Asian Journal of Control, Vol. 00, No. 0, pp. 1–14, Month 0000
Published online in Wiley InterScience (www.interscience.wiley.com) DOI: 10.1002/asjc.0000

ADAPTIVE ACTION FOR MULTI-AGENT PERSISTENT COVERAGE

C. Franco, G. López-Nicolás, C. Sagüés and S. Llorente

ABSTRACT

Persistent coverage with a team of agents refers to the task of covering an area where the coverage degrades with time. This is a dynamic task as opposed to the deployment problem. A key issue here is the coverage degradation that prevents the complete coverage fulfilment and requires persistence in the action. We propose a new method for persistent coverage where the agents' actions are computed in a partially distributed manner. The contribution is a control strategy based on variable coverage action and variable coverage range of the agents. This control provides adaptive behaviour in terms of actuator power and actuator domain in order to reduce the coverage error and energy consumption. The proposal is tested in simulation, showing clear improvement in terms of efficiency, flexibility, and scalability.

Key Words: Persistent coverage, multi-agent system, adaptive control

I. INTRODUCTION

The problem of area coverage has been widely studied during the last decades. This is a challenging problem with a great variety of approaches depending on the framework and conditions considered. The task of coverage has been classically tackled as a static problem, where a set of agents is deployed to cover an area. However, if the number of agents is not enough to cover the area all in one go, a dynamic approach is required. Moreover, we consider the task of covering an area where the achieved coverage degrades with time. Therefore, the coverage degradation prevents the total coverage fulfilment and requires persistence in the action. This dynamic and complex task, referred as persistent coverage, is still an open issue.

In terms of efficiency, a key issue in the task of persistent coverage is the consumption of energy. One way of optimizing this energy consumption is by reducing the path length travelled [2], [9], [15]. These previous works address the problem of efficient path planning for area coverage considering a single agent. A more challenging problem is considering a multi-agent setup [1]. Work in [22] also considers changing environments in the task definition, and proposes the adaptation of the agents' speed as a function of the environment changes. Another approach considered in [3], [21], and [23] consists of computing the path depending on the interest of covering each part of the domain.

These previous approaches basically perform an initial path computation to cover the whole domain. This can be efficient in general but rigid and not appropriate when long-term coverage problems or changing environments are considered. Additionally, centralised approaches are well-known to be difficult to scale. This has been addressed in the multi-robot patrolling problem with distributed techniques able to adapt to system changes with variable number of agents [20]. Other related works [18], [19] consider simultaneously the design of optimal trajectories and

C. Franco, G. López-Nicolás and C. Sagüés are with the Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain (email: cfranco@unizar.es, gonlopez@unizar.es, csagues@unizar.es)
S. Llorente is with the Research and Development Department, Induction Technology, Product Division Cookers, BSH Home Appliances Group, Zaragoza, Spain (email: sergio.llorente@bshg.com).

This work was supported by projects IPT-2011-1158-920000 of subprogram INNPACTO, RTC-2014-1847-6 of subprogram Retos-Colaboración, and by DPI2012-32100 from Ministerio de Economía y Competitividad/European Union. The first author was supported by grant B139/2010-DGA.

distributed controllers with several performance criteria such as refresh time and latency. This issue is also addressed in [8], [10], [12], or [17]. Additionally, the idea of a variable density function over the domain denoting the interest of the points to be covered is presented in [24] to produce denser paths over the areas of interest. The issue of collision avoidance with unknown obstacles and global convergence of the dynamic coverage task is also addressed with a decentralised control strategy in [25].

Another way to improve the system efficiency is to modify the range of the coverage actuators to save energy. This issue has been widely studied in the context of wireless communications for the deployment problem, where sensor or wireless networks adapt their range to save energy while they are placed strategically in the workspace optimising their coverage [4], [11], [16], [27]. However, there are still no approaches addressing the persistent area coverage problem with variable range of the actuator.

In this paper, we consider the problem of persistent coverage and we present a new method where the agents' paths and actions are computed online in a partially distributed manner. Although some works focus on uncertainty maps with intermittent communications [29] or constraining the motion to maintain the communication network [14], we consider that the coverage information of the domain is available for all the agents, whereas the coverage and motion actions are computed locally by each agent resulting in a partially distributed system. This type of topology has been previously used for instance in [13], [22], [24]. An illustrative example of this topology could be a fleet of ships cleaning up an oil spill spread into the sea. Locally they compute their motion in a distributed way while they also receive information by satellite about the distribution of the oil released to avoid get stuck in local minima. Moreover, since there are applications that require a particular coverage level (e.g. higher coverage leads to a waste of energy in cleaning tasks, or to bad results in painting tasks), we also consider adaptive action. The contribution is a control strategy based on coverage action with variable power and variable range, which allows not only to reduce the coverage error but also to save energy. On the one hand, the coverage action of the agents is variable and can be adapted depending on the coverage error in the actuator domain. On the other hand, we extend [6] and the agents' controllers can now modify its coverage range so the actuator domain can be adapted to reduce the energy consumption and the coverage error. Regarding the agents' motion controller, we combine a gradient technique for local coverage, and blob analysis to develop a global strategy [7], [6].

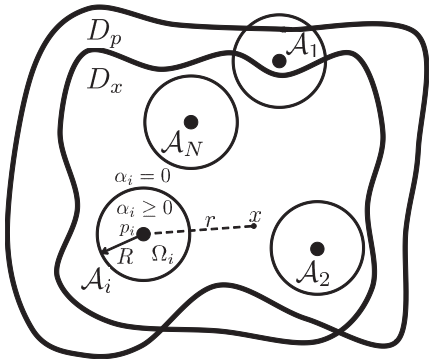


Fig. 1. Problem setup for persistent coverage. Several agents $\mathcal{A}_i$ are shown in the area to be covered D_x. The different parameters and variables are described in the text.

II. COVERAGE CONTROL

In this section, we introduce the problem formulation and present the coverage control laws to regulate the power level and coverage range of each agent's actuator to perform persistent coverage.

2.1. Problem formulation

The problem formulation is introduced by including the dependencies of the variables only when they are defined for ease of the notation. Let us consider a team of N agents $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_N\}$. We consider that mobile agents are holonomic, $\dot{p}_i = u_i$, where $p_i(t) = [p_{i1}(t), p_{i2}(t)]^T$ is the position of the agent $\mathcal{A}_i$ in a domain $D_p \subset \mathbb{R}^2$ and $u_i \in \mathbb{R}^2$ its input motion. The aim is to reach a desired coverage level $\Lambda^*(x) > 0$ for all the points $x \in D_x$ over a bounded domain $D_x \subset \mathbb{R}^2$. An example to illustrate the coverage level meaning could be a spray painting application where the coverage level is the thickness of the actual paint layer, and the desired coverage level is the required value of paint thickness. Thus, the agents paint the area and either lack or excess of painting with respect to the desired coverage level is undesirable. Note that D_p can be different from D_x being the reason why, depending on the application, the agents can actuate in an area bigger, or smaller, than the space they occupy. The problem setup is illustrated in Fig. 1.

2.2. Coverage power control

In the following we present a power control for the coverage action of the agents in order to design an

adaptive and efficient method. Let us define $r(x, p_i) \in \mathbb{R}^+ \cup \{0\}$ as the Euclidean distance between a point $x \in D_x$ and the position p_i of agent $\mathcal{A}_i$, and $\alpha_i(r) \in \mathbb{R}^+$ as the coverage action that the agent develops over points at distance r inside the coverage domain $\Omega_i(p_i, R_i)$ of each agent. We consider circular actuators with range $R_i(t) \in \mathbb{R}^+$ so that their coverage function $\alpha_i(r)$ is:

$$\begin{cases} \alpha_i \geq 0 & if \quad r < R_i \quad (x \in \Omega_i) \\ \alpha_i = 0 & if \quad r \geq R_i \quad (x \notin \Omega_i) \end{cases} .$$

The total coverage action of the team of agents is defined as $\alpha = \sum_{i \in \{1,\dots,N\}} \alpha_i$. We also define $\Lambda(t, x) \in [0, \infty)$ as the coverage developed by the team of agents over a point x at time t.

The coverage information evolution in each point x is modelled with the following differential equation:

$$\frac{\partial \Lambda}{\partial t} = A \cdot \Lambda + B \cdot \alpha \ .$$

where $A, B \in \mathbb{R}$, $A < 0$ is the state gain and $B > 0$ the input gain. Note that $A < 0$ is required for stability and it is related with the coverage decay. We assign $\Lambda(t = 0, x) = 0, \quad \forall x \in D_x$, which means that at the beginning all points are assumed as uncovered.

At this point, let us introduce $\Phi(x) \in (0, 1]$, $\forall x \in D_x$, as the priority to cover each point x. Φ is a function that weights the interest of the points in the domain to give more or less coverage priority to particular areas. The quadratic coverage error $e_{D_x}(t)$ of the domain is defined as:

$$e_{D_x} = \int_{D_x} \Phi \cdot (\Lambda^* - \Lambda)^2 dx \ . \tag{1}$$

Let us compute the evolution of the error assuming that we have one agent in order to simplify the analysis:

$$\frac{\partial e_{D_x}}{\partial t} = -2 \left[\int_{D_x} A \cdot \Phi \cdot \Lambda \cdot (\Lambda^* - \Lambda) dx + \int_{D_x} B \cdot \alpha_i \cdot \Phi \cdot (\Lambda^* - \Lambda) dx \right] .$$

The first term is the coverage decay that drives Λ towards 0, i.e. increasing the coverage error in general. In the second term, α_i can be designed to make it always positive or null in such a way that the error decreases, or at least increases as less as possible.

Let us define $\sigma_i(r)$ as the normalized action of the agent such that:

$$\int_{\Omega_i} \sigma_i dx = 1 \ . \tag{2}$$

Notice that, for example, a lawn mower may be modelled by a constant σ_i, whereas a spray painter may be modelled by a decreasing function with radius r. We now propose to define the action of an agent as:

$$\alpha_i = K \cdot \sigma_i \ , \tag{3}$$

$$K = C \cdot \left[\int_{\Omega_i} B \cdot \Phi \cdot \sigma_i \cdot (\Lambda^* - \Lambda) dx \right]^{2 \cdot q - 1} , \tag{4}$$

where $C > 0$ is the control gain, $q \in \mathbb{N}$, and the powered term can be interpreted as the weighted error of the agent's domain. This is the formulation of a proportional controller that considers the distribution of the robots' action in order to increase the performance. Note that $K \in \mathbb{R}$ and that $\sigma_i = 0, \ \forall x \notin \Omega_i$. In this way we obtain:

$$\frac{\partial e_{D_x}}{\partial t} = -2 \left[\int_{D_x} A \cdot \Phi \cdot \Lambda \cdot (\Lambda^* - \Lambda) dx + C \cdot \left(\int_{\Omega_i} B \cdot \sigma_i \cdot \Phi \cdot (\Lambda^* - \Lambda) dx \right)^{2 \cdot q} \right] . \tag{5}$$

The change of the integration domain from D_x, which is constant, to Ω_i, which is variable, is done taking into account the results of [5] and [26] for differentiation of double integrals depending on a parameter and the assumptions over the coverage function ($\alpha_i(r) = 0, \ \forall r \geq R$). Therefore, the second term is never negative and the agents reduce the error according to the shape of the coverage action. Now let us discuss the evolution of the quadratic error inside the coverage domain of agents.

Let us first define the quadratic coverage error of the agents domain $e_{\Omega_i}(t, \Lambda)$ as:

$$e_{\Omega_i} = \int_{\Omega_i} \Phi \cdot (\Lambda^* - \Lambda)^2 dx \ . \tag{6}$$

Proposition 1 *Consider the persistent coverage task carried out by a team of agents $\mathcal{A}$ under the coverage control law defined in (3) and (4), with $\Phi(x) = 1, \ \forall x \in D_x$. Then, the quadratic error inside the coverage domain of agents e_{Ω_i} results in a non-positive derivative. In particular, simplifying the analysis to a static case, it can be seen that the error is bounded if $A^{-1} \cdot B \cdot C \int_{\Omega_i} \sigma_i^2 dx \neq 1$ and it is in inverse relation to the control gain C.*

Proof: Considering $q = 1$ in (5) without loss of generality, we can compute:

$$\frac{\partial e_{\Omega_i}}{\partial t} = -2 \cdot \Phi \cdot A \int_{\Omega_i} \Lambda \cdot (\Lambda^* - \Lambda) dx - 2 \cdot \Phi^2 \cdot B^2 \cdot C \cdot \left(\int_{\Omega_i} \sigma_i \cdot (\Lambda^* - \Lambda) dx \right)^2 . \tag{7}$$

Second term in (7) is non-positive, which guarantees that the coverage action does not increase the error. However, the first term can be positive or negative. Let us compute $\Lambda_{max}(x)$ that maximizes (7) and then, $\partial e_{\Omega_i}/\partial t(\Lambda_{max}) \geq \partial e_{\Omega_i}/\partial t(\Lambda)$:

$$\frac{\partial^2 e_{\Omega_i}}{\partial t \partial \Lambda}(\Lambda_{max}) = 0 \,, \tag{8}$$

$$\frac{\partial^3 e_{\Omega_i}}{\partial t \partial^2 \Lambda}(\Lambda_{max}) < 0 \,. \tag{9}$$

We begin by computing (8):

$$\frac{\partial^2 e_{\Omega_i}}{\partial t \partial \Lambda} = -2 \cdot \Phi \left[A \int_{\Omega_i} (\Lambda^* - 2 \cdot \Lambda) dx \right.$$
$$\left. -2 \cdot \Phi \cdot B^2 \cdot C \int_{\Omega_i} \sigma_i \cdot (\Lambda^* - \Lambda) dx \right] = 0 \,.$$

Reordering and taking derivatives with respect to x:

$$A \cdot (\Lambda^* - 2 \cdot \Lambda) = 2 \cdot \Phi \cdot B^2 \cdot C \cdot \sigma_i \cdot (\Lambda^* - \Lambda) \,,$$

and thus:

$$\Lambda_{max} = \frac{A \cdot \Lambda^* - 2 \cdot \Phi \cdot B^2 \cdot C \cdot \sigma_i \cdot \Lambda^*}{2 \cdot A - 2 \cdot \Phi \cdot B^2 \cdot C \cdot \sigma_i} .$$

Notice that in this equation both Λ_{max} and σ_i depends on x. Note also that $\Lambda_{max} \in (\Lambda^*/2, \Lambda^*)$. Next, we compute (9):

$$\frac{\partial^3 e_{\Omega_i}}{\partial t \partial^2 \Lambda} = -4 \cdot \Phi \cdot \left(\Phi \cdot B^2 \cdot C - A \int_{\Omega_i} dx \right) ,$$

which is always negative and guarantees that Λ_{max} makes the derivative of the error maximum. Negative sign of $(\Lambda^* - \Lambda_{max})$ in (7) guarantees that $\partial e_{\Omega_i}/\partial t < 0$, $\forall \Lambda$. Since this is a proportional control law, there is an offset (or steady error) which prevents $\partial e_{\Omega_i}/\partial t(\Lambda_{max}) < 0$ in the worst case with $\Lambda = \Lambda_{max}$. Higher control gain C reduces the offset and, in the limit, when $C \to \infty$, $\Lambda_{max} \to \Lambda^*$, and substituting into (7), $\partial e_{\Omega_i}/\partial t \to 0$ which results in a non-positive derivative of the error. Thus, after computing the worst case, we obtain similar conclusions to the well known results of proportional controllers dealing with linear systems.

Now, it would be interesting to find bounds to the offset of the controller. However, this is a complex task in the general case with the agents moving, since the amount of coverage inside the agent's domain is varying in an unknown way. Thus, let us assume in the following analysis that the agent is static (denoted with superindex s). This assumption is closer to reality if the motion of the agents is slow and the coverage level varies smoothly in the space. Otherwise, without this assumption there is no analytical solution. We can compute the coverage distribution of the agent domain $\Lambda^s(x \in \Omega_i, t)$ including (3) and (4) in (1) with $\Phi = 1$, $q = 1$ and assuming $\Lambda(x, 0) = 0$:

$$\frac{\partial \Lambda}{\partial t} = A \cdot \Lambda + B \cdot C \cdot \sigma_i \int_{\Omega_i} \sigma_i \cdot (\Lambda^* - \Lambda) dx$$
$$= A \cdot \Lambda + B \cdot C \cdot \sigma_i \cdot \left(\Lambda^* - \int_{\Omega_i} \sigma_i \cdot \Lambda dx \right) . \tag{10}$$

The coverage level $\Lambda(t, x)$ depends on time and space, and with the previous assumption we will obtain Λ^s that can be written as $\Lambda^s(t, x) = \eta(t) \cdot \sigma_i(x)$. In order to solve the previous differential equation we carry out the change of variable to η:

$$\frac{\partial \eta}{\partial t} = \eta \left(A - B \cdot C \cdot \int_{\Omega_i} \sigma_i^2 dx \right) + B \cdot C \cdot \Lambda^* \,. \tag{11}$$

Solving this first order differential equation:

$$\eta = \frac{B \cdot C \cdot \Lambda^*}{e^{-(A - B \cdot C \int_{\Omega_i} \sigma_i^2 dx)}} \int_0^t e^{-(A - B \cdot C \int_{\Omega_i} \sigma_i^2 dx)} dt \,.$$

Finally, computing the integral in time and undoing the change of variable, we have:

$$\Lambda^s = \frac{B \cdot C \cdot \sigma_i \cdot \Lambda^* \cdot \left(e^{(A - B \cdot C \int_{\Omega_i} \sigma_i^2 dx) t} - 1 \right)}{A - B \cdot C \int_{\Omega_i} \sigma_i^2 dx} \,.$$

and then, we can compute the error evolution $e^s_{\Omega_i}(\Lambda^s, t)$:

$$e^s_{\Omega_i} = \int_{\Omega_i} \Phi \cdot (\Lambda^* - \Lambda^s)^2 dx \,,$$

and the derivative over time:

$$\frac{\partial e^s_{\Omega_i}}{\partial t} = 2 \cdot B \cdot C \cdot \Phi \cdot {\Lambda^*}^2 \cdot e^{(A - B \cdot C \int_{\Omega_i} \sigma_i^2 dx) t}$$
$$\cdot \left(A - 2 \cdot B \cdot C \cdot e^{(A - B \cdot C \int_{\Omega_i} \sigma_i^2 dx)} \int_{\Omega_i} \sigma_i^2 dx \right) ,$$

which is non-positive. We can also compute the steady value of the coverage ($t \to \infty$) in the static case $\Lambda^{ss}(x)$:

$$\Lambda^{ss} = \frac{-A^{-1} \cdot B \cdot C \cdot \sigma_i \cdot \Lambda^*}{1 - A^{-1} \cdot B \cdot C \int_{\Omega_i} \sigma_i^2 dx} \,.$$

In that case, the steady error in the agent's domain is:

$$e^{ss}_{\Omega_i} = \int_{\Omega_i} \Phi \cdot (\Lambda^* - \Lambda^{ss})^2 dx$$
$$= {\Lambda^*}^2 \int_{\Omega_i} \Phi \cdot \left(1 - \frac{-A^{-1} \cdot B \cdot C \cdot \sigma_i}{1 - A^{-1} \cdot B \cdot C \int_{\Omega_i} \sigma_i^2 dx} \right)^2 dx, \tag{12}$$

which is the offset of the proportional controller in the static case. Taking the derivative with respect to C:

$$\frac{\partial e_{\Omega_i}^{ss}}{\partial C} = \frac{-2 \cdot A^{-1} \cdot B \cdot \Phi \cdot {\Lambda^*}^2}{\left(A^{-1} \cdot B \cdot C \int_{\Omega_i} \sigma_i^2 dx - 1\right)^3},$$

which is non-positive and implies that increasing the controller gain results in a decreasing of the offset.

The offset also depends on the distribution of σ_i. In order to analyse this influence, let us introduce a coverage function with cylindrical shape $\sigma_i^c(r)$:

$$\sigma_i^c = \begin{cases} 1/(\pi \cdot r_m^2) & if \quad 0 \leq r \leq r_m \\ 0 & if \quad r_m < r \end{cases}, \qquad (13)$$

where $r \in \mathbb{R}^+$ is $r = \|x - p_i(t)\|$, and $r_m \in [0, R)$ is the actuator percentage of the range where the action is not null, being $R \in \mathbb{R}^+$ the maximum actuator range. This function with cylindrical shape presents a constant coverage in $r = [0, r_m]$ and null in $r = [r_m, R)$. Low values of r_m produce sharper action whereas values of r_m close to R results in a uniform action throughout the range. Including coverage function (13) in (12) the steady error in the agent's domain becomes:

$$e_{\Omega_i}^{ss} = \pi \cdot {\Lambda^*}^2 \cdot \Phi \cdot \left(R^2 - r_m^2 + \frac{\pi^2 \cdot r_m^6}{\left(A^{-1} \cdot B \cdot C - \pi \cdot r_m^2\right)^2}\right).$$

Taking the derivative with respect to r_m:

$$\frac{\partial e_{\Omega_i}^{ss}}{\partial r_m} = 2 \cdot \pi \cdot \Phi \cdot r_m \cdot (\Lambda^* \cdot A^{-1} \cdot B \cdot C)^2 \cdot \frac{\left(3\,\pi\, r_m^2 - A^{-1} \cdot B \cdot C\right)}{\left(A^{-1} \cdot B \cdot C - \pi\, r_m^2\right)^3},$$

which is always negative and therefore, the steady state static error (12) with respect to the maximum error decreases as the action shape becomes more uniform throughout the range.

Summing up, we have proposed a proportional controller whose error is bounded if $A^{-1} \cdot B \cdot C \int_{\Omega_i} \sigma_i^2 dx \neq 1$ (which is easy to achieve by tuning C) and whose offset decreases by increasing the gain of the controller C. The offset also depends on the distribution of the covering function. With plane coverage actions offset is near to 0 and with sharp actions it grows. This effect can be modified with the motion controller since agents adapt their positions to the error distribution of the domain. ■

2.3. Coverage range control

In this section, we propose a novel controller for the agents' coverage range to increase the performance of the coverage and to reduce the power consumption. It is based on an adaptive coverage action range R as well as the adaptive coverage power level introduced in the previous section. Let us first obtain the second derivative of the error over time:

$$\frac{\partial^2 e_{D_x}}{\partial t^2} = -2 \left[\int_{D_x} A \cdot \Phi \cdot \frac{\partial \Lambda}{\partial t} \cdot (\Lambda^* - 2 \cdot \Lambda) dx + 2 \cdot q \cdot C \cdot \left(\int_{\Omega_i} B \cdot \Phi \cdot \sigma_i \cdot (\Lambda^* - \Lambda) dx \right)^{2 \cdot q - 1} \cdot \int_{\Omega_i} B \cdot \Phi \cdot \left(\frac{\partial \sigma_i}{\partial R} \cdot \frac{\partial R}{\partial t} \cdot (\Lambda^* - \Lambda) - \sigma_i \cdot \frac{\partial \Lambda}{\partial t} \right) dx \right], \qquad (14)$$

where the partial derivative of the range with respect to time can be tuned to make the corresponding term always positive. Then, we design the range control by imposing the coverage range evolution as:

$$\frac{\partial R}{\partial t} = k_i^R \int_{\Omega_i} B \cdot \Phi \cdot \sigma_i \cdot (\Lambda^* - \Lambda) dx \cdot \int_{\Omega_i} B \cdot \Phi \cdot \frac{\partial \sigma_i}{\partial R} \cdot (\Lambda^* - \Lambda) dx, \qquad (15)$$

with $k_i^R \in \mathbb{R}^+$ the range controller gain. We analyse the effect of the range control by replacing (15) into (14):

$$\frac{\partial^2 e_{D_x}}{\partial t^2} = -2 \left[\int_{D_x} A \cdot \Phi \cdot \frac{\partial \Lambda}{\partial t} \cdot (\Lambda^* - 2 \cdot \Lambda) dx - 2 \cdot q \cdot C \cdot \left(\int_{\Omega_i} B \cdot \Phi \cdot \sigma_i \cdot (\Lambda^* - \Lambda) dx \right)^{2 \cdot q - 1} \cdot \int_{\Omega_i} B \cdot \Phi \cdot \sigma_i \cdot \frac{\partial \Lambda}{\partial t} dx \right] - 4 \cdot k_i^R \cdot q \cdot C \cdot \left(\int_{\Omega_i} B \cdot \Phi \cdot \sigma_i \cdot (\Lambda^* - \Lambda) dx \right)^{2 \cdot q} \cdot \left(\int_{\Omega_i} B \cdot \Phi \cdot \frac{\partial \sigma_i}{\partial R} \cdot (\Lambda^* - \Lambda) dx \right)^2.$$

It can be seen that the second term of the addend is always negative and the first one depends on the conditions of the problem. Thus, the aim of the range controller is to make the second derivative of the error as low as possible to reduce the coverage error.

III. MOTION STRATEGY

In this section, we present the motion control law for coverage that relies in a local gradient-based approach with a global strategy extending ideas previously presented [6], [7], [8].

3.1. Motion control

The objective of our motion control law is to keep the coverage error decreasing. Thus, we want to minimize the variation of each agent's error computed in (5) with respect to its own position. Then, we compute $u_i^L(t)$ as the gradient of the variation of the error (right-hand side of (5)) with respect to the position of agent i:

$$u_i^L = -4 \cdot q \cdot C \cdot \left(\int_{\Omega_i} B \cdot \sigma_i \cdot \Phi \cdot (\Lambda^* - \Lambda) dx \right)^{2 \cdot q - 1} \cdot \int_{\Omega_i} B \cdot \Phi \cdot \frac{\partial \sigma_i}{\partial r} \cdot \frac{p_i - x}{\|p_i - x\|} \cdot (\Lambda^* - \Lambda) dx \; .$$

From this gradient we can extract the direction of the motion, $\hat{u}_i^L(t)$, to get the maximum benefit covering the neighbourhood of the agent:

$$\hat{u}_i^L = \begin{cases} u_i^L / \|u_i^L\| & if \; \|u_i^L\| \neq 0 \\ 0 & otherwise \end{cases} ,$$

However, when the error in the coverage domain is low, the benefit of covering the neighbourhood of the agent is small. Moreover, gradient techniques are known to get stuck in local minima. Thus, we propose to combine the gradient strategy with a global law $u_i^G(t)$ that depends on the coverage level of the whole domain to bring the agents to areas with higher error.

In order to drive the agents to other areas with higher error we propose a strategy to select global objectives $p_i^o(t) \in D_p$, which is developed in Section 3.2, and a control law to reach them. Let us define $d_i^o(t)$ as the Euclidean distance between the agent and objective positions, and a global gain $k_i^G(d_i^o) \in [0, 1]$ as:

$$k_i^G = tanh\left(\frac{2 \cdot d_i^o}{R} \right), \tag{16}$$

where R is the range of the coverage action. This function is close to 1 until the distance to the objective is in the range of the coverage actuator and then it decreases to 0 when the agent is on the objective. Finally, the global control law is defined as:

$$u_i^G = k_i^G \cdot \frac{p_i - p_i^o}{\|p_i - p_i^o\|}.$$

In order to combine global and local control laws, let us introduce the local error $e_{\Omega_i}^L(t) \in [0, 1]$ as:

$$e_{\Omega_i}^L = \max\left(0, \int_{\Omega_i} \Phi \cdot \sigma_i \cdot \frac{(\Lambda^* - \Lambda)}{\Lambda^*} dx \right) .$$

Compared with (6), this is a more qualitative error measurement indicating that agent's vicinity is satisfactorily covered when it is negative or null. Let us also introduce a local $W_i^L(t)$ and global weight $W_i^G(t)$:

$$W_i^L = (e_{\Omega_i}^L)^\beta \cdot (1 - e_{\Omega_i}^L) \; , \tag{17}$$

$$W_i^G = 1 - (e_{\Omega_i}^L)^\beta \cdot (1 - e_{\Omega_i}^L) \; , \tag{18}$$

where $\beta \in \mathbb{R}^+$ is a design parameter. Finally, we compute the motion control law $u_i(t)$ as follows:

$$u_i = k_i \cdot (W_i^L \cdot \hat{u}_i^L + W_i^G \cdot u_i^G) \; , \tag{19}$$

where $k_i \in \mathbb{R}$ is the motion gain and represents the maximum velocity of each robot. The term $(1 - e_{\Omega_i}^L)$ in (17) and (18) slows down the agents to develop coverage in the surroundings when the local error is high, and speeds up the agent when the error is close to 0 to leave the area. Note also that the weights make the agents to obey local control law, when the local error is high, moving slowly in the direction of the gradient of the error and performing coverage thoroughly, and make the agents to obey global control law when the local error is low, heading towards new uncovered areas rapidly to increase the performance of the coverage. This combination of local and global strategies was firstly presented in [8] where a proof of total coverage was given for environments without decay. In this case, reaching the total coverage of the area is not possible since agents are not able to cover all the domain simultaneously, but as it will be shown in the simulations a steady error is eventually reached.

3.2. Selection of global objectives

The motion strategy defined in the previous section is compounded by local and global components in such a way that the problem of local minima with gradient methods is avoided. Note that the computation of the control actions by the agents is decentralized despite the usage of global information. The global strategy requires the assignment of global objectives p_i^o to each agent. This strategy is based on blob detection of the uncovered information as previously introduced in [6]. We use this processing algorithm to find islands of uncovered information in the map Λ, and then we compute their sizes and their centroids. With this

information we use a similar criterion as in [7] to select the objective which is based on the uncertainty and the proximity of the blobs.

Let us define $\Psi(t) = \{\psi_1, \psi_2, ..., \psi_j, ..., \psi_M\}$ as the collection of $M(t)$ global objectives, $\psi_j(t) \in D_x$. Let us also define $\pi_j(t)$ as the collection of points of the domain composing each blob and whose global objective is $\psi_j(t)$, $\Pi(t) = \bigcup_{j=1}^{M} \pi_j$ as the collection of points of the domain assigned to objectives ψ_j. Finally, let us introduce $\mathfrak{T}$ as the tolerance or percentage of admissible error, and $\pi^{\emptyset}(t) = \{x \in D_x | (\Lambda^* - \Lambda) \leq \mathfrak{T} \cdot \Lambda^*\}$ as the collection of points that are covered.

The method proposed starts by checking if some of the M global objectives ψ_j have been covered. Those covered are discarded from the list of global objectives Ψ and the corresponding points π_j are released. It also erases objectives that are closer than a distance R, which is the coverage radius of the agents. Afterwards, the domain to obtain the new blobs of the scene is computed by subtracting the covered points $\pi^{\emptyset}$ and the assigned points Π from the domain to cover D_x. Using any standard blob detection algorithm of computer vision the candidate centroids ψ^f and the points π^f of the F regions of the space to be covered are obtained. Then, we check if the candidate centroids ψ^f belong to the points π^f of the blob. If the centroids ψ^f are inside the blob, they and the points of the blobs π^f are saved, and the points π^f are erased of the blob domain.

From the blob-based set of global objectives obtained as candidates we need to assign them to the agents. The procedure presented is similar to the algorithm introduced in [7], The difference is that in [7] the assignment only weighted distances to centroids, whereas now, we propose to weight distances and blob coverage errors in order to improve performance. Let us compute the coverage error of the blobs assigned to each centroid $e_{\pi_j}(t)$ as:

$$e_{\pi_j} = \int_{\pi_j} \Phi \cdot (\Lambda^* - \Lambda) dx \ . \tag{20}$$

The choice of the objective p_i^o for each agent i is done with a criterion that weights distance to the centroids, and their coverage error. Notice that, contrary to (1), the coverage error error in (20) is not quadratic, keeping the error sign in such a way that the lack of coverage will increase the attraction weight of the blobs, whereas the excess of coverage will reduce their weight. 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:

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

Then, with these scores, the global objectives are assigned using the following algorithm that matches a global objective 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.

IV. SIMULATION RESULTS

In this section, we present simulation results of the control strategy proposed. First, let us introduce some performance metrics: the average absolute error $\hat{e}_{D_x}(t) = \sqrt{e_{D_x}} / \int_{D_x} \Phi dx$, the integral average error $\bar{e}_{D_x}(t) = \int_0^t \hat{e}_{D_x} dt/t$, and the integral average power consumption $\bar{\alpha}(t) = \int_0^t \int_{D_x} \alpha \, dx \, dt/t$. Let us introduce the coverage function $\sigma_i^p(\hat{r})$ [8]:

$$\sigma_i^p = \begin{cases} \sigma_M \quad \text{, if} \quad 0 \leq \hat{r} \leq r_m \\ \sigma_M \cdot \left(1 + 2 \cdot r_p^3 - 3 \cdot r_p^2\right) \quad \text{, if } r_m \leq \hat{r} < 1 \\ 0 \quad \text{, if} \quad \hat{r} \geq 1 \end{cases} ,$$

where $r_p = (\hat{r} - r_m)/(1 - r_m)$, $\hat{r} \in \mathbb{R}^+$ is $\hat{r} = \|x - p_i(t)\|/R$, being $R \in \mathbb{R}^+$ the total actuator range, and $r_m \in [0, 1)$ the actuator range percentage where the action is maximum. The maximum action $\sigma_M \in \mathbb{R}$ is defined to fulfil the requirement of (2) as:

$$\sigma_M = \frac{10}{\pi \cdot R^2 \cdot (3 \cdot r_m^2 + 4 \cdot r_m + 3)} \ .$$

This function with plateau shape presents a constant maximum coverage in $\hat{r} = [0, r_m]$ and decreases from $\hat{r} = r_m$ to $\hat{r} = 1$. With $r_m \approx 1$ the coverage is uniform throughout the range, whereas $r_m \approx 0$ produces a sharper action. We also introduce a quadratic coverage function $\sigma_i^q(r)$ previously used by other authors [13], [28], [24], [26]:

$$\sigma_i^q = \begin{cases} \frac{3}{\pi \cdot R^6} \cdot (r^2 - R^2)^2 & r \leq R \\ 0 & r > R \end{cases} , \tag{21}$$

where r, as commented before, is the Euclidean distance between a point x and the position of agent

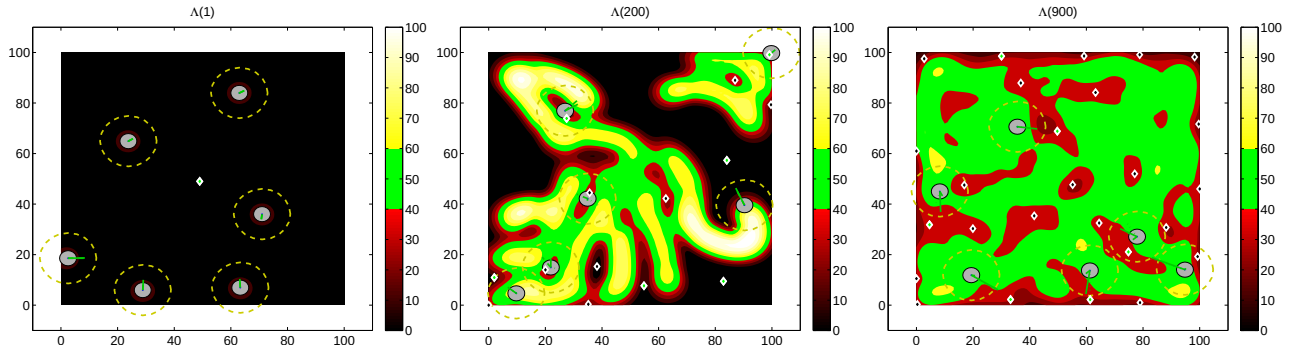
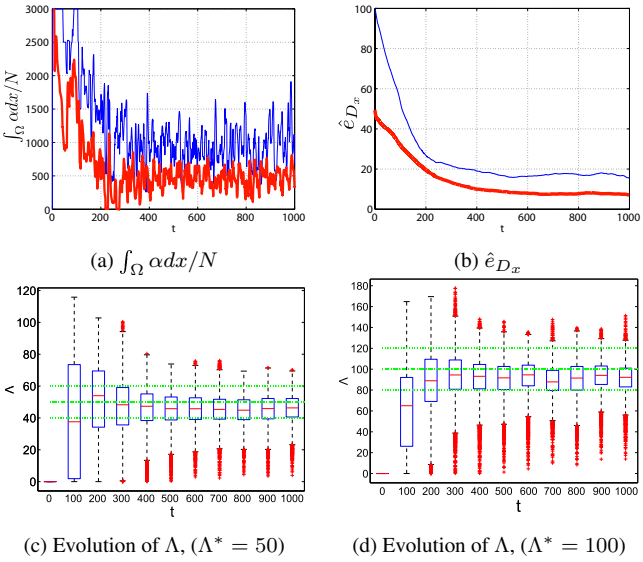


Fig. 2. Evolution of the coverage map of simulation in Section 4.1.1 with $\Lambda^* = 50$ (green color in the plots). The agents are represented by small circles and their coverage domain is represented by a dashed circle. The small rhombi represent the global objectives.

p_i. Note that the flexibility of the approach presented allows the use of a great variety of sensors/actuators, and the user can model different coverage functions adapted to a particular problems or applications. Let us also introduce $\tau = -1/A$ and $g = B \cdot \tau$ as the time constant and the gain of the dynamic coverage of the domain. Higher τ means a slower evolution of the domain and vice versa.

In the simulations, the space length discretization is 1 unit and the period discretization is 0.1 units of time. We have computed the integrations over time using the first order Euler scheme, and integrations over the domain using the trapezoidal method. The agents are modelled as vehicles with holonomic kinematics. In the following simulations we test different values of the parameters to analyse the behaviour of the control algorithms. The coverage domain D_x is a 100×100 units square area with constant coverage priority $\Phi = 1$ $\forall x \in D_x$ and gain $g = 1/2$. The team is composed of 6 agents with a saturated coverage power $K \in [0, 3000]$, $C = 300/B$ and with $q = 1$. The agents move over $D_p = \mathbb{R}^2$ with $\beta = 1/3$.



(a) $\int_\Omega \alpha dx/N$ (b) $\hat{e}_{D_x}$

(c) Evolution of Λ, ($\Lambda^* = 50$) (d) Evolution of Λ, ($\Lambda^* = 100$)

Fig. 3. Results of simulations in Section 4.1.1. (a) Evolution of the average action of the agents. (b) Evolution of the average absolute error $\hat{e}_{D_x}$ with $\Lambda^* = 50$ (thick red line) and $\Lambda^* = 100$ (thin blue line). Boxplot of the distribution of the coverage at several different times with $\Lambda^* = 50$ (c) and $\Lambda^* = 100$ (d).

4.1. Variable coverage power

In the following simulations the coverage function σ_i^q (21) is used together with $k_i = 3$, $R = 10$, $k_i^R = 0$.

4.1.1. Variable coverage power with different Λ^*

We first test the coverage performance when using variable coverage power but constant coverage range. We present a simulation in detail with $\Lambda^* = 50$ and $\tau = 400$ and compare the results to the case of $\Lambda^* = 100$. The evolution of the coverage map is shown in Fig. 2. The evolution of the average action of the agents ($\int_\Omega \alpha dx/N$), average absolute coverage error ($\hat{e}_{D_x}$), and a boxplot of the histogram of the coverage evolution is shown in Fig. 3.

Approximately in 500 seconds, the errors reach steady values with absolute average errors of $\hat{e}_{D_x}(\Lambda^* = 50) \approx 9$ and $\hat{e}_{D_x}(\Lambda^* = 100) \approx 18$. It can be seen that the power consumption doubles when increasing the desired coverage level from $\Lambda^* = 50$ to $\Lambda^* = 100$. Therefore, the algorithm can adapt the power to the new objective, achieving a similar percentage of error by doubling the coverage power. The average coverage power consumptions are $\bar{\alpha}(\Lambda^* = 50) = 638$, and $\bar{\alpha}(\Lambda^* = 100) = 1235$. The boxplot chart (Fig. 3) and the coverage maps (Fig. 2) show that most part

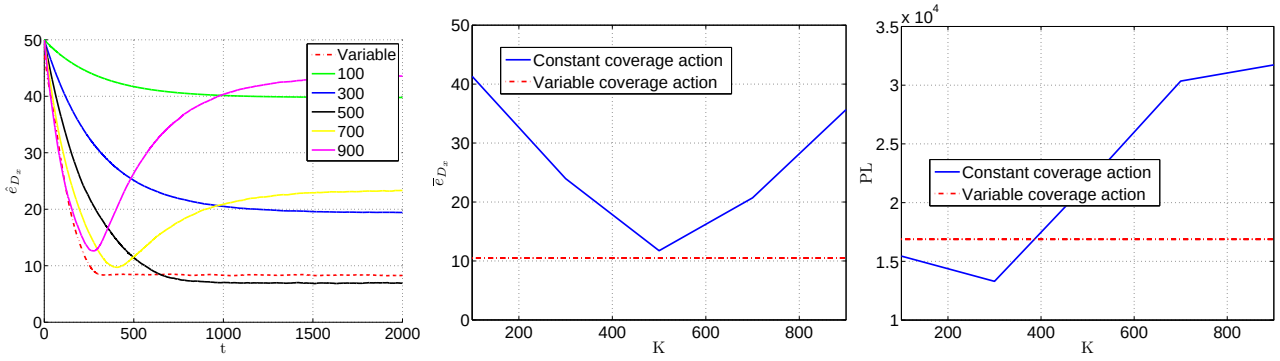


Fig. 4. Aggregated results of simulation in Section 4.1.2. From left to right, average error over time $\hat{e}_{D_x}$, integrated average error $\bar{e}_{D_x}$ and average path length (PL). The coverage objective is $\Lambda^* = 50$ and the resultant average power over time of the proposed variable coverage power control law is $\bar{\alpha} = 499$.

of the domain is finally around the coverage objective, although there are also points over and below the desired coverage level. This is due to the shape of the coverage action and the coverage decay that prevents reaching and keeping the coverage objective with zero error in all the domain simultaneously.

4.1.2. Constant vs variable coverage power

In this simulation, we present a comparison with $\Lambda^* = 50$ and $\tau = 200$ between our proposal with variable coverage power and the same without this feature in order to show the benefits of the variable coverage power approach with respect the constant one. We perform 100 simulations starting at random positions with the same variables as the previous simulation, and 100 more with each one of the following values of the coverage power: $K = \{100, 300, 500, 700, 900\}$. In Fig. 4 we show the results. The evolution of the error with time shows that the variable coverage is the fastest in reaching the steady state with a final value of $\hat{e}_{D_x} = 8$ (Fig. 4 left). In this case, the only constant power that reaches a lower value is $K = 500$, which is a very similar value to the resultant average coverage $\bar{\alpha}$ of our variable coverage, which is 499. Over $K = 500$, the error reaches a minimum and then it grows since the domain becomes over-covered. By integrating the error over time $\bar{e}_{D_x}$ (Fig. 4 centre), the variable coverage has the lowest value. Finally, in Fig. 4 right, it can be seen that the average path length (PL) with coverage power results in shorter path lengths. This is due to the term $(1 - e^L_{\Omega_i})$ in (19) which slows down agents when their domains are uncovered, whereas agents speed up when the domain is covered or over-covered and then the path length grows.

4.2. Variable coverage power and range

In this section, we study the improvement of performing the coverage with a variable range actuator. In the following simulations the desired coverage level is $\Lambda^* = 50$ and the coverage function σ^p_i (21) is used together with following parameters: $k_i = \{1, 3, ..., 9\}$, $r_m = 0.8$, $R \in [10, 50]$, $k^R_i = 0.1/B^2$. The following simulations test different combinations of parameters. Each result is obtained from mean values of 200 simulations.

4.2.1. Simulation with variable power and range

First, we introduce an experiment in detail showing the results using our approach with $\tau = 200$, $k_i = 3$, and range limits $R \in [5, 15]$. In Fig. 5 we show the evolution of the coverage map. In Fig. 6 we show the evolution of the coverage power ($\int_\Omega \alpha dx/N$), the average absolute coverage error ($\hat{e}_{D_x}$), a boxplot of the histogram of the evolution of the coverage, and the evolution of the actuator range of the agents throughout the simulation. In 400 units of time the steady state is reached with an error $\hat{e}_{D_x} \approx 8$, and average coverage power consumption $\bar{\alpha} = 641$. As the domain is being covered the agents adapt their ranges as expected to the coverage error of the domain. The resultant average radius R of the agents computed by the control law during the simulation is 9.99.

4.2.2. Simulation with high decay and sharp action

Here we study the ability of the algorithm to adapt their position and range in an environment that varies rapidly ($\tau = 200$) with a sharp action ($r_m = 0.1$). Results are shown in Fig. 7 with constant range (left column) and variable range (right column). In the case

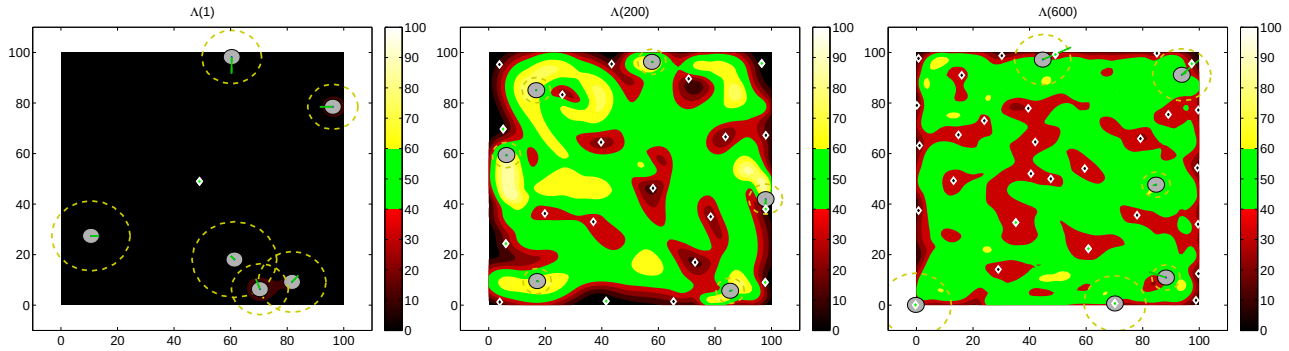
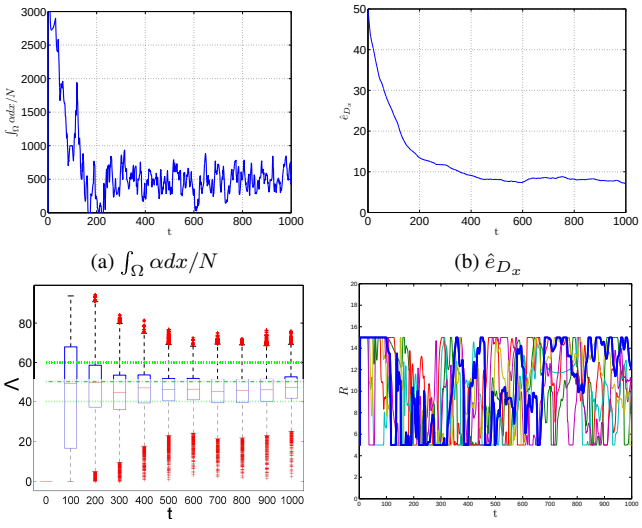


Fig. 5. Evolution of the coverage map of simulation in Section 4.2.1 with $\Lambda^* = 50$ (green color in the plots). The agents are represented by small circles and their coverage domain is represented by a dashed circle. The small rhombi represent the global objectives.



(a) $\int_\Omega \alpha dx/N$ (b) $\hat{e}_{D_x}$

(c) Evolution of Λ, ($\Lambda^* = 50$) (d) Sensor range evolution

Fig. 6. Results of simulations in Section 4.2.1. (a) Evolution of the average action of the agents. (b) Evolution of the average absolute error $\hat{e}_{D_x}$. (c) Boxplot of the distribution of the coverage at several different times. (d) Sensor range evolution of the agents, one of them is depicted in thick line.

of variable range, the value of R in the plots denotes its maximum limit. The integrated average error ($\overline{e}_{D_x}$), average power consumption ($\bar{\alpha}$), and average path length (PL) are represented for different values of $R \in [10, 50]$. In these plots, each line corresponds to a value of $k_i \in [1, 9]$. In general, as resources increase (because agents move faster by means of a higher k_i, or because they cover further points by means of a higher R), the error $\overline{e}_{D_x}$ decreases. In the case of constant range, higher R reduces the error until a minimum is reached and if the range is increased, error also increases since the agents become less flexible to cover small areas or narrow corridors. In the case of the variable range coverage, as the agents adapt their range to the conditions of the environment, the higher maximum radius never decreases the coverage performance. The behaviour of the average power consumption $\overline{\alpha}$ is inverse to the error, so more power consumption leads to lower error and vice versa. Finally, and as expected, the path length increases with the maximum speed (k_i).

4.2.3. Simulation with high decay and flat action

Here we introduce a simulation with a flat coverage action ($r_m = 0.8$) and the rest of parameters the same as in simulation of Section 4.2.2 in order to compare the behaviour of flat and sharp actions. Results are shown in Fig. 8. In this case, the differences between the strategy with constant range and the one with variable range are also evident. Variable range achieves less error with lower coverage power consumption. The path length of the constant range strategy reduces as range increases since the agents can cover a bigger area with less motion.

4.2.4. Simulation with slow decay and flat action

In this example we choose the time constant of the domain ($\tau = 1000$) to have a slow decay of the coverage information. The results are depicted in Fig. 9. It can be seen that the error throughout the simulation span is less variable. Given that the coverage decay is slow, the coverage level loss of points outside the agents' coverage range is also lower, and slower agents (or with less coverage range) achieve similar error to others with higher capacity. If we compare the error with constant and variable range we see an error reduction with the variable range algorithm. The average power consumption is also similar or lower.

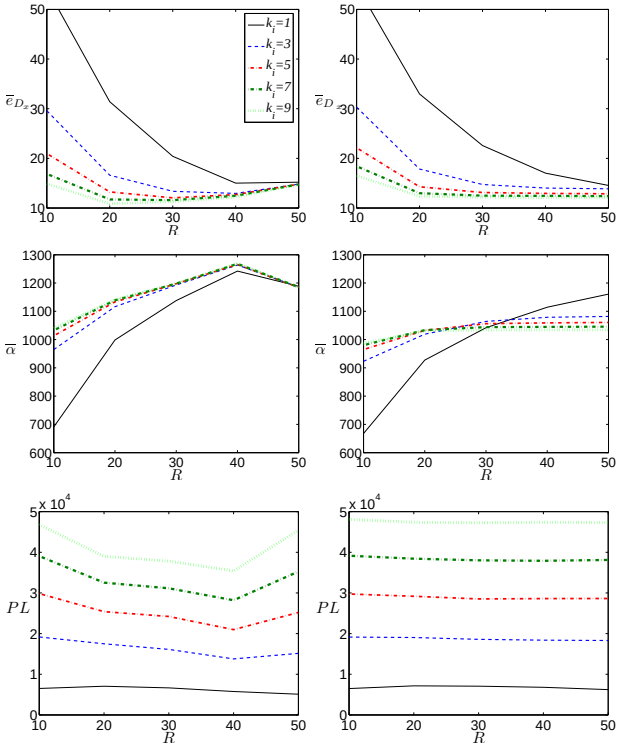


Fig. 7. Aggregated results of simulations of Section 4.2.2 with constant range (left column) and variable range (right column). All plots share the same legend. Simulation parameters: $\tau = 200$, $r_m = 0.1$, $R \in [10, 50]$, and $k_i \in [1, 9]$. From first to third row: integrated average error $\overline{e}_{D_x}$, average power consumption ($\bar{\alpha}$), and average path length (PL).

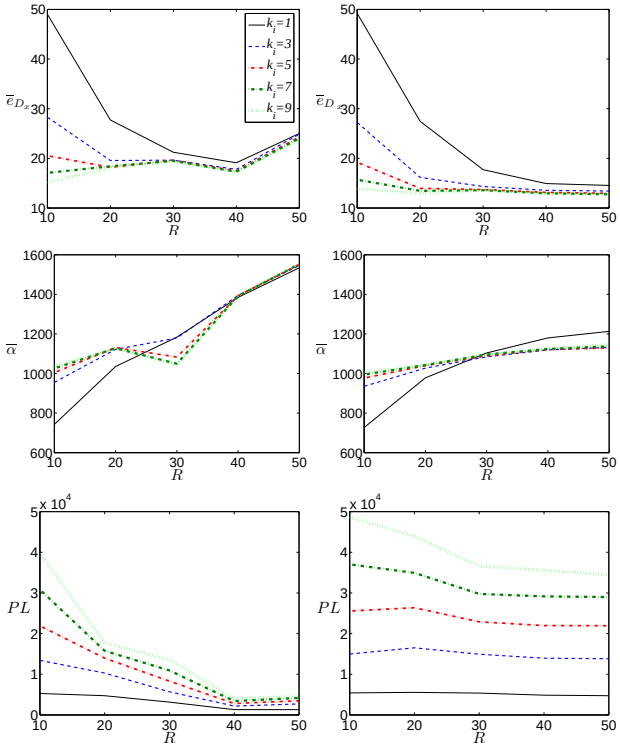


Fig. 8. Aggregated results of simulations of Section 4.2.3 with constant range (left column) and variable range (right column). All plots share the same legend. Simulation parameters: $\tau = 200$, $r_m = 0.8$, $R \in [10, 50]$, and $k_i \in [1, 9]$. From first to third row: integrated average error $\overline{e}_{D_x}$, average power consumption ($\bar{\alpha}$), and average path length (PL).

4.2.5. Different desired coverage levels and priorities

The following examples demonstrate the correct performance and versatility of our proposal in more demanding environments and different setups. In the first experiment (Fig. 10) the area to cover D_x is defined by a map and $D_p = \mathbb{R}^2$. There are areas of the map with different levels of desired coverage $\Lambda^* = \{37.5, 75\}$ to be performed by $N = 4$ agents. The parameters of the problem are defined as $k_i = 10$, $\beta = 1/4$, $R = [3, 15]$, $K = [0, 5000]$, $\Phi = 1$, $\tau = 800$, and $g = 1$. In the second experiment, presented in Fig. 11, the area has to be covered with $\Lambda^* = 50$ by $N = 8$ agents. However, there are 17 zones with different values of coverage priority Φ. Parameters of the problem are $k_i = 3$, $\beta = 1/4$, $R = [5, 15]$, $K = [0, 3000]$, $\tau = 400$, and $g = 1/2$. The results show that given that the set of agents do not have enough capacity for completely covering the environment, the high-priority areas will have better coverage. In the third experiment, shown in Fig. 12, six isolated areas are required to be covered with different desired coverage levels ($\Lambda^* = \{12, 39, 50\}$) and also different priority levels ($\Phi = \{0.12, 1\}$). The problem parameters are set as follows: $N = 6$ agents, $k_i = 5$, $\beta = 1/4$, $R = [3, 10]$, $K = [0, 500]$, $\tau = 600$, and $g = 2/5$.

Different simulations with variable coverage power, with variable range, and with different desired coverage levels and priorities are presented in the **video attachment**, showing the coverage evolution together with the evolution of the agents' power and range.

4.3. Discussion on control tuning

The addressed problem of persistent coverage developed by a team of agents is complex and involves a great variety of parameters. However, many of them are related with the intrinsic features of the environment and setup and therefore they are imposed and not subject to tuning. In particular, A and B represent the dynamics of the environment, k_i is the maximum agents' velocity, while the actuator is modelled with parameters α_i, which depends on σ_i, C, q, and R_i. The requirement of the problem is defined with Λ^* and Φ.

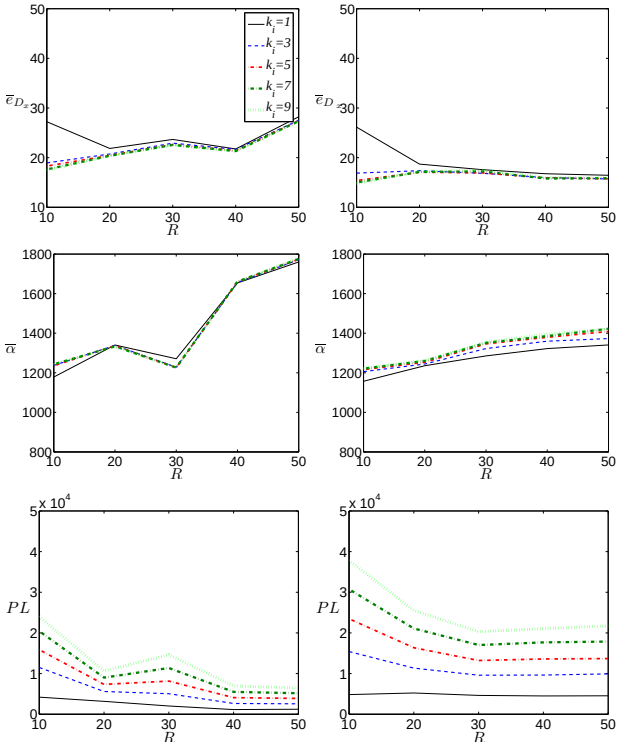


Fig. 9. Aggregated results of simulations of Section 4.2.4 with constant range (left column) and variable range (right column). All plots share the same legend. Simulation parameters: $\tau = 1000$, $r_m = 0.8$, $R \in [10, 50]$, and $k_i \in [1, 9]$. From first to third row: integrated average error $\overline{e}_{D_x}$, average power consumption ($\bar{\alpha}$), and average path length (PL).

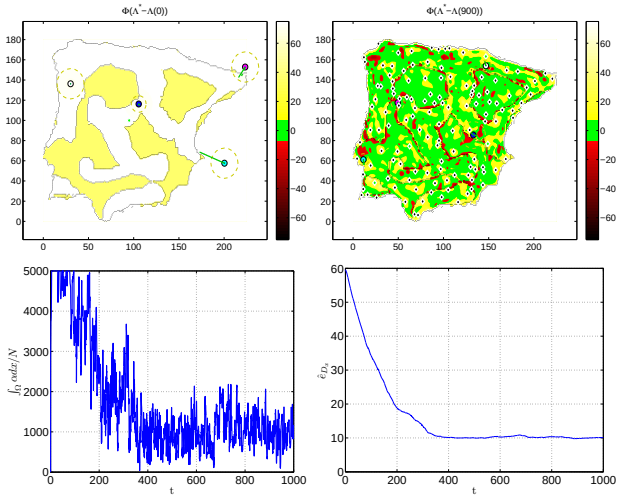


Fig. 10. Results of first experiment in Section 4.2.5. Top-left: Map to perform coverage. The marked zones denote the two different levels of desired coverage. Top-right: Coverage error in $t = 900$. Bottom-left: Evolution of the average action of the agents. Bottom-right: Evolution of the average absolute error.

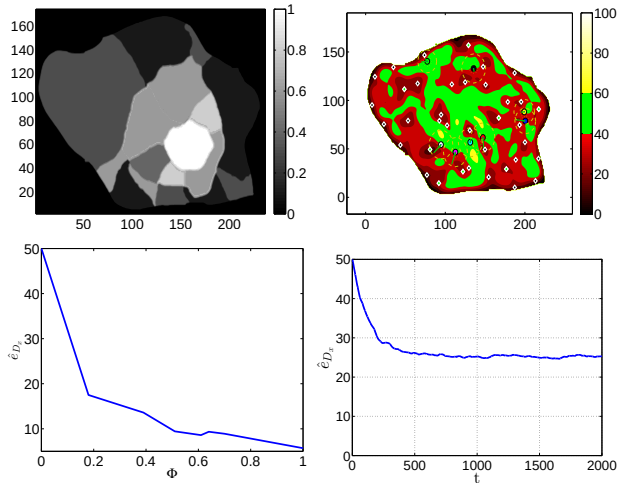


Fig. 11. Results of second experiment in Section 4.2.5. Top-left: Map to perform coverage with zones of different coverage priority. Top-right: Coverage error in $t = 900$. Bottom-left: Final average absolute error with respect priority level. It can be seen that lower error is obtained with higher priority levels. Bottom-right: Evolution of the average absolute error.

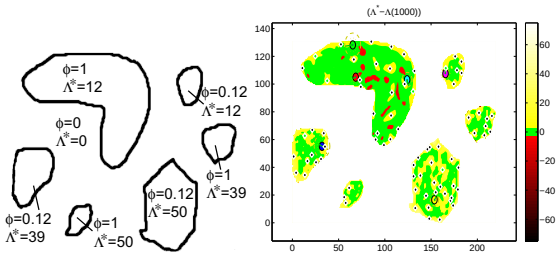


Fig. 12. Results of third experiment in Section 4.2.5. left: Map to perform coverage with priority values and desired coverage levels. Right: Coverage error in $t = 1000$.

Finally, the parameters that need to be tuned in the control algorithm are k_i^R, k_i^G, and β.

Parameter β allows modifying the influence of global and local coverage control law components. In general, hybrid strategies with global and local objectives achieve more efficient coverage, and a good compromise between global and local components is $1/2 < \beta < 1/3$. Regarding k_i^G, we propose to choose values close to 1 until the distance from the agent i to the target is almost the coverage radius R_i, and then decrease its value (16). This allows the agents to reach the targets quickly, and then slow down when the target is being accomplished. Finally, high values of k_i^R makes fast the actuator range variation, allowing maximum reduction of energy consumption. On the other hand, low values make slow the actuator response reducing the energy efficiency. However, note that very high

values of k_i^R may result in a noisy behaviour of the range evolution, which could not be afforded by the dynamics of a real actuator.

V. CONCLUSION

In this paper, we have proposed a solution to the problem of persistent area coverage with variable coverage power and variable range. We have presented a formulation and a new control algorithm based on that formulation. Finally, we evaluate the performance of our control algorithm and we compare it against the performance of the coverage with a constant power coverage actuator. The results show that the variable coverage controller is able to adapt its action to different references achieving lower error and less power consumption. Furthermore, the possibility of range adaptation allows performance increase when dealing with uniform sensing functions.

REFERENCES

1. N. Agmon, N. Hazon, and G. A. Kaminka. Constructing spanning trees for efficient multi-robot coverage. In *IEEE Int. Conf. on Robotics and Automation*, pages 1698 –1703, 2006.
2. E. M. Arkin, S. P. Fekete, and J. S. B. Mitchell. Approximation algorithms for lawn mowing and milling. *Computational Geometry: Theory and Applications*, 17(1-2):25–50, 2000.
3. H.-L. Choi and J. P. How. Continuous trajectory planning of mobile sensors for informative forecasting. *Automatica*, 46(8):1266–1275, 2010.
4. S. Davari, M. H. F. Zarandi, A. Hemmati, and I. B. Turksen. The variable radius covering problem with fuzzy travel times. In *IEEE Int. Conf. on Fuzzy Systems*, pages 1–6, 2010.
5. H. Flanders. Differentiation under the integral sign. *The Amer. Mathematical Monthly*, 80(6):615–627, 1973.
6. C. Franco, G. Lopez-Nicolás, C. Sagüés, and S. Llorente. Persistent coverage control with variable coverage action in multi-robot environment. In *52nd IEEE Conf. on Decision and Control*, pages 6055–6060, 2013.
7. C. Franco, G. López-Nicolás, D. Stipanović, and C. Sagüés. Anisotropic vision-based coverage control for mobile robots. In *IROS Workshop on Visual Control of Mobile Robots (ViCoMoR)*, pages 31–36, 2012.
8. C. Franco, D. Paesa, G. Lopez-Nicolás, C. Sagüés, and S. Llorente. Hierarchical strategy for dynamic coverage. In *IEEE/RSJ Int. Conf. on Intell. Robots and Systems*, pages 5341–5346, 2012.
9. E. Galceran and M. Carreras. Efficient seabed coverage path planning for ASVs and AUVs. In *IEEE/RSJ Int. Conf. on Intell. Robots and Systems*, pages 88–93, 2012.
10. S. K. Gan and S. Sukkarieh. Multi-UAV target search using explicit decentralized gradient-based negotiation. In *IEEE Int. Conf. on Robotics and Automation*, pages 751–756, 2011.
11. J. Gomez and A. T. Campbell. A case for variable-range transmission power control in wireless multihop networks. In *Conf. of the IEEE Computer and Comm. Societies*, volume 2, pages 1425–1436, 2004.
12. I. I. Hussein. Kalman filtering with optimal sensor motion planning. In *Amer. Control Conf.*, pages 3548–3553, 2008.
13. I. I. Hussein and D. M. Stipanović. Effective coverage control for mobile sensor networks with guaranteed collision avoidance. *IEEE Trans. on Control Systems Technology*, 15(4):642–657, 2007.
14. Y.-L. Jian, F.-L. Lian, and H.-T. Lee. Deployment of a team of biomimetic searching agents based on limited communication quantity. *Asian J. of Control*, 10(4):439–448, 2008.
15. J. W. Kang, S. J. Kim, M. J. Chung, H. Myung, J. H. Park, and S. W. Bang. Path planning for complete and efficient coverage operation of mobile robots. In *Int. Conf. on Mechatronics and Automation*, pages 2126 –2131, 2007.
16. J. Li, Z.-H. Guan, and G. Chen. Multi-consensus of nonlinearly networked multi-agent systems. *Asian J. of Control*, 2013.
17. J. M. Luna, R. Fierro, C. T. Abdallah, and J. Wood. An adaptive coverage control for deployment of nonholonomic mobile sensor networks over time-varying sensory functions. *Asian J. of Control*, 15(4):988–1000, 2013.
18. F. Pasqualetti, J. W. Durham, and F. Bullo. Cooperative patrolling via weighted tours: Performance analysis and distributed algorithms. *IEEE Trans. on Robotics*, 28(5):1181–1188, 2012.
19. F. Pasqualetti, A. Franchi, and F. Bullo. On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms. *IEEE Trans. on Robotics*, 28(3):592–606, 2012.
20. D. Portugal and R. P. Rocha. Distributed multi-robot patrol: A scalable and fault-tolerant

framework. *Robotics and Autonomous Systems*, 61(12):1572–1587, 2013.
21. A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser. Efficient informative sensing using multiple robots. *J. of Artif. Intell. Research*, 34(1):707–755, 2009.
22. S. L. Smith, M. Schwager, and D. Rus. Persistent robotic tasks: Monitoring and sweeping in changing environments. *IEEE Trans. on Robotics*, 28(2):410–426, 2012.
23. D. E. Soltero, M. Schwager, and D. Rus. Generating informative paths for persistent sensing in unknown environments. In *IEEE/RSJ Int. Conf. on Intell. Robots and Systems*, pages 2172–2179, 2012.
24. C. Song, G. Feng, Y. Fan, and Y. Wang. Decentralized adaptive awareness coverage control for multi-agent networks. *Automatica*, 47(12):2749 – 2756, 2011.
25. C. Song, G. Feng, and Y. Wang. Decentralized dynamic coverage control for mobile sensor networks in a non-convex environment. *Asian J. of Control*, 15(2):512–520, 2013.
26. D. M. Stipanović, C. Valicka, C. Tomlin, and T. R. Bewley. Safe and reliable coverage control. *Numerical algebra control and optimization*, 3(1):31–48, 2013.
27. J. Wang and S. Medidi. Energy efficient coverage with variable sensing radii in wireless sensor networks. In *Third IEEE Int. Conf. on Wireless and Mobile Computing, Networking and Comm.*, page 61, 2007.
28. Y. Wang and I. I. Hussein. Awareness coverage control over large-scale domains with intermittent communications. *IEEE Trans. Autom. Control*, 55(8):1850–1859, 2010.
29. Y. Wang and I. I. Hussein. Bayesian-based domain search using multiple autonomous vehicles with intermittent information sharing. *Asian J. of Control*, 16(1):20–29, 2014.