x^0 (Y) = \{x \in D_x | Y = 0\}$. Notice that both domains are complementary and their union is the whole coverage domain D_x . The domains depend on time since the sign of Y changes as the domain is covered by the agents or become uncovered due to the decay. We want to minimize the variation of the error of each agent with respect to its own position, dropping $\int_{D_x} \Phi dx$ because it is just a scaling factor.

We compute the derivative of the error over time as [10]:

$$\begin{aligned} \frac{de_{D_x}}{dt} &= \frac{d}{dt} \left(\int_{D_x^+} Y \Phi dx + \int_{D_x^0} Y \Phi dx \right) \\ &= \iint_{D_x^+} \frac{\partial Y}{\partial t} \frac{\partial \Lambda}{\partial t} \Phi dx_1 dx_2 + \int_{D_x^+} Y \Phi \left(\frac{\partial \Lambda_1}{\partial t} \frac{\partial x_2}{\partial x_1} - \frac{\partial \Lambda_2}{\partial t} \frac{\partial x_1}{\partial x_2} \right), \end{aligned} \quad (7)$$

where \overline{D}_x^+ is the boundary of D_x^+ and $x = (x_1, x_2)$. In the integration we take out the points that do not have lack of coverage and, as the limits of integration of D_x^+ depend on the derivative variable, we apply the Leibniz integral rule to the case of two dimensions in the Cartesian coordinates (x_1, x_2) . We obtain two terms, the variation inside the domain, and the variation of the boundary. Notice that Y on the boundary is 0 by definition of the domain and thus the second term is 0. By substituting (3) into (7) we get

$$\begin{aligned} \frac{de_{D_x}}{dt} &= \int_{D_x^+} (K_d \Lambda - K_s (\overline{\Lambda} - \Lambda) \alpha) \frac{\Phi}{\Lambda^*} dx \\ &= \int_{D_x^+} K_d \Lambda \frac{\Phi}{\Lambda^*} dx - \sum_{i=1}^N \int_{D_x^+ \cap \Omega_i} K_s (\overline{\Lambda} - \Lambda) \alpha_i \frac{\Phi}{\Lambda^*} dx. \end{aligned} \quad (8)$$

The first term is the degradation of the information, which increases the error and takes place over the whole domain. The second term represents the agent's coverage action over the domain and it reduces the error. The amount of coverage developed by the agents depends on the error over Ω_i , which is the actuator's footprint that depends on the position of the i -th agent. Thus, it is possible to decrease the variation of the error with respect to the position of the agents by decreasing this term. As we aim for a distributed control law, we would like to optimize this term with respect to the position of each agent.

3.1. Local control law

Instantaneously we can locally compute the action to decrease the error function with the gradient $u_i^{grad}(t) \in \mathbb{R}^2$ for each agent i :

$$u_i^{grad} = - \int_{D_x^+ \cap \Omega_i} K_s (\overline{\Lambda} - \Lambda) \cdot \frac{\partial \alpha_i}{\partial r} \frac{(p_i - x)}{\|p_i - x\|} \frac{\Phi}{\Lambda^*} dx. \quad (9)$$

Eq. (9) can be seen as the gradient of the integral on the right hand side of Eq. (8) with respect to p_i , that is, the position of the agent i . Note that since error function is only a function of time, we cannot mathematically write this expression as the gradient of $\partial e / \partial t$ with respect to p_i . From this equation we can extract the direction of the motion $\hat{u}_i^{loc}(t) \in \mathbb{R}^2$ to get the maximum benefit covering the neighborhood of the agent as

$$\begin{cases} \hat{u}_i^{loc} = \frac{u_i^{grad}}{\|u_i^{grad}\|} & \text{if } \|u_i^{grad}\| \neq 0, \\ \hat{u}_i^{loc} = (0, 0)^T & \text{if } \|u_i^{grad}\| = 0. \end{cases} \quad (10)$$

However, gradient techniques are known to get trapped in local minima and, when the local error e_{Ω_i} is low, the benefit of covering the neighborhood of an agent is small. Because the information decays in other subareas of the domain, it may happen that the error over the domain e_{D_x} increases. Therefore, it is advisable to combine this gradient with a global law $u_i^{glo}(t) \in \mathbb{R}^2$ that depends on the coverage of the whole domain to bring the agents to places with higher error and improve the overall performance of the coverage.

3.2. Global control law

To reach other areas in the domain with higher error in an environment with obstacles, we propose the use of a modified version of the tangent-bug algorithm. Tangent-bug is a reactive algorithm

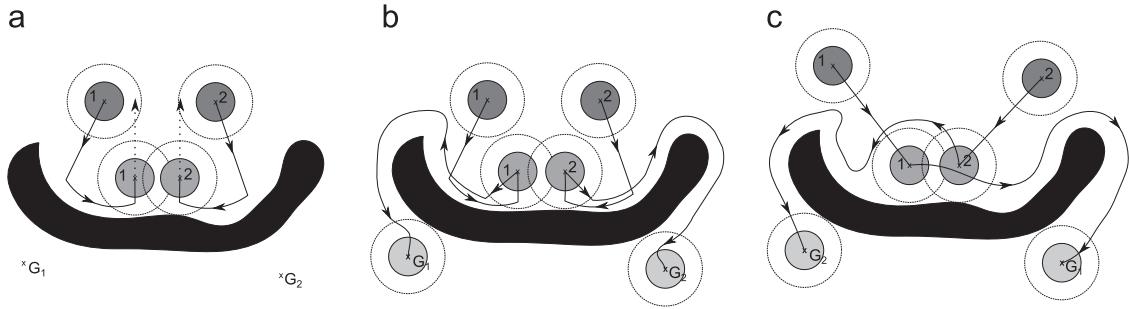


Fig. 2. Blocking example of tangent-bug algorithm applied to multi-agent environments and behavior with our modification proposal. Agents are represented by grey circles, black area represents the obstacle and the goal of the i -th agent is denoted by G_i . Figure (a) represents the behavior of the standard tangent-bug algorithm proposed by [20]. Figure (b) represents the behavior of the modified tangent-bug proposed if trajectories between the agents to their goals do not intersect. Figure (c) represents the modified algorithm if trajectories intersect.

which ensures that a single agent with range sensor will reach its global goal if there exists a path to reach it as presented by [20]. Roughly speaking, it is based on searching for free directions within the sensor range, and following the boundary of obstacles if there is no free direction toward the goal. Tangent-bug algorithm has been developed in environments with one agent and static obstacles. In multi-agent environment there are no guarantees of goal reaching. For example, in the case that two non-point agents are following the boundary of the same obstacle in opposite directions and they meet, they will leave the obstacle in perpendicular direction, going both in parallel trajectories trying to follow a contour which is moving at the same velocity as the agents. Therefore, both agents can move continuously and without stopping.

To solve this issue, we propose a modification. It consists in leaving the contour if the only obstacles detected are other agents. Then if the segments that connect the agents' position and their respective objectives intersect, all the agents, but the one nearest to its objective, stop. Thus the stopped agents are treated as static obstacles and keep the guarantee of reaching the goal. When those segments do not intersect, the agents start the search for a new free direction to move. This modification guarantees that agents do not leave the domain and assures that the agents never stop performing coverage of the domain.

An example is depicted in Fig. 2. The agents start at dark grey positions and they start moving toward G_i along straight lines until they detect the obstacle. Then, they decide to go toward the center of the U-shaped obstacle since that direction of the outline is closer to the direction of the goal. Afterwards, both follow the contour until they meet. With the standard tangent-bug algorithm, once they meet, they move in parallel ways to infinity, following the dotted lines as plotted in Fig. 2(a). After the agents meet, if they stop following the boundary once they detect just mobile obstacles, and they search for new free directions to move, they would go straight toward the goal until they find and follow the boundary again, this time in opposite directions while reaching the goal Fig. 2(b). Let us focus now in the situation in Fig. 2(c). In that case, when both agents meet with the standard tangent-bug algorithm, the agents will go in parallel to infinity as depicted in Fig. 2(a). With our proposal, since the segment $\overline{G_1 p_1}$ intersects with $\overline{G_2 p_2}$ the agent which is further to the objective stops. Then, agent 2 moves around it, and when both segments do not intersect, agent 1 can continue following the contour until it reaches G_1 .

Thus, the modified tangent-bug algorithm provides a direction to move $\hat{u}_i^{glo}(t)$ based on a goal position $p_i^*(t) \in D_p$, from where uncovered points can be reached by the agents, and a set of obstacles. After obtaining the direction to move, the desired global

action u_i^{glo} is computed with

$$u_i^{glo} = k_i^{glo} \hat{u}_i^{glo} \quad (11)$$

where $k_i^{glo}(\|p_i - p_i^*\|) \in (0, 1)$ is a function that depends on the distance to the goal. A good choice we propose is the following one:

$$k_i^{glo} = \tanh\left(\frac{2\|p_i - p_i^*\|}{R}\right). \quad (12)$$

This function is close to 1 until the distance from an agent to a goal is almost the coverage radius R , and then, it decreases. It allows the agents to reach the goals quickly, and then slow down when the goals are being accomplished.

The selection of the global goals p_i^* is made through a strategy which finds areas with high error. It is based on blob detection of the uncovered information [13,12]. We use this image processing algorithm to find uncovered areas, and then we compute their centroids $\psi_j \in \mathbb{R}^2$ and their coverage error, $e_{\psi_j} \in \mathbb{R}$ where $j \in 1, \dots, M$, by integrating the coverage error of each blob's area,. The objective p_i^* for each agent i is chosen by weighting distances to the centroids and coverage error of each blob. Each agent i obtains for each centroid j a score $0 < S_{(i,j)}(t) \leq 2$ to compose a matrix S of dimension $N \times M$, in the following way:

$$S_{(i,j)} = \left(1 - \frac{\|p_i - \psi_j\|}{\max(\|p_i - \psi_j\|)}\right) + \frac{e_{\psi_j}}{\max(e_{\psi_j})}. \quad (13)$$

Then, the global goals are assigned using the **Algorithm 1** which matches a global goal with each agent taking into account the matrix of scores S . The algorithm is repeated N times, each time finding the maximum score and pairing the corresponding objective j with an agent i . After each match is made, the algorithm reduces the score of all the centroids for agent i by $2N$ units to prevent the assignment of a new centroid to the same agent. Then, the algorithm also reduces the score of the centroid by 2 units (the maximum possible score) for all agents. If there are more agents than centroids, the aim is to produce an even distribution of the agents amongst the centroids. On the other hand, if there are more centroids than agents, the algorithm will assign each agent to a different centroid.

Algorithm 1. Assignment of objectives.

```

require  $S, \Psi$ 
ensure  $p_i^*$ 
1:   repeat
2:      $[i, j] = \{i, j : S_{(i,j)} = \max(S)\};$ 
3:      $p_i^* = \psi_j;$ 
4:      $S_{(i,:)} = S_{(i,:)} - 2N;$ 

```

5: $S_{(i,j)} = S_{(i,j)} - 2$;
 6: **until** All agents have an objective

3.3. Coverage control law

To combine both global and local control laws let us introduce a local weight $W_i^{loc}(t) \in [0, 1]$ and a global weight $W_i^{glo}(t) \in [0, 1]$ as

$$W_i^{loc} = e_{\Omega_i}^\beta \quad (14)$$

$$W_i^{glo} = 1 - e_{\Omega_i}^\beta \quad (15)$$

where the exponent $\beta \in \mathbb{R}^+$ is a parameter to be tuned depending on the desired behavior of the algorithm and the parameters of the problem. Further details will be provided in Section 5. The target direction of the coverage $u_i^{cov} \in \mathbb{R}^2$ is obtained with

$$u_i^{cov} = W_i^{loc} \hat{u}_i^{loc} + W_i^{glo} u_i^{glo}. \quad (16)$$

The weights force the agents to obey the local control law when the local error e_{Ω_i} is high, moving to the direction of the gradient of the error, and force the agents to obey the global control law when the local error is low, heading toward new uncovered areas.

3.4. Collision avoidance

In this section we propose a new repulsive control law for agents with range sensors which is bounded, and has a limited range of actuation. We start by introducing $d_{il}(t) \in \mathbb{R}^+$ as the Euclidean distance between the position of the agent p_i and the detected position of an obstacle $p_l(t) \in \mathbb{R}^2$ with $l \in \{1, \dots, L\}$. Notice that the detected obstacles include other agents and static obstacles and from the point of view of the repulsion control law there is no need to distinguish between them. The avoidance gain for each obstacle $k_{il}^{col}(d_{il}) \in \mathbb{R}^+$ is computed as

$$k_{il}^{col} = \left(\frac{1}{2} + \frac{1}{2} \cos \left(\pi \frac{d_{il} - r_i^{saf}}{r_i^{avo} - r_i^{saf}} \right) \right)^\gamma, \quad d_{il} < r_i^{avo} \quad (17)$$

where $r_i^{saf} \in \mathbb{R}^+$ is the safety distance and $r_i^{avo} \in \mathbb{R}^+$ is the avoidance distance, i.e., the distance at which the repulsive action starts. Variables involved are shown in Fig. 3. In addition, $\gamma \in \mathbb{R}^+$ is a parameter that allows changing the shape of the repulsion (as illustrated in Fig. 4). Higher γ allows approaching obstacles closer or passing through narrower corridors, thus improving the coverage performance but also reducing the security margin for collisions.

The repulsive force $u_{il}^{col}(t) \in \mathbb{R}^2$ of an agent i to an obstacle l is computed as

$$u_{il}^{col} = \frac{\hat{p}_{il}}{d_{il} - r_i^{saf}}, \quad (18)$$

where $\hat{p}_{il}(t) \in \mathbb{R}^2$ is the normalized vector between the position of the agent p_i , and the position of the detected obstacle p_l . To obtain

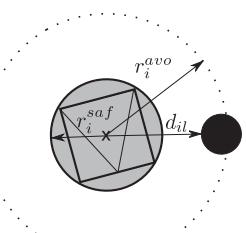


Fig. 3. Variables involved in the computation of k_{il}^{col} at (17). The grey circle represents the security region of agent i , the dashed line the avoidance region and the black circle represents the obstacle l .

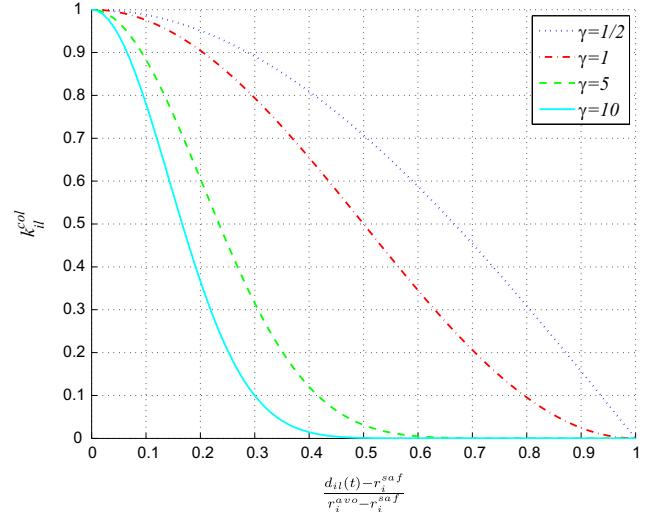


Fig. 4. Avoidance gain k_{il}^{col} for different values of γ .

the resultant direction $\hat{u}_i^{col}(t) \in \mathbb{R}^2$ of the avoidance action of agent i we normalize the sum of the repulsive forces as

$$\hat{u}_i^{col} = \frac{\sum_{l=1}^L u_{il}^{col}}{\| \sum_{l=1}^L u_{il}^{col} \|}. \quad (19)$$

The direction of the repulsive force is then a weighted sum of the repulsive forces for each obstacle. The obstacles that are nearer weigh more than the obstacles that are further away. We also compute the resultant collision avoidance gain $k_i^{col}(k_{il}^{col}) \in \mathbb{R}$ as the maximum of the repulsive collision avoidance gains:

$$k_i^{col} = \max_l \{k_{il}^{col}\}. \quad (20)$$

Then, we propose the following total avoidance action $u_i^{col}(t) \in \mathbb{R}^2$ for each agent:

$$u_i^{col} = k_i^{col} \hat{u}_i^{col}. \quad (21)$$

4. Safe coverage

In this section we show how to combine the control laws related to coverage and obstacle avoidance in order to accomplish both objectives. We also propose a control law for the nonholonomic model considered. First, we introduce coverage gain as the complementary of the collision gain $k_i^{cov}(t) = 1 - k_i^{col}$ and then we compute the desired motion action $u_i(t) \in \mathbb{R}^2$ as

$$u_i = k_i^{cov} u_i^{cov} + u_i^{col} = k_i^{cov} u_i^{cov} + k_i^{col} \hat{u}_i^{col}. \quad (22)$$

Then, we can extract the desired orientation $\theta_{d_i}(t) \in (-\pi, \pi]$ for the unicycle, from the components of u_i as

$$\theta_{d_i} = a \tan 2(u_{i2}, u_{i1}). \quad (23)$$

Applying a proportional controller to the minimum angular distance, we get the angular velocity input:

$$\omega_i = \begin{cases} k_{\omega_i}(\theta_i - \theta_{d_i} + 2\pi) & \text{if } (\theta_i - \theta_{d_i}) < -\pi \\ k_{\omega_i}(\theta_{d_i} - \theta_i) & \text{if } -\pi \leq (\theta_i - \theta_{d_i}) \leq \pi \\ k_{\omega_i}(\theta_i - \theta_{d_i} - 2\pi) & \text{if } (\theta_i - \theta_{d_i}) > \pi \end{cases} \quad (24)$$

where $k_{\omega_i} \in \mathbb{R}^+$ is the angular velocity gain.

If we define the angular error $e_{\theta_i}(t) \in [0, 2]$ as

$$e_{\theta_i} = \begin{cases} \frac{(\theta_i - \theta_{d_i} + 2 \cdot \pi)}{\pi/2} & \text{if } (\theta_i - \theta_{d_i}) < -\pi \\ \frac{|\theta_{d_i} - \theta_i|}{\pi/2} & \text{if } -\pi \leq (\theta_i - \theta_{d_i}) \leq \pi \\ -\frac{(\theta_i - \theta_{d_i} - 2 \cdot \pi)}{\pi/2} & \text{if } (\theta_i - \theta_{d_i}) > \pi \end{cases} \quad (25)$$

then to control the linear velocity we propose

$$v_i = k_{v_i}(1 - e_{\Omega_i})(1 - e_{\theta_i})\|u_i\| \quad (26)$$

where k_{v_i} is the velocity gain of agent i and its maximum value must be the maximum speed of the agent. Note that by definition $v_i \in [-k_{v_i}, k_{v_i}]$. The term $(1 - e_{\Omega_i})$ forces the agent to stay in uncovered zones when $e_{\Omega_i} \rightarrow 1$, or makes it go faster to leave covered zones when $e_{\Omega_i} \rightarrow 0$. The term $(1 - e_{\theta_i})$ allows the agent to choose maximum backward or forward velocity, when the desired angle and the angle of the agent are aligned, or slow the agent down as the minimum angular distance ($\min(|\theta_i - \theta_{d_i}|, |\theta_{d_i} - \theta_i|)$) approaches $\pi/2$, when movement cannot be developed in the desired direction because of the nonholonomic constraints and $0 \leq \|u_i\| \leq 1$.

Lemma 1. *The projection of the motion vector of agent i (governed by (1) with the control laws (24) and (26)) denoted by \dot{p}_i on the desired direction is always non-negative. Thus,*

$$\cos(\dot{p}_i, u_i) \geq 0, \quad (27)$$

Proof. Let us compute a change of coordinates in which θ_{d_i} is aligned with the first coordinate of the new system, and therefore $\theta_{d_i} = 0$, i.e. we use the desired orientation as a reference frame. Let us keep the names of the variables in the new coordinate system invariable. Analyzing the sign of the terms of \dot{p}_{i_1} after substituting (26) in (1) as

$$\dot{p}_{i_1} = k_{v_i}(1 - e_{\Omega_i})(1 - e_{\theta_i})\|u_i\| \cos(\theta_i), \quad (28)$$

we see that $k_{v_i} > 0$, $(1 - e_{\Omega_i}) \geq 0$, and $\|u_i\| \geq 0$. Thus, let us pay attention to terms $(1 - e_{\theta_i})$ and $\cos(\theta_i)$. As it is well known,

$$\cos(\theta_i) \geq 0 \quad \text{if } 0 \leq |\theta_i| \leq \pi/2, \quad (29)$$

$$\cos(\theta_i) < 0 \quad \text{if } \pi/2 < |\theta_i| \leq \pi. \quad (30)$$

Furthermore, it is easy to evaluate (25) by substituting $\theta_{d_i} = 0$ and θ_i in the range $[-\pi, \pi]$. The results are

$$(1 - e_{\theta_i}) \geq 0 \quad \text{if } 0 \leq |\theta_i| \leq \pi/2, \quad (31)$$

$$(1 - e_{\theta_i}) < 0 \quad \text{if } \pi/2 < |\theta_i| \leq \pi. \quad (32)$$

Therefore both terms always have the same sign and then

$$\cos(\theta_i) \cdot (1 - e_{\theta_i}) \geq 0 \quad \forall \theta_i. \quad (33)$$

Consequently, $\dot{p}_{i_1} \geq 0 \quad \forall \theta_i$ and the projection of the motion vector of agent i on the desired direction is always non-negative, which is equivalent to $\cos(\dot{p}_i, u_i) > 0$. \square

Note that when $|\theta_i - \theta_{d_i}| < \pi/2$ agent i moves forward, i.e., $v_i > 0$, and when $|\theta_i - \theta_{d_i}| > \pi/2$ agent i moves backwards since $v_i < 0$. In this way agents obey the desired directions of movement.

Lemma 2. *If the distance between the i -th agent and any obstacle is smaller than $d_{0.5}$, where*

$$d_{0.5} = r_i^{saf} + \frac{1}{\pi} \arccos(2\sqrt{0.5} - 1)(r_i^{avo} - r_i^{saf}), \quad (34)$$

then, the projection of the desired motion action of the i -th agent u_i

on the avoidance action \hat{u}_i^{col} is always positive

$$\cos(u_i, \hat{u}_i^{col}) > 0. \quad (35)$$

Proof. We compute (35) as

$$\cos(u_i, \hat{u}_i^{col}) = \frac{u_i \cdot \hat{u}_i^{col}}{\|u_i\| \cdot \|\hat{u}_i^{col}\|}, \quad (36)$$

Substituting (22) in (36) we have

$$\cos(u_i, \hat{u}_i^{col}) = \frac{(k_i^{cov} \cdot u_{i_1}^{cov} + k_i^{col} \cdot \hat{u}_{i_1}^{col}) \cdot \hat{u}_i^{col}}{\|u_i\| \cdot \|\hat{u}_i^{col}\|}. \quad (37)$$

Here, let us consider a change of coordinates such that the first coordinate of the coordinate system is aligned with \hat{u}_i^{col} . Let us also keep the same notation in the new coordinate system. As a consequence we have $\hat{u}_i^{col} = (1, 0)$ and in the new coordinate system (37) is equivalent to

$$\cos(u_i, \hat{u}_i^{col}) = \frac{k_i^{cov} \cdot u_{i_1}^{cov} + k_i^{col}}{\|u_i\| \cdot \|\hat{u}_i^{col}\|}, \quad (38)$$

where $u_{i_1}^{cov}$ is the first component of \hat{u}_i^{cov} . Notice that $\|u_i\| \in [0, 1]$ and $\|\hat{u}_i^{col}\| = 1$ and thus they do not affect to the sign of the cosine. Furthermore, substituting $d_{il} \in [r_i^{saf}, d_{0.5}]$ into (17) we obtain $k_{il}^{col} \in (0.5, 1]$ and therefore $k_i^{cov} = (1 - k_i^{col}) \in [0, 0.5]$. Also, notice that $\|u_i^{cov}\| \leq 1$ and also that $|u_{i_1}^{cov}| \leq 1$. Therefore

$$k_i^{col} > |k_i^{cov} \cdot u_{i_1}^{cov}| \quad \text{if } d_{il} \in [r_i^{saf}, d_{0.5}] \quad (39)$$

and then

$$\cos(u_i, \hat{u}_i^{col}) = \frac{k_i^{cov} \cdot u_{i_1}^{cov} + k_i^{col}}{\|u_i\| \cdot \|\hat{u}_i^{col}\|} > 0, \quad (40)$$

As $\cos(u_i, \hat{u}_i^{col}) > 0$ then $u_i \hat{u}_i^{col} \in (-\pi/2, \pi/2)$ and the projection of the desired motion action of the i -th agent u_i over the avoidance action \hat{u}_i^{col} is always positive when $d_{il} < d_{0.5}$ \square

Note that if there is only one obstacle inside the agent detection range and the distance is lower than $d_{0.5}$, it would be moving away from it. Thus, on the one hand the agent obeys the desired direction and on the other hand the desired direction obeys the avoidance control law when the danger of collision arises.

Lemma 3. *An agent i with dynamics given in (1) and governed by the control laws (24) and (26) approaching an obstacle or other agents will keep a safety distance with respect to all of them to be larger than r_i^{saf} :*

$$d_{il} > r_i^{saf} \quad (41)$$

Proof. Let us assume that we have an agent i surrounded by L obstacles in a set $O = \{O_1, \dots, O_l, \dots, O_L\}$, no matter whether they are static obstacles or other agents. Let us assume that the nearest obstacle is O_1 . Then, the direction of our repulsive force can be computed with (19) as

$$\hat{u}_i^{col} = \frac{\hat{p}_{i1} \cdot Q_{i1} + \dots + \hat{p}_{il} \cdot Q_{il} + \dots + \hat{p}_{iL} \cdot Q_{iL}}{\|\hat{p}_{i1} \cdot Q_{i1} + \dots + \hat{p}_{il} \cdot Q_{il} + \dots + \hat{p}_{iL} \cdot Q_{iL}\|}, \quad (42)$$

where $Q_{il} = 1/(d_{il} - r_i^{saf})$ from (18). We can compute the angle formed by the total avoidance action with the avoidance action of the nearest obstacle $u_{i1}^{col} \hat{u}_i^{col}$, and if it is in $(-\pi/2, \pi/2)$ that means that the direction of the total repulsive force is against the nearest obstacle. Let us proceed by computing the cosine of both vectors as

$$\cos(u_{i1}^{col}, \hat{u}_i^{col}) = \frac{|u_{i1}^{col}| \cdot |\hat{u}_i^{col}|}{\|u_{i1}^{col}\| \cdot \|\hat{u}_i^{col}\|}, \quad (43)$$

where $\|u_{i1}^{col}\| = Q_{i1}$ and $\|\hat{u}_i^{col}\| = 1$. Let us compute a change of coordinates such that the first coordinate is aligned with u_{i1}^{col} and

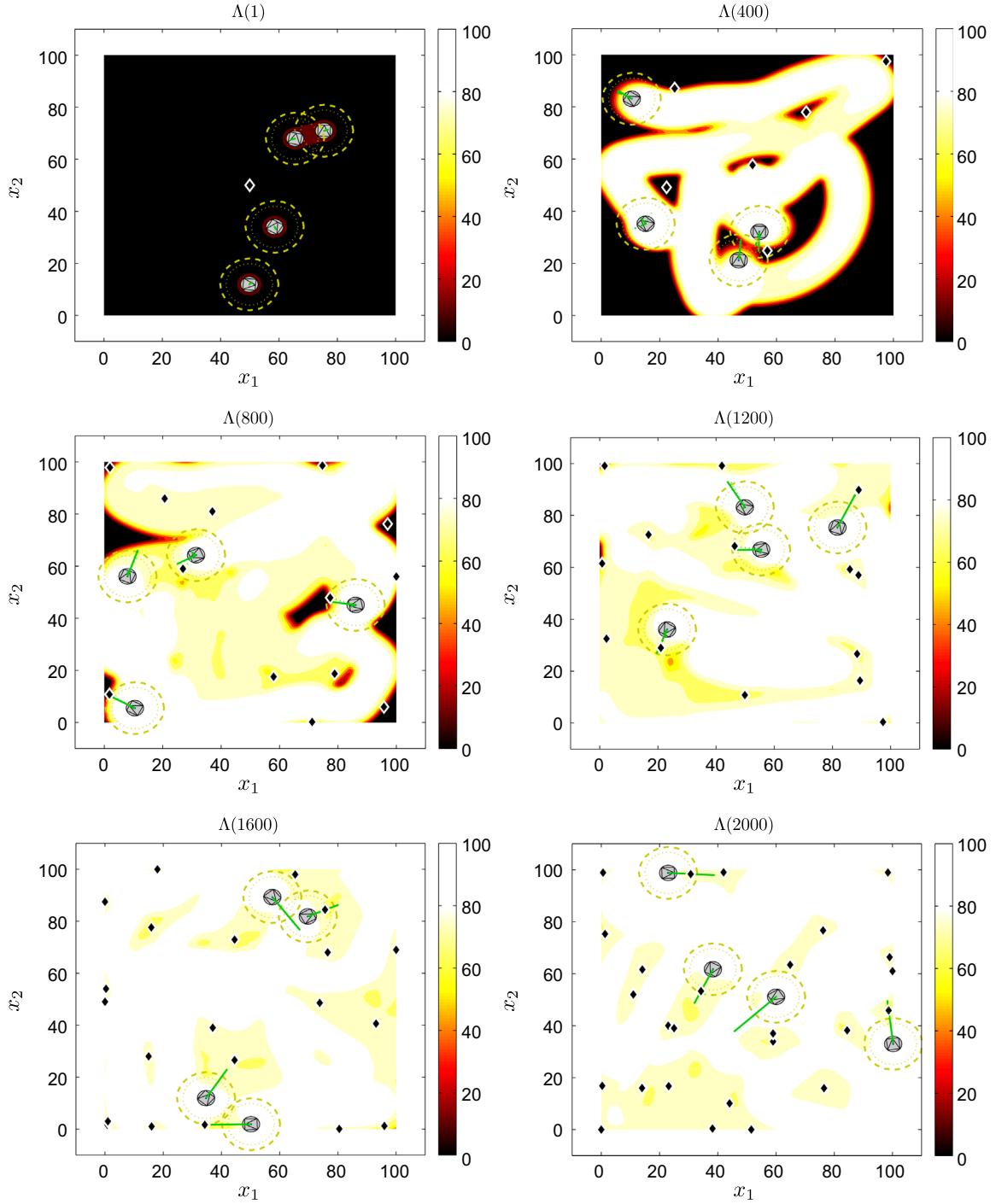


Fig. 5. Evolution of the coverage map. Small circles represent the positions of the agents, their coverage domain is represented by a dashed line, and the avoidance regions are represented by the thin dotted line. Solid green straight lines represent the total action, dotted green lines represent the global coverage actions, and solid blue lines represent the repulsion actions. The small rhombi represent the global goals. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

then $u_{i1}^{col} = (Q_{i1}, 0, \dots, 0)$. Let us also keep the notation the same in the new coordinate system. After the change of coordinates we can compute the cosine as

$$\cos(u_{i1}^{col}, \hat{u}_i^{col}) = \frac{Q_{i1} \cdot u_{i1}^{col}}{\|u_{i1}^{col}\| \cdot \|\hat{u}_i^{col}\|}, \quad (44)$$

where $u_{i1}^{col} \in \mathbb{R}$ is the first direction vector of \hat{u}_i^{col} , which is computed as

$$u_{i1}^{col} = \frac{p_{i1_1} \cdot Q_{i1} + \dots + p_{il_1} \cdot Q_{il} + \dots + p_{il_1} \cdot Q_{il}}{\|\hat{p}_{i1} \cdot Q_{i1} + \dots + \hat{p}_{il} \cdot Q_{il} + \dots + \hat{p}_{1L} \cdot Q_{il}\|}. \quad (45)$$

However, notice that $\|u_{i1}^{col}\| = Q_{i1}$ and that $\|\hat{u}_i^{col}\| = 1$. Therefore $\cos(u_{i1}^{col}, \hat{u}_i^{col}) = u_{i1}^{col}$, and then

$$\cos(u_{i1}^{col}, \hat{u}_i^{col}) = \frac{p_{i1_1} \cdot Q_{i1} + \dots + p_{il_1} \cdot Q_{il} + \dots + p_{il_1} \cdot Q_{il}}{\|\hat{p}_{i1} \cdot Q_{i1} + \dots + \hat{p}_{il} \cdot Q_{il} + \dots + \hat{p}_{1L} \cdot Q_{il}\|}. \quad (46)$$

Notice also that $p_{i1_1} = 1$ since $\|\hat{p}_{i1}\| = 1$ and it is aligned with the first direction vector of the coordinate system. Additionally, as the agent is approaching the nearest obstacle, Q_{i1} grows and

$$\lim_{d_{i1} \rightarrow r_i^{saf}} Q_{i1} = \lim_{d_{i1} \rightarrow r_i^{saf}} \frac{1}{d_{i1} - r_i^{saf}} = \infty \quad (47)$$

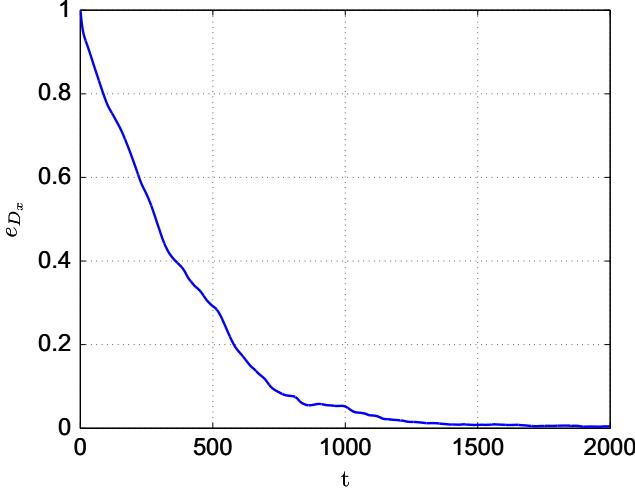


Fig. 6. Normalized coverage error evolution in simulation of Fig. 5.

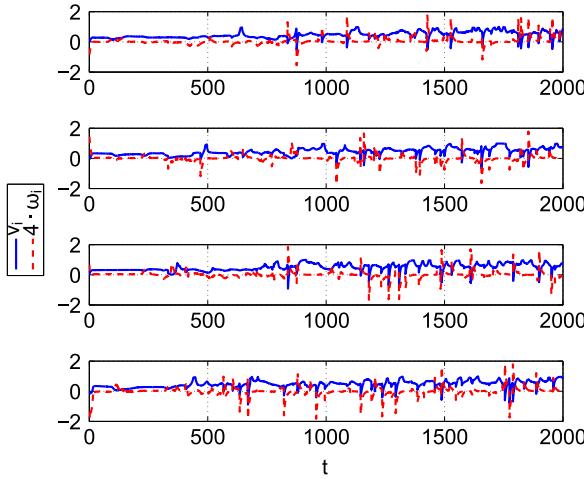


Fig. 7. Actions of each agent during the simulation of Fig. 5. Solid lines represent v_i , while dotted lines represent ω_i scaled $\times 4$ for better visibility.

As we consider non-point agents or obstacles, and limited avoidance range, it is impossible to fit an infinite number of obstacles in the avoidance range and then

$$Q_{i1} > |p_{i2_1} \cdot Q_{i2} + \dots + p_{il_1} \cdot Q_{il} + \dots + p_{il_1} \cdot Q_{il}| \quad \text{if } d_{il} = r_i^{saf}. \quad (48)$$

Consequently

$$\cos(u_{i1}^{col}, \hat{u}_i^{col}) = 1 \quad \text{if } d_{il} = r_i^{saf} \quad (49)$$

Thus, when the distance to the nearest obstacle is r_i^{saf} the avoidance action points against the nearest obstacle. By Lemma 2 the desired motion obeys the avoidance control law if $d_{il} < d_{0.5}$. In the particular case of $d_{il} = r_i^{saf}$, $k_i^{cov} = 0$ and then

$$\cos(u_i, u_{i1}^{col}) = 1 \quad \text{if } d_{il} = r_i^{saf}. \quad (50)$$

As a consequence, the desired motion is opposite to the direction of the nearest obstacle. Furthermore, by Lemma 1 agents respect the desired motion direction. Therefore:

$$\cos(\hat{p}_i, u_{i1}^{col}) > 0 \quad \text{if } d_{i1} = r_i^{saf} \quad (51)$$

Therefore, if an agent always moves toward the desired direction, and it is governed by the avoidance control law, and the avoidance

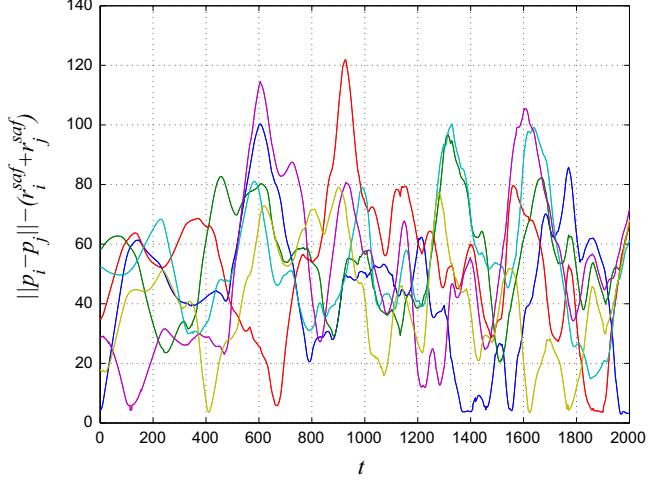


Fig. 8. Distance between the agents during the simulation of Fig. 5 computed as $\|p_i - p_j\| - (r_i^{saf} + r_j^{saf})$. Notice that the safety distances are included to show the distance between the contours of the agents. It can be seen that there are no collisions since the distances are always greater than 0.

Table 1
Testing environments and the corresponding values for the motion parameters.

| Environment | K_s | α_M |
|-------------|-------|------------|
| 1 | 1/500 | 40 |
| 2 | 1/250 | 40 |
| 3 | 1/250 | 80 |

law points against the nearest obstacle then, the distance to the nearest obstacle is always going to be greater than r_i^{saf} . \square

Lemma 4. A team of agents performing coverage with control law provided in Eqs. (9)–(26), in an environment without decay ($K_d=0$) and with $R > d_{0.5}$, drives the coverage error of the domain to zero, that is, $e_{D_x} \rightarrow 0$ as $t \rightarrow \infty$.

Proof. If $K_d=0$ in (8) we obtain

$$\frac{de_{D_x}}{dt} = - \sum_{i=1}^N \int_{D_x^+ \cap \Omega_i} K_s (\bar{\Lambda} - \Lambda) \alpha_i \frac{\Phi}{\Lambda^*} dx, \quad (52)$$

which is always negative, or zero if the agents' domain $\bigcup_{i=1}^N \Omega_i$ is covered:

$$\frac{de_{D_x}}{dt} < 0 \quad \text{if } \sum_{i=1}^N e_{\Omega_i} > 0 \quad (53)$$

$$\frac{de_{D_x}}{dt} = 0 \quad \text{if } \sum_{i=1}^N e_{\Omega_i} = 0 \quad (54)$$

The latter (54) happens if the whole domain D_x is covered, and then $e_{D_x} = 0$. And also when $D_x^+ \cap \Omega_i = \emptyset$ but $D_x^+ \neq \emptyset$, i.e., if the robots' actuator domain is covered but some point of the whole coverage domain D_x is not covered yet. If $D_x^+ \cap \Omega_i = \emptyset$, then $e_{\Omega_i} = 0$ and consequently $u_i^{cov} = u_i^{glo}$. Global coverage action is governed by the modified tangent-bug algorithm which guarantees each agent to reach its goal position if there exists a path. If the goal position p_i^* provided is not covered yet, then to reach it implies to continue covering the domain after some period of time. However, coverage action is combined with the collision avoidance law. As introduced in Lemma 2 the desired motion u_i of agent i never goes against avoidance action if any obstacle is closer than $d_{0.5}$. Then, if the remaining points to be covered are close to the obstacles, the agents will approach the obstacles at most $d_{0.5}$. With $R > d_{0.5}$ we

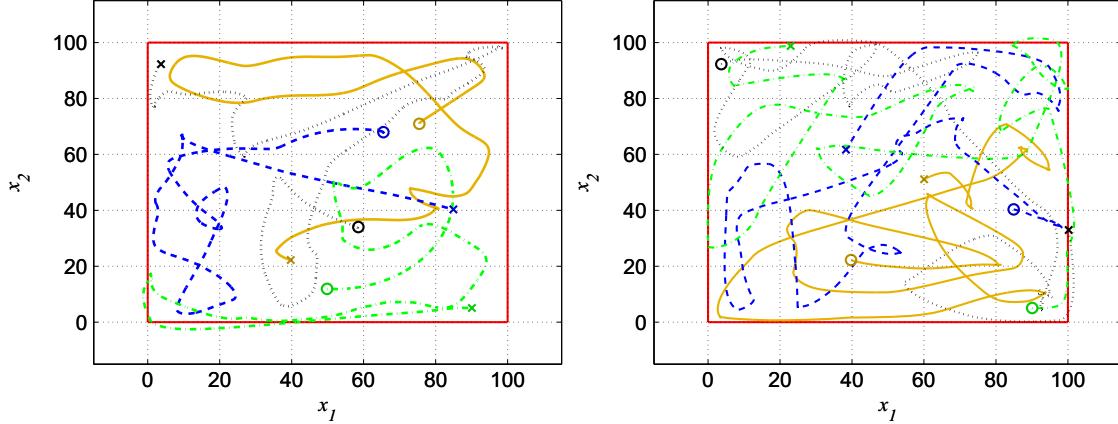


Fig. 9. Trajectories of agents from $t=1$ to $t=1000$ (left) and from $t=1000$ to $t=2000$ (right) during simulation of Fig. 5. The paths of agents start at circles and end at crosses. We divide the figures into two intervals of time for a better perception of the trajectories.

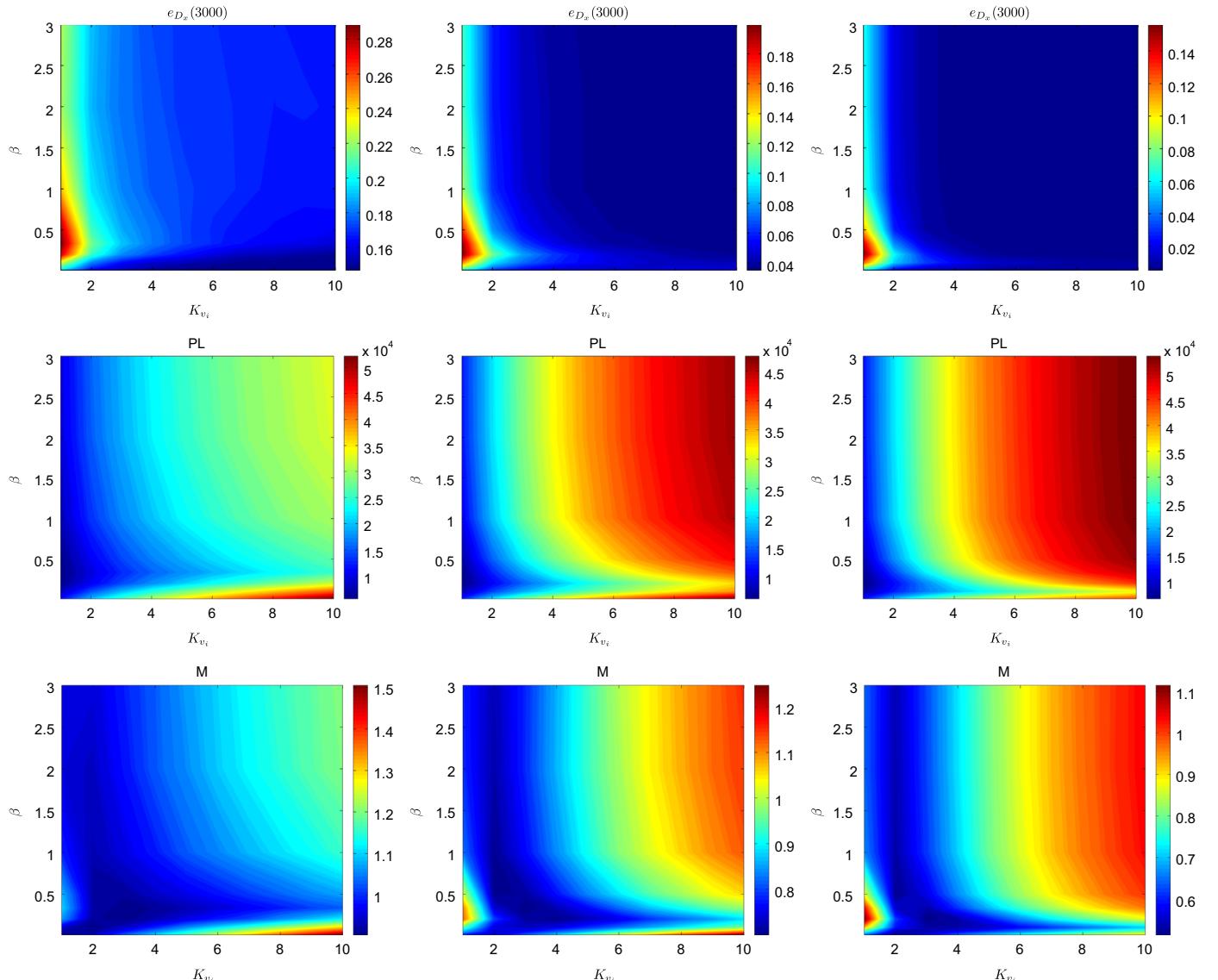


Fig. 10. Simulation results of Section 5.1. From first to third column, environments 1–3 are tested. In the first row we show the average steady error of the domain e_{D_x} of the 100 simulations carried out with each combination of parameters. The second row presents the average path length traveled (PL) to develop the coverage by the team of agents. The third row presents an efficiency measurement (M) of the motion parameters.

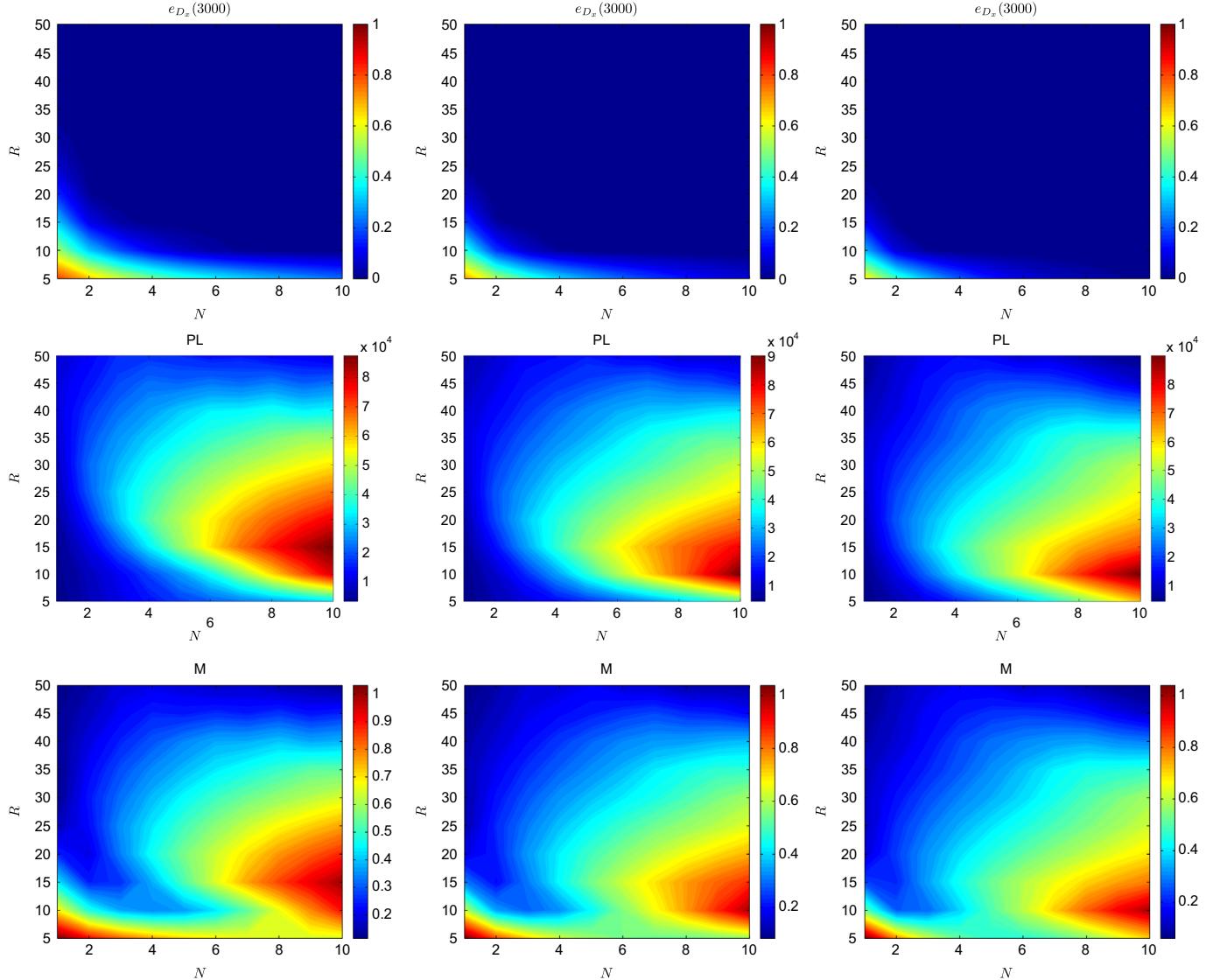


Fig. 11. Simulation results of Section 5.2. From first to third column, environments 1 to 3 are tested. In the first row we show the average steady error of the domain e_{D_x} of the 100 simulations carried out with each combination of parameters. The second row presents the average path length traveled (PL) to develop the coverage by the team of agents. The third row presents an efficiency measurement (M) of the motion parameters.

guarantee that the points can be covered from reachable positions and then a position where $D_x^+ \cap \Omega_i = \emptyset$ can be always reached when $e_{D_x} > 0$. If it is possible to reach positions where $de/dt < 0$ always when $e_{D_x} > 0$ then, it is possible to force the error to decrease until $e_{D_x} = 0$. \square

In the case $K_d \neq 0$ and if agents cannot cover all the points of the domain simultaneously, full coverage is not guaranteed. Since points in the domain will increase its error continuously and covered points do not attract agents, then agents will take some time to reach points which are covered and become uncovered. However, it is guaranteed that the agents will not stop covering the domain while $e_{D_x} > 0$. Note also that unless an agent is pushed against an obstacle by other agents, it will approach obstacles at most a distance equal to $d_{0.5}$. Then, if $R < d_{0.5}$ the points of the domain which are near an obstacle will never be covered. A similar case might happen if two global goals are side by side and two agents try to reach them in opposite direction. Agents will try to reach them by the global control law, but the repulsion law will prevent from approaching the other agent closer than $d_{0.5}$ and then, will prevent from reaching the goal. However, it is not necessary for the agents to be at their global goals to cover them,

and if $R > d_{0.5}/2$ the goal positions would be covered, and new goals would be assigned to the agents. Finally, note that $d_{0.5}$ can be tuned by increasing γ to make it as small as $r_{i,saf}^*$ and that it is very unlikely to have global goals for the agents being too close. Our strategy selects the centroids of uncovered areas as goals for the global motion control law. The areas with higher errors are selected firstly as targets to the agents. Those areas use to be the bigger ones and then, their centroids are far.

Theorem 5. *A team of agents whose dynamics are given by (1) and governed by the control laws (24) and (26) in an environment with static obstacles, develops persistent coverage without collisions.*

Proof. The proof follows directly from Lemmas 3 and 4. \square

5. Simulation results

In this section we show simulation results of the proposed control laws. Firstly we introduce the coverage function of agent i :

$$\alpha_i(r) = \begin{cases} \frac{\alpha_M}{R} (r^2 - R^2)^2, & r \leq R \\ 0, & r > R \end{cases} \quad (55)$$

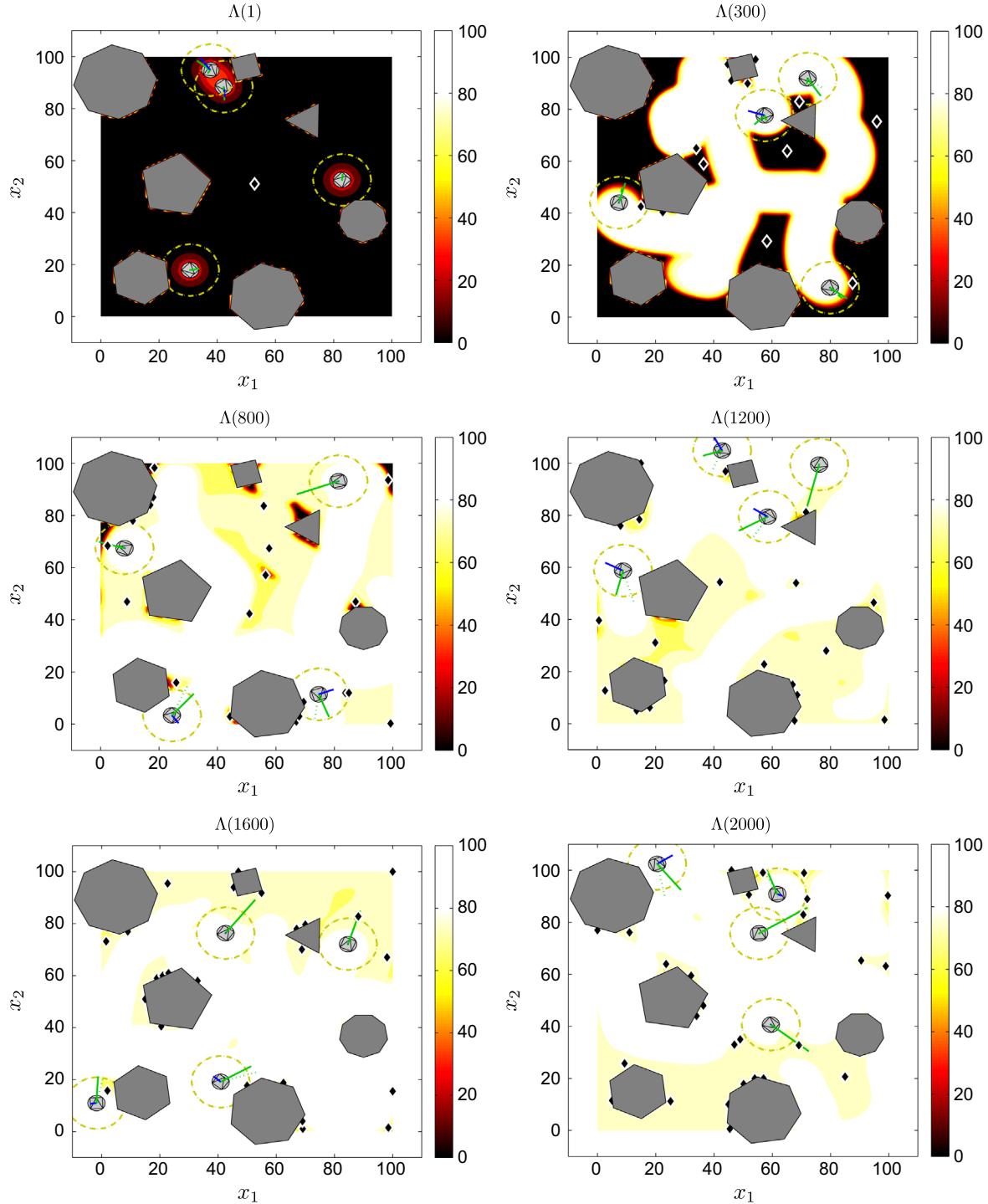


Fig. 12. Evolution of the coverage map with convex obstacles. Small circles represent the positions of the agents, their coverage domain is represented by a dashed line, and the avoidance regions are represented by the thin dotted line. Solid green straight lines represent the total action, dotted green lines represent the global coverage actions, and solid blue lines represent the repulsion actions. The small rhombi represent the global goals. The solid dark gray blocks represent the obstacles. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

where α_M is the maximum level of coverage and R is the range of the agent. Note that other coverage functions could be chosen.

Using this coverage action we present a coverage simulation for 2000 units of time over a square domain D_x of 100×100 units by a team of four agents. The parameters of the coverage function are as follows: $K_s = 1/250$, $K_d = -1/2000$, $\bar{\Lambda} = 100$. The coverage objective is $\Lambda^* = 80$ with a constant interest $\Phi = 1$. The parameters of the collision avoidance law are as follows: $r_i^{avo} = 8$, $r_i^{saf} = 3$, $\gamma = 1$. The parameters of the motion controller are as follows: $k_{v_i} = 1$, $k_{\omega_i} = 0.15$, $\beta = 1/3$. The parameters of the coverage action are as

follows: $\alpha_M = 40$, $R = 10$. Fig. 5 shows the evolution of the coverage map. The agents first cover the domain, and in approximately 1000 units of time it is almost covered. After that, the team maintains the coverage level.

Fig. 6 shows the evolution of the total coverage error e_{D_x} which never reaches 0 due to the decay of the information and thus, agents are continuously moving to cover new points. Fig. 7 represents the linear and angular velocities of each agent. Fig. 8 shows the distances between the contour of the agents, which are always greater than 0, i.e. agents do not collide. Fig. 9 shows the trajectories

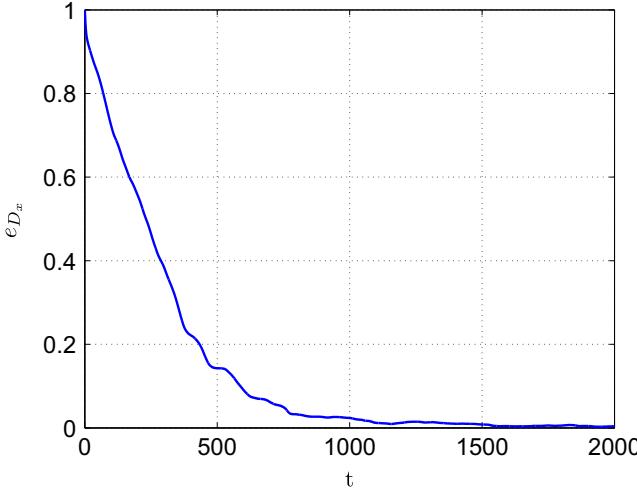


Fig. 13. Normalized coverage error evolution of the simulation presented in Fig. 12.

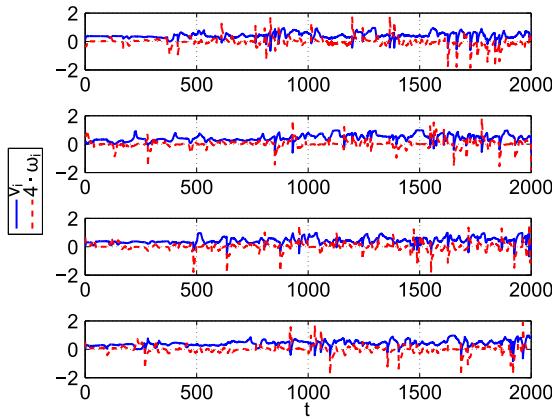


Fig. 14. Actions of each agent during the simulation presented in Fig. 12. Solid lines represent v_i , and dotted lines represent ω_i scaled $\times 4$ for better visibility.

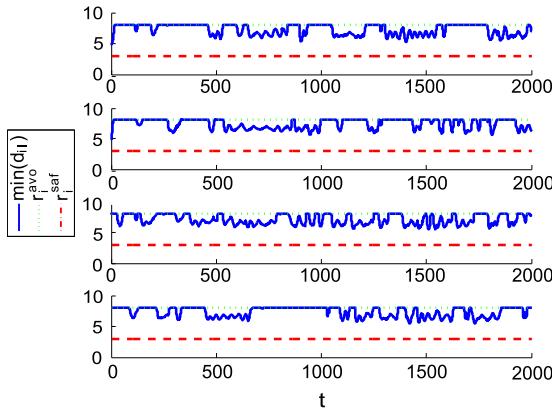


Fig. 15. Distance between the agents and the nearest obstacle detected by the range sensor at every time in the simulation presented in Fig. 12. It can be seen that there are no collisions since the distances are always greater than the safety distance of each agent r_i^{saf} .

of the agents. For better clarity we depicted the resulting paths from 0 to 1000 units of time, where the main coverage action is performed, and from 1000 to 2000, where the agents have already visited once all the points of the domain and the action consists in coverage maintenance.

To analyze the influence of different parameters of the problem we provide simulations in three environments. In the experiments

we vary K_s and α_M according to Table 1. From environment 1 to 3, the actuation ability of agents is increased. The rest of the common parameters are duration of 3000 units of time, developed in a square domain D_x of 100×100 units. The parameters of the coverage function are $K_d = -1/2000$, $\bar{A} = 100$. The coverage objective is $A^* = 80$ with a constant interest $\Phi = 1$.

5.1. Motion parameters: k_{v_i} vs β

In what follows, we provide 100 simulations with teams of three agents starting at random positions combining each value of $k_{v_i} = \{1, 2, \dots, 10\}$ and $\beta = \{1/100, 1/50, 1/20, 1/10, 1/5, 1/3, 1/2, 1, 2, 3\}$. The parameters of the collision avoidance law are as follows: $r_i^{avo} = 8$, $r_i^{saf} = 3$, $\gamma = 1$. The angular velocity gain is $k_{\omega_i} = 0.15$ and the coverage action range is $R = 10$.

In Fig. 10 we show the results of the simulations. From the first to the third column the experiments are carried out in the environments 1, 2 and 3 respectively. In the first row we show the average steady coverage error (e_{D_x}) of the 100 simulations carried out with each combination of parameters. The second row presents the average path length (PL) traveled to develop the coverage by the team of agents during the total number of simulations (N_{sim}):

$$PL = \frac{\sum_{i=1}^N \int_{t=0}^{3000} \| \dot{p}_i \| dt}{N_{sim}} \quad (56)$$

The third row presents a combination of both values (M) defined as follows:

$$M_{(k_{v_i}, \beta)} = \frac{PL_{(k_{v_i}, \beta)}}{\max(PL)} + \frac{e_{D_x}(k_{v_i}, \beta)}{\max(e_{D_x})} \quad (57)$$

Lower values mean higher efficiency in the coverage process because a smaller error with shorter path lengths is obtained. From the first to the third column the experiments are carried out in the environments 1, 2 and 3, respectively. As the linear speed gain k_{v_i} grows, the agents achieve a smaller coverage error, but also produce longer path lengths. Furthermore, the benefit obtained increasing the speed decreases as speed grows, and saturates at a speed which depends on the actuation ability of the agents. By combining both criteria, the coverage error and the path length in the third row of Fig. 10 where M is represented, we can conclude that the most efficient strategy for these environments is the one with $\beta = 1/3$, and $k_{v_i} = 3$, because it is a combination of parameters that obtain the lowest value of M in the three environments.

5.2. Available resources: N vs R

In this subsection we analyze how the coverage is developed with different available resources given the same three environments of previous section (Table 1). We perform simulations with a variable number of agents $N = \{1, 2, \dots, 10\}$ and variable actuator range $R = \{5, 10, \dots, 50\}$ combining each value of N and R , and carrying out 100 simulations with each combination. The parameters of the collision avoidance law are: $r_i^{avo} = R$, $r_i^{saf} = 3$, $\gamma = 1$. The linear velocity gain is $k_{v_i} = 3$, the angular velocity gain is $k_{\omega_i} = 0.15$ and finally $\beta = 1/3$.

In Fig. 11 we show the results of the simulations. As in the previous section, from the first to the third column the experiments are carried out in the environments 1, 2 and 3, respectively. In the first row we show the average steady coverage error (e_{D_x}) of the 100 simulations carried out with each combination of parameters. The second row presents the average path length traveled (PL), while coverage is performed by the team of agents. The third row presents a combination of both values (M) showing efficiency and is computed with (57). The average steady error decreases with the number of agents and the range of actuation in all the environments. The path lengths grow with the number of agents, but do not evolve monotonously with the actuator range. For low values, $R = \{5, 10, 15\}$, the

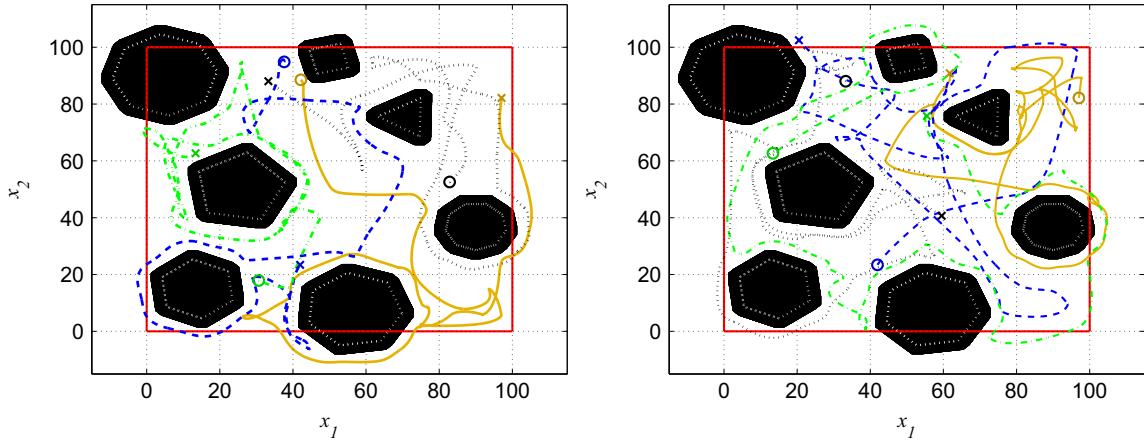


Fig. 16. Trajectories of agents from $t=1$ to $t=1000$ (left) and from $t=1000$ to $t=2000$ (right) of the simulation presented in Fig. 12. The paths of the agents start at circles and end at crosses. Obstacles are represented by black blocks. They have been expanded by r_i^{avo} from their actual boundaries (white dotted lines) to show the avoidance area. We divide the figures into two intervals of time for a better perception of the trajectories.

path lengths grow with the actuator range. As the agents cover a bigger area with larger coverage ranges R , the speed grows to reach uncovered areas and then they produce larger path lengths. With $R > 15$ the teams of agents reach a low steady error rapidly and the agents start to run through covered areas following the global control law. As they have a larger coverage action, it takes shorter time to reach global goals. In the representation of M in the third row, it is possible to obtain the optimal number of agents for a given actuator range and vice versa in these environments.

5.3. Collision avoidance parameters and coverage with obstacles

The parameters of the collision avoidance law have to be chosen according to the size and dynamics of the agents. r_i^{saf} is the distance the i -th agent do not violate and thus, it must correspond with the size of the security margin. r_i^{avo} is the distance where the repulsion starts and then it must be equal or smaller than the coverage range R to allow covering points near obstacles. γ allows changing the slope of the intensity of the repulsion depending on the distance. $\gamma > 1$ causes a soft repulsion if obstacles are around r_i^{avo} of the agent, and a strong repulsion when agents are close to an obstacle, whereas $\gamma < 1$ causes already a strong repulsion around r_i^{avo} .

We present now a simulation with 7 static polygonal obstacles and with the same parameters as the first simulation provided in Section 5. The simulation is 2000 units of time long, developed in a square domain D_x of 100×100 units. The team of agents is composed of 4 agents. The parameters of the coverage function are as follows: $K_s = 1/250$, $K_d = -1/2000$, $\bar{\Lambda} = 100$. The coverage objective is $\Lambda^* = 80$ with a constant interest $\Phi = 1$. The parameters of the collision avoidance law are: $r_i^{avo} = 8$, $r_i^{saf} = 3$, $\gamma = 1$. The parameters of the controller are: $k_{v_i} = 1$, $k_{\omega_i} = 0.15$, $\beta = 1/3$. The parameters of the coverage action are: $\alpha_M = 40$, $R = 10$. Fig. 12 shows the evolution of the coverage map throughout the 2000 units of time. During the first 800 units of time, the agents cover the domain, whereas the rest of the time, the team maintains the coverage level.

Fig. 13 shows the evolution of the total coverage error. We consider that the obstacles do not need to be covered and then the coverage domain is reduced by the obstacles. Therefore, the coverage domain is smaller and the transient time is lower. Fig. 14 presents the linear and angular velocities of each agent. In spite of including a repulsion term, the smoothness of this law produces a behavior very similar to the simulation without obstacles. Fig. 15 shows the distance between each agent and the nearest obstacle detected by the range sensor through the coverage process and Fig. 16 shows the trajectory of agents. In this figure, the obstacles have been dilated r_i^{avo} to indicate the collision area.

Both figures show that the agents develop safe coverage avoiding static obstacles and other agents.

Finally, we present a simulation with two unconnected domains to be covered. One of the domains is divided in two parts by an obstacle, and the other one is bigger and contains a nonconvex obstacle. The parameters of the problem are the same as the previous simulation except $\gamma = 3$ to allow agents to enter in narrow U-shaped obstacles and $k_{v_i} = 3$. We also give a different priority to the coverage of both zones. The priority of the larger one is $\phi = 1$ whereas the smaller one has a priority $\phi = 0.3$. Fig. 17 shows the coverage map during the first 2000 units of time, Fig. 18 shows the evolution of the total error, and Fig. 19 shows the distance between each agent and the nearest obstacle detected by the range sensor. Finally, Fig. 20 shows the trajectory of the agents. The chart shows that the larger area has a higher density of trajectories because it has higher priority. In Fig. 21 the average coverage level of both areas is presented. At the beginning of the simulation, our strategy to find global objectives detects three unconnected areas (see Fig. 17, $t=1$). Two of them are in the zone with the lower priority, and the other one is in the zone with the higher priority. Since there are three agents on the team, each agent moves towards one of the objectives and initially, the area with lower priority, which is also smaller, is better covered than the area with higher priority. However, once there are new global objectives (see Fig. 17, $t=300$), and more than one global objective is found in the bigger area, the global objectives with a higher coverage error attract the agents. From that moment on, the area with higher priority maintains a coverage level around the objective whereas the area with lower priority has a lower coverage level as expected. In spite of the significant difference in the coverage priority, the difference in the coverage level is not so high. In this case, the team has a high coverage ability and once the higher priority area has a coverage level near the objective there is no error to attract the agents. Then, agents go toward the lower priority area according to the coverage error. With a lower coverage ability of the team of agents, by reducing the number of agents, their speed of coverage action, the difference between the coverage levels of both areas is higher. As it is shown in the figures, our methodology is effective in the coverage of unconnected domains due to the blob based strategy and it can also avoid nonconvex obstacles due to the application of the modified tangent-bug algorithm presented in Section 3.2. It is also sensitive to the priorities of different areas by developing a better coverage of the areas with higher priority. Additionally, three videos of the simulations presented in Figs. 5, 12 and 17 are provided.¹

¹ <http://webdiis.unizar.es/%7Eglopez/coverage.html>

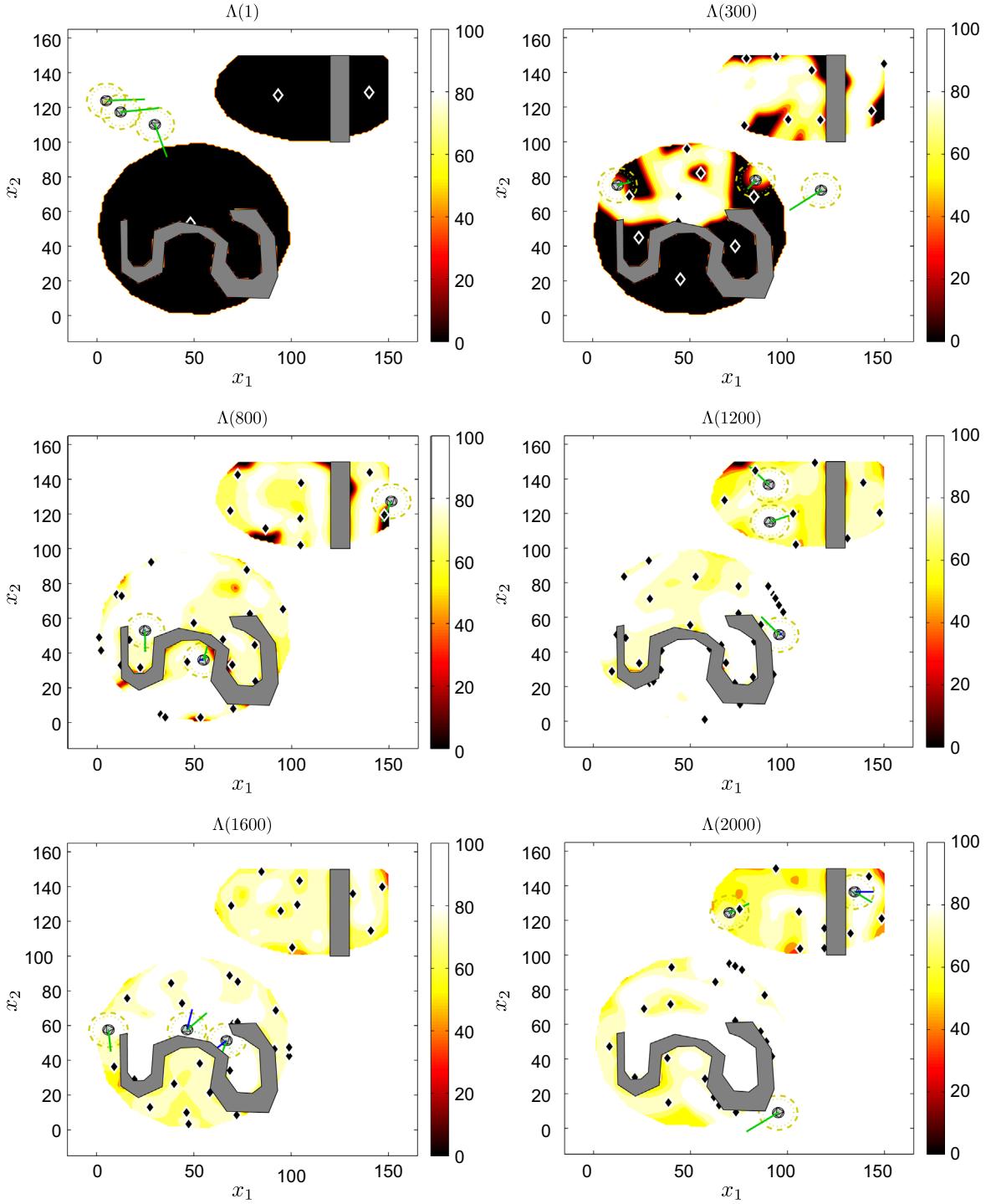


Fig. 17. Evolution of the coverage map with nonconvex obstacles and unconnected domains. Small circles represent the positions of the agents, their coverage domain is represented by a dashed line, and the avoidance regions are represented by the thin dotted line. Solid green straight lines represent the total action, dotted green lines represent the global coverage actions, and solid blue lines represent the repulsion actions. The small rhombi represent the global goals. The solid dark gray blocks represent the obstacles. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

There, the performance and smooth motions of the agents can be observed.

5.4. Comments on the algorithm tuning

As shown, the problem of persistent coverage developed by a team of nonholonomic agents is complex and there are many parameters involved. Throughout the paper many parameters have appeared and

the tuning of the algorithm could seem complicated. However, the only parameters that need to be adjusted are β , r_i^{avo} , γ , $k_{v_i}^{glo}$, ϕ . The rest of the parameters represent characteristics of the environment or the robots and thus are imposed and not subjected to changes. In particular, $\bar{\Lambda}$, K_s , K_d represent the environment and Λ^* represents the requirement of the problem. r_i^{saf} , k_{v_i} , k_{ω_i} are characteristics of the robots, namely their sizes, and maximum linear and angular velocities. Finally, α_i and R are characteristics of the available actuators.

Based on the theoretical analysis and simulation results provided we give some tuning recommendations. β is the parameter which allows modifying the importance of the coverage control law. With $\beta > 1$ the agents almost only obey the global control law going from one global goal to another, with $0.1 < \beta < 1$ both

strategies have influence, and with $0 < \beta < 0.1$ agents follow almost only the local control law, thus obeying a gradient strategy. Hybrid strategies, that is, the ones which take into account both global and local strategies, are the ones that achieve a more efficient coverage. If the team of agents have a high coverage capacity, for example a lawn mower where the grass is cut instantaneously and it takes several days to grow again, global strategy should has more weight and then $\beta \simeq 1$. On the other hand, if the team has lower capacity to cover the environment then $\beta \simeq 1/10$. A good compromise between both is $1/5 < \beta < 1/3$. The parameters of the collision avoidance law have to be chosen according to the size and dynamics of the agents. r_i^{avo} is the distance where the repulsion starts and then it must be equal or smaller than the coverage range R to allow covering points near obstacles. γ allows changing the slope of the intensity of the repulsion depending on the distance. If $\gamma > 1$ a soft repulsion is produced when obstacles are around r_i^{avo} of the agent i , and a strong repulsion when agents are close to obstacles, whereas $\gamma < 1$ causes already a strong repulsion around r_i^{avo} . Higher γ requires more braking capacity from the agent but also allows approaching more to obstacles, performing a better coverage of the environment. A closer approaching is also achieved as r_i^{avo} decreases but also requires more braking capacity. For k_i^{glo} we recommend values

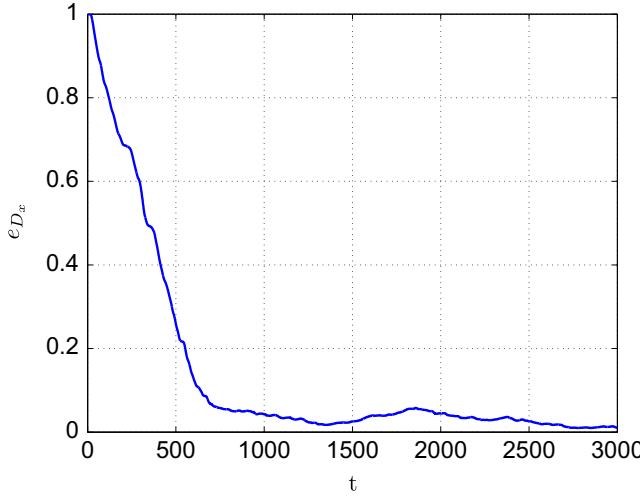


Fig. 18. Normalized coverage error evolution of the simulation of Fig. 17.

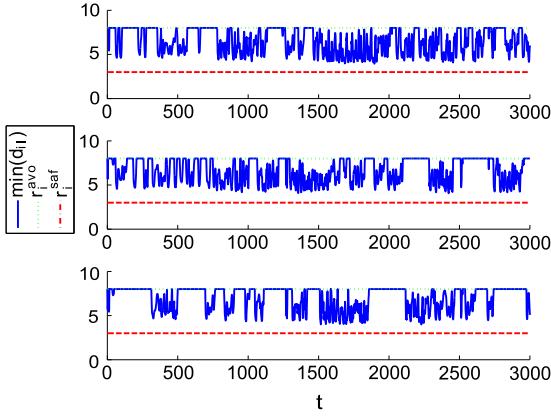


Fig. 19. Distance between the agents and the nearest obstacle detected by the range sensor at every time in the simulation presented in Fig. 17. It can be seen that there are no collisions since the distances are always greater than the safety distance of each agent r_i^{saf} .

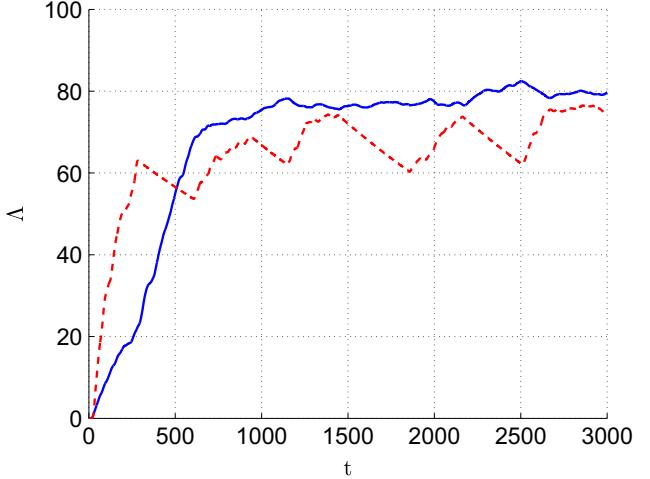


Fig. 21. Average coverage of the two areas of simulation of Fig. 17. Solid blue line represents the average coverage of the bigger area, with $\phi=1$. Dashed red line represents the average coverage of the smaller area, with $\phi=0.3$. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

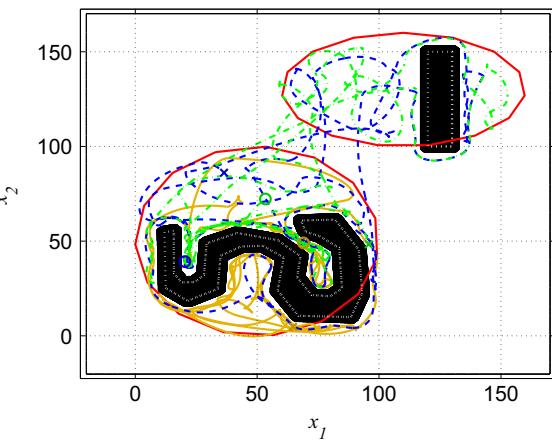
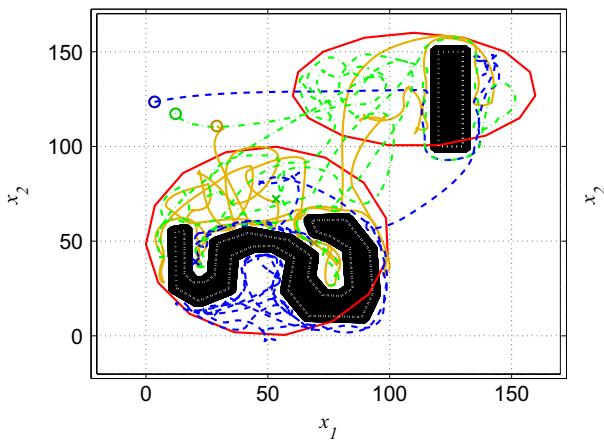


Fig. 20. Trajectories of agents from $t=1$ to $t=1500$ (left) and from $t=1500$ to $t=3000$ (right) of the simulation of Fig. 17. The paths of the agents start at circles and end at crosses. Obstacles are represented by black blocks. They have been expanded by r_i^{avo} from their actual boundaries (white dotted lines) to show the avoidance area. We divide the figures into two intervals of time for a better perception of the trajectories.

close to 1 until the distance from an agent to a goal is almost the coverage radius R , and then decrease the value. It allows the agents to reach the goals quickly, and then slow down when the goal is being accomplished. However, any other function of similar characteristics could be chosen. Finally, ϕ represents the importance of the zones to cover in such a way that constant ϕ makes the agents to cover all the domain equivalently. However, as proposed in the last of our simulations, if some area requires a better coverage, it can be selected with higher priority than the rest of the domain.

6. Conclusion

In this paper we presented the first control algorithm that develops persistent coverage with reactive avoidance control laws. This is based on a new model for the evolution of the coverage level with decay. Assuming the unicycle model for the dynamics of the agents, we provide a controller which combines local and global control laws guaranteeing full coverage of the domain if there is no decay, even when there are non-convex obstacles or unconnected domains. The controller also provides a persistent coverage if there is a coverage decay. Furthermore we proposed a new bounded repulsive avoidance control law and a strategy to combine coverage and avoidance objectives with a proof of collision avoidance. Finally, we provided simulation results showing the behaviors of the algorithm and we discussed the choices of the design parameters of the algorithm. An open issue is the problem of developing coverage with a variable power of the coverage action in order to save energy, and to keep a desired coverage level. Another research line is to develop an adaptive behavior of the parameter which rules the importance of the local and global control laws depending on the error.

Acknowledgements

This work was supported by projects DPI2012-32100, IPT-2011-1158-9200000, RTC-2014-1847-6 from Ministerio de Economía y Competitividad, by FEDER funds, and by Grant B139/2010 by DGA.

References

- [1] E.M. Arkin, S.P. Fekete, J.S.B. Mitchell, Approximation algorithms for lawn mowing and milling, *Comput. Geom.: Theory Appl.* 17 (1–2) (2000) 25–50.
- [2] J. Borenstein, Y. Koren, The vector field histogram-fast obstacle avoidance for mobile robots, *IEEE Trans. Robot. Autom.* 7 (3) (1991) 278–288.
- [3] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, D. Rus, Voronoi coverage of non-convex environments with a group of networked robots, in: *IEEE International Conference on Robotics and Automation*, 2010, pp. 4982–4989.
- [4] C.G. Cassandras, W. Li, Sensor networks and cooperative control, *Eur. J. Control* 11 (4–5) (2005) 436–463.
- [5] D. Chang, S. Shadden, J. Marsden, R. Olfati-Saber, Collision avoidance for multiple agent systems, in: *42nd IEEE Conference on Decision and Control*, 2003.
- [6] H. Choset, Coverage for robotics—a survey of recent results, *Ann. Math. Artif. Intell.* 31 (1–4) (2001) 113–126.
- [7] J. Cortés, S. Martínez, T. Karatas, F. Bullo, Coverage control for mobile sensing networks, *IEEE Trans. Robot. Autom.* 20 (2) (2004) 243–255.
- [8] D.V. Dimarogonas, S.G. Loizou, K.J. Kyriakopoulos, M.M. Zavlanos, A feedback stabilization and collision avoidance scheme for multiple independent non-point agents, *Automatica* 42 (2) (2006) 229–243.
- [9] Z. Drezner, *Facility Location: A Survey of Applications and Methods*, Springer-Verlag, New York, 1995.
- [10] H. Flanders, Differentiation under the integral sign, *Am. Math. Mon.* 80 (6) (1973) 615–627.
- [11] D. Fox, W. Burgard, S. Thrun, The dynamic window approach to collision avoidance, in: *IEEE Robotics Automation Magazine*, 1997.
- [12] C. Franco, G. López-Nicolás, C. Sagüés, S. Llorente, Persistent coverage control with variable coverage action in multi-robot environment, in: *52th IEEE Conference on Decision and Control*, 2013.
- [13] C. Franco, G. López-Nicolás, D. Stipanović, C. Sagüés, Anisotropic vision-based coverage control for mobile robots, in: *IROS Workshop on Visual Control of Mobile Robots (ViCoMoR 2012)*, 2012.
- [14] C. Franco, D. Paesa, G. López-Nicolás, C. Sagüés, S. Llorente, Hierarchical strategy for dynamic coverage, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.
- [15] A. Gusrialdi, S. Hirche, T. Hatanaka, M. Fujita, Voronoi based coverage control with anisotropic sensors, in: *American Control Conference*, 2008, pp. 736–741.
- [16] N. Hübel, S. Hirche, A. Gusrialdi, T. Hatanaka, M. Fujita, O. Sawodny, Coverage control with information decay in dynamic environments, in: *Proceedings of the IFAC*, vol. 17, 2008, pp. 4180–4185.
- [17] P. Hokayem, D. Stipanović, M. Spong, On persistent coverage control, in: *46th IEEE Conference on Decision and Control*, 2007.
- [18] I.I. Hussein, D.M. Stipanović, Effective coverage control for mobile sensor networks with guaranteed collision avoidance, *IEEE Trans. Control Syst. Technol.* 15 (4) (2007) 642–657.
- [19] P. Jones, B. Ludington, J. Reimann, G. Vachtsevanos, Intelligent control of unmanned aerial vehicles for improved autonomy, *Eur. J. Control* 13 (2–3) (2007) 320–333.
- [20] I. Kamon, E. Rivlin, E. Rimon, Tangentbug: a range-sensor based navigation algorithm, *Int. J. Robot. Res.* 17 (9) (1998) 934–953.
- [21] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, in: *IEEE International Conference on Robotics and Automation*, 1985.
- [22] D. Liu, D. Wang, G. Dissanayake, A force field method based multi-robot collaboration, in: *IEEE Conference on Robotics, Automation and Mechatronics*, 2006.
- [23] C. Luo, S. Yang, D. Stacey, Real-time path planning with deadlock avoidance of multiple cleaning robots, in: *IEEE International Conference on Robotics and Automation*, 2003.
- [24] M.J. Mataric, A. Howard, G.S. Sukhatme, Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem, in: *International Symposium on Distributed Autonomous Robotics Systems*, 2002.
- [25] J. Minguez, L. Montano, Nearness diagram (nd) navigation: collision avoidance in troublesome scenarios, *IEEE Trans. Robot. Autom.* 20 (1) (2004) 45–59.
- [26] N. Moshtagh, N. Michael, A. Jadbabaie, K. Daniilidis, Vision-based, distributed control laws for motion coordination of nonholonomic robots, *IEEE Trans. Robot.* 25 (4) (2009) 851–860.
- [27] D.O. Popa, C. Helm, H.E. Stephanou, A.C. Sanderson, Robotic deployment of sensor networks using potential fields, in: *IEEE International Conference on Robotics and Automation*, 2004, pp. 642–647.
- [28] S. Quinlan, O. Khatib, Elastic bands: connecting path planning and control, in: *IEEE International Conference on Robotics and Automation*, 1993.
- [29] M. Saska, J.S. Mejía, D. Stipanović, V. Vonásek, K. Schilling, L. Preučil, Control and navigation in manoeuvres of formations of unmanned mobile vehicles, *Eur. J. Control* 19 (2) (2013) 157–171.
- [30] S. Smith, M. Schwager, D. Rus, Persistent robotic tasks: monitoring and sweeping in changing environments, *IEEE Trans. Robot.* 28 (2) (2012) 410–426.
- [31] D.M. Stipanović, P.F. Hokayem, M.W. Spong, D.D. Šiljak, Cooperative avoidance control for multiagent systems, *ASME J. Dyn. Syst. Meas. Control* 129 (5) (2007) 699–707.
- [32] X. Zheng, S. Koenig, D. Kempe, S. Jain, Multirobot forest coverage for weighted and unweighted terrain, *IEEE Trans. Robot.* 26 (6) (2010) 1018–1031.
- [33] M. Zhong, C.G. Cassandras, Distributed coverage control and data collection with mobile sensor networks, in: *IEEE Conference on Decision and Control*, 2010, pp. 5604–5609.