Underwater coverage with a mobile robot of limited control authority*

E. Aranda-Escolástico[1], J. Cortés[2], M. Guinaldo[1], S. Dormido[1]

***Abstract*— This work considers the coverage of underwater areas with a mobile robot with constrained control and communication capabilities. While underwater, the robot can control its depth but it is subject to flow in the other directions. While on the surface, it can move (essentially) freely. The aim of the work is the coverage of the areas with the minimum waste of resources. For that, we propose a two-part algorithm, where one part is a genetic algorithm and the other part is an algorithm based on Newton's method. Numerical simulations are provided to illustrate the efficiency of the algorithm.**

I. Introduction

This paper studies the underwater coverage surfaces using an autonomous vehicle with very limited possibilities of control and communication. Specifically, we consider that the vehicle can only communicate with the base station while at the surface of the water. In addition, we assume that the vehicle is only able to control its depth, while its movement in other directions is determined by the tides and other effects. In spite of these limitations, this type of submarines can be very useful for oceanography, aquaculture, hydrographic survey, etc, where vehicles with more capabilities but much more expensive might have a reduced success compared with a swarm of smaller submarines.

Literature review: The study of the coverage problem has been carried out from multiple points of view. Sensor coverage algorithms have received a great attention in recent years. In [1], a distributed control algorithm is proposed for finding a locally optimal sensing configuration for groups of vehicles. There have been several extensions to this formulation of coverage control (see, e.g., [2], [3]). In [4], authors incorporated heterogeneous robots, and extended the algorithm to handle non-convex environments. Other extensions to non-convex environments were proposed in [5] and [6]. Similar frameworks have also been formulated for stochastic settings [7]. A dynamic coverage control algorithm with limited transmission of information is proposed in [8]. The works [9], [10] study the dynamic coverage control under different conditions. The nonuniform coverage problem is addressed for example in [11]. In the context of underwater systems, methods have been proposed for mine detection [12], 3D coverage [13], and surveillance [14]. Specially, [15] considers the coverage under the effect of sea current disturbances. An algorithm to take into account environmental factors during the coverage is presented in [16]. The coverage of 3D structures is studied in [17].

Statement of contributions: As opposed to the previous works, the control actions are very restricted in our work and the movement of the vehicles is determined by the current flow. Therefore, the coverage problem might be significantly different. The unique possible action in the coverage of the surface is the choice of the position and time of immersion. In order to make this decision, we use an estimation of the current flow to optimize an estimated area that the vehicle should cover. Since the difference between the estimated and the real current flow produces an accumulative error in our knowledge of the position of the vehicle, the time of immersion plays an important role in order to determine when returning to the water surface to obtain new information is better than continuing the coverage. To solve the optimization problem, we propose a two-part algorithm composed by a genetic algorithm, which provides a "good" region of immersion, and a Newton's method, which provides the optimal position of immersion inside the region.

Notation: We let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{N}$ denote respectively the set of real, nonnegative real, and natural numbers. The n-dimensional real space is defined by $\mathbb{R}^n$. We refer to the Euclidean norm of vector $v \in \mathbb{R}^n$ as $\|v\| = \sqrt{v^T v}$. For $A \in \mathbb{R}^{n\times m}$, we let $A^\top$ denote its transpose matrix. We denote the identity matrix $\mathbb{I} \in \mathbb{R}^{n\times n}$ by $\mathbb{I}_n$. A function $\mu : [0, a) \to [0,\infty)$ is class $\mathcal{K}$ if it is continuous, zero at zero and strictly increasing and it is class $\mathcal{K}_\infty$ if, in addition, $\lim_{r\to\infty} \mu(r) = \infty$. A function $\sigma : [0, a) \times [0,\infty) \to [0,\infty)$ is class $\mathcal{KL}$ if, for each fixed s, the mapping $\sigma(r, s)$ is class $\mathcal{K}$ w.r.t. r and, for each fixed r, the mapping $\sigma(r, s)$ is decreasing w.r.t. s and $\lim_{s\to\infty} \sigma(r, s) = 0$. A dynamical system $\dot{x} = f(x, u)$ is input-to-state stable (ISS) [18, Definition 4.7] if there exist a class $\mathcal{KL}$ function β and a class $\mathcal{K}$ function γ such that for any initial state $x(t_0)$ and any bounded input $u(t)$, the solution $x(t)$ exists for all $t \geq t_0$ and satisfies $\|x(t)\| \leq \beta(\|x(t_0)\|, t-t_0)+\gamma\left(\sup_{t_0\leq\tau\leq t} \|u(\tau)\|\right)$. The existence of an ISS-Lyapunov function is sufficient to ensure that the system is ISS stable. Formally, let $V : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function such that

$$\alpha_1(\|x\|) \leq V(t, x) \leq \alpha_2(\|x\|) \tag{1a}$$

$$\dot{V} \leq -W_3(x),\ \|x\| \geq \mu(\|u\|) > 0 \tag{1b}$$

for all $(t, x, u) \in [0,\infty) \times \mathbb{R}^n \times \mathbb{R}^m$, where α_1, α_2 are class $\mathcal{K}_\infty$ functions, μ is a class $\mathcal{K}$ function, and W_3 is a

*This work was supported in part by the Spanish Ministry of Economy and Competitiveness (MINECO) under the Projects DPI2012-31303, DPI2014-55932-C2-1-R and DPI2017-84259-C2-2-R. Part of this work was carried out in the UC - San Diego thanks to the grant EEBB-I-17-12464.

[1]Departamento de Informática y Automática, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain earandae@bec.uned.es, mguinaldo@dia.uned.es, sdormido@dia.uned.es

[2]Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA, 92093, USA cortes@ucsd.edu

continuous positive definite function on $\mathbb{R}^n$. Then, the system $\dot{x} = f(x, u)$ is ISS stable with $\gamma = \alpha_1^{-1} \circ \alpha_2 \circ \mu$.

II. Problem statement

We consider the problem of covering an underwater region $\mathcal{D} \in \mathbb{R}^2$, such that all points to be measured are located at the same depth d, i.e., all point $\tilde{q} \in \mathcal{D}$ can be denoted as $\tilde{q} = \begin{pmatrix} \tilde{x} & \tilde{y} & d \end{pmatrix}^\top$. For the fulfillment of this objective, we use a set of vehicles capable of moving in the surface as well as drifting underwater. An example of such vehicle platform is the Data Diver developed by Apium [19]. To achieve this objective, the data diver performs a sequence of immersions in the water. In each immersion, the data diver will cover some part of $\mathcal{D}$ until the whole region results covered. The question is then where to immerse and for how long so that the covered region is maximized in order to efficiently perform the task.

First of all, we describe the behavior and sensing capabilities of the data diver in order to present the intricacies of the problem. The design of the data diver implies that its movement is composed by two parts. On one hand, it can move freely on the surface of the water. On the other hand, when it is submerged, only the depth can be controlled, while the flow of the water determines its movements in the others directions. The communications are carried out through GPS and short range radios. For that reason, the localization and communication on the surface are considerably accurate but we lose these capabilities underwater. In addition, we have to take into account important sources of error. To decide the immersion position, we work with a model of the flow which is an approximation of the actual one. We can obtain the flow estimation using the data collected during the dives (see, e.g., [20], [21]). Besides, several disturbances can occur underwater and the lack of position measurements hinders their rejection. For these reasons, when we decide how much time the data diver is submerged, we have to assume that the actual coverage might be considerably different to our estimation. Consequently, if we want to measure $\mathcal{D}$, we have to move the data diver on the surface such that, when it is submerged, the current flow sweeps along it enabling the coverage of some part of $\mathcal{D}$. At a certain moment, the data diver returns to the surface to decide its new immersion. Hence, our control the coverage performed in each immersion depends exclusively on the position where the data diver is submerged and on the time that the data diver is submerged. Therefore, the possible coverage is going to be maximized as a function of these two variables.

The process of immersion and emersion of the data diver is described in Figure 1. The position of the data diver is determined by its Cartesian position and its orientation with respect to the x-axis, which form the vector $q(t) \in \mathbb{R}^4$. The data diver emerges at the instant t_k^{em}, $k \in \mathbb{N}$, being $t_0^{\text{em}} = 0$ the initial time and it submerges at the instant t_k^{im}, $k \in \mathbb{N}$. We decide where the data diver submerges ($q(t_k^{\text{im}})$) and how much time is submerged ($T_{k,\text{sub}} = t_{k+1}^{\text{em}} - t_k^{\text{im}}$) to maximize the covered area of $\mathcal{D}$.

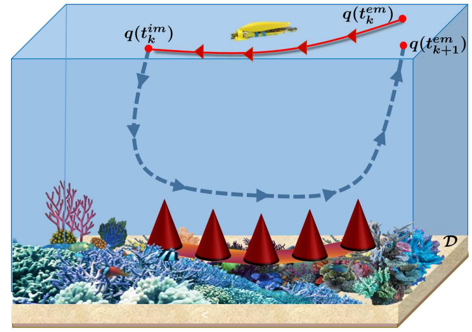

Fig. 1. Diagram of path followed by data diver during each immersion.

Next, we describe the data diver dynamics, its communication and sensing capabilities and the current flow model.

A. Data diver modeling

The data diver is a hybrid system, whose dynamics change depending on its depth. If the data diver is on the surface of the water, we assume that it can move like a non-holonomic mobile vehicle. In contrast, the currents determine its underwater dynamics. Let us consider the following assumptions:

Assumption 2.1: The change between dynamics is instantaneous.

Assumption 2.2: The position of the data diver is exactly known on the water surface, while underwater only depth measurements are available.

According to Assumption 2.1, the dynamics of the system can be written as follows

$$\dot{q}(t) = f(q(t), t) = \begin{cases} g_{\text{sur}}(q(t))u_{\text{sur}}(t) & \text{(2a)} \\ \quad \text{for } t \in [t_k^{\text{em}}, t_k^{\text{im}}) & \\ f_{\text{sub}}(q(t), t) + g_{\text{sub}} u_{\text{sub}}(t) & \text{(2b)} \\ \quad \text{for } t \in [t_k^{\text{im}}, t_{k+1}^{\text{em}}) & \end{cases}$$

with $t \in \mathbb{R}_{\geq 0}$ and initial conditions $q_0 = q(0)$, and where $q(t) = \begin{pmatrix} x(t) & y(t) & z(t) & \theta(t) \end{pmatrix}^\top \in \mathbb{R}^4$ are the generalized coordinates of the system. The Cartesian position of the data diver is $\begin{pmatrix} x(t) & y(t) & z(t) \end{pmatrix}$ and $\theta(t)$ is the orientation with respect to the x axis. $u_{\text{sur}}(t) \in \mathbb{R}^2$ is the control input on the surface and $u_{\text{sub}}(t) \in \mathbb{R}$ is the underwater control input. Note that we can define a map $\Phi_t : \mathbb{R}^4 \to \mathbb{R}^4$ such that

$$q(t) = \Phi_f^t(q_0), \tag{3}$$

which determines the position of the data diver after a time t.

The dynamics which describes the movement on the surface of the water is

$$g_{\text{sur}}(q(t)) = \begin{bmatrix} \cos\theta(t) & \sin\theta(t) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^\top. \tag{4}$$

In (4), we are ignoring the effect of the water in the dynamics while the vehicle is on the surface of the water. This is a reasonable simplification, since the nonholonomic model above is globally controllable.

The underwater movement is determined by the flow of the water in the xy-plane, while the depth z can be controlled through the measurement of the pressure.

Then, $f_{\text{sub}}(q(t),t) = \begin{bmatrix}\nu^\top(q(t),t) & 0\end{bmatrix}^\top$ and $g_{\text{sub}} = \begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix}^\top$, where the flow of the water which sweeps along the data diver is represented by the vector $\nu(q(t),t) = \begin{pmatrix}\nu_x(q(t),t) & \nu_y(q(t),t) & \nu_z(q(t),t)\end{pmatrix}^\top \in \mathbb{R}^3$.

The instantaneous coverage $s(\tilde{q}, q(t))$ represents the coverage of the point $\tilde{q}$ when the data diver is in $q(t)$. A possible parametrization of this instantaneous coverage is

$$s(\tilde{q}, q(t)) = \begin{cases} \left(\delta^2(t) - r^2\right)^4 / r^8 \\ \quad \text{if } \|z(t) - d\| \leq h \text{ and } \delta(t) \leq r, \\ 0 \quad \text{otherwise.} \end{cases} \tag{5}$$

where $\delta(t) = \|(x(t), y(t)) - (\tilde{x}, \tilde{y})\|$, and h and r are the height and the radius, respectively, which define the maximum cone that can be captured by the cam of the data diver. Note that (5) is only a possible model of the instantaneous coverage. However, it can be described in different ways depending on the sensor characteristics [22].

B. Flow and sensing estimation

To solve the optimization problem, we need some knowledge of the current flow to estimate the underwater position. Let us assume that the estimation of the trajectory is $\hat{q}(t) \in \mathbb{R}^4$ such that $\hat{q}(t) = \begin{pmatrix}\hat{x}(t) & \hat{y}(t) & \hat{z}(t) & \hat{\theta}(t)\end{pmatrix}^\top$ and that we have a model of the current flow $\hat{\nu}(\hat{q}(t),t) = \begin{pmatrix}\hat{\nu}_x(\hat{q}(t),t) & \hat{\nu}_y(\hat{q}(t),t) & \hat{\nu}_z(\hat{q}(t),t)\end{pmatrix}^\top \in \mathbb{R}^3$. Hence, its estimated dynamics is

$$\dot{\hat{q}}(t) = \hat{f}(\hat{q}(t),t) = \begin{cases} g_{\text{sur}}(\hat{q}(t))u_{\text{sur}}(t) & \text{(6a)} \\ \quad \text{for } t \in [t_k^{\text{em}}, t_k^{\text{im}}) \\ \hat{f}_{\text{sub}}(\hat{q}(t),t) + \hat{g}_{\text{sub}}\hat{u}_{\text{sub}}(t) & \text{(6b)} \\ \quad \text{for } t \in [t_k^{\text{im}}, t_{k+1}^{\text{em}}) \end{cases}$$

with initial conditions $\hat{q}_0 = \hat{q}(0)$ and where $\hat{f}_{\text{sub}}(\hat{q}(t),t) = \begin{bmatrix}\hat{\nu}^\top(\hat{q}(t),t) & 0\end{bmatrix}^\top$ and $\hat{g}_{\text{sub}} = \begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix}^\top$.

Assumption 2.2 implies that the position is known on the surface, i.e., $\hat{q}(t) = q(t)$, while underwater only $\hat{\theta}(t) = \theta(t)$. Note that $\hat{z}(t) \neq z(t)$ because even when the depth can be measured, the estimation is carried out before the immersion. Analogous to (3), $\hat{q}(t) = \hat{\Phi}^t_{\hat{f}}(\hat{q}_0)$. Since we can consider that the initial conditions of the data diver are always on the surface, we can assume $\hat{q}_0 = q_0$. With this information, we can define the following error vector

$$e(t) = q(t) - \hat{q}(t) = \begin{cases} 0_{4\times1} \text{ for } t \in [t_k^{\text{em}}, t_k^{\text{im}}) \\ \begin{pmatrix}e_x(t) & e_y(t) & e_z(t) & 0\end{pmatrix}^\top \\ \quad \text{for } t \in [t_k^{\text{im}}, t_{k+1}^{\text{em}}) \end{cases}.$$

This knowledge of the flow enables the coverage estimation in the next immersion. First, let us assume the following:

Assumption 2.3: There exists bounds on the estimation error such that $\|(e_x(t), e_y(t))\| \leq \bar{e}_{1,k}(t - t_k^{\text{im}}, q(t_k^{\text{im}}))$ and $\|e_z(t)\| \leq \bar{e}_{2,k}(t - t_k^{\text{im}}, q(t_k^{\text{im}}))$, where $\bar{e}_{1,k}$ and $\bar{e}_{2,k}$ are known functions which are zero if $t = t_k^{\text{im}}$ and are strictly increasing with $t - t_k^{\text{im}}$.

Assumption 2.3 implies that we have an expected error of our estimation. This simplification enables the abstraction of the problem to carry out the later development. The assumption might be a strong constraint if the flow has a big dependance with the time. In contrast, if the time dependance is small, we can use the obtained data in previous immersions to update the bound. In addition, if the flow does not suffer large changes in the Cartesian coordinates, we can consider a constant bound. With this information, replacing $(x(t), y(t))$ by $(\hat{x}(t), \hat{y}(t)) + (e_x(t), e_y(t))$ and $z(t)$ by $\hat{z}(t) + e_z(t)$ in (5) and using the properties of the norm, we obtain an estimation of the instantaneous coverage

$$\hat{s}(\tilde{q}, \hat{q}(t)) = \begin{cases} \left(\left(\hat{\delta}(t) + \bar{e}_{1,k}(t - t_k^{\text{im}}, q(t_k^{\text{im}}))\right)^2 - r^2\right)^4 / r^8 \\ \quad \text{if } \|\hat{z}(t) - d\| \leq h - \bar{e}_{2,k}(t - t_k^{\text{im}}, q(t_k^{\text{im}})) \\ \quad \text{and } \hat{\delta}(t) \leq r - \bar{e}_{1,k}(t - t_k^{\text{im}}, q(t_k^{\text{im}})) \\ 0 \text{ otherwise,} \end{cases} \tag{7}$$

where $\hat{\delta}(t) = \|(\hat{x}(t), \hat{y}(t)) - (\tilde{x}, \tilde{y})\|$, and which satisfies $\hat{s}(\tilde{q}, \hat{q}(t)) \leq s(\tilde{q}, q(t))$ since $\bar{e}_{1,k} \geq 0$ and $\bar{e}_{2,k} \geq 0$. So, (5) provides the instantaneous coverage of point $\tilde{q}$ and (7) is an estimation of this coverage made before the immersion.

III. Algorithm design

In this section, we design the coverage algorithm. To perform the desired coverage, it is necessary to define the control law $u_{\text{sur}}(t)$, which brings the data diver to the point $q(t_k^{\text{im}})$, and the control law $u_{\text{sub}}(t)$, which brings the data diver to the desired depth $d^\star$, $0 < d^\star < d$ sufficiently close to $\mathcal{D}$ to do the measurement. First, these control laws are described. Then, the coverage problem is studied.

A. Surface control

The inputs vector $u_{\text{sur}}(t)$ is composed by the driving and steering velocities such that $u_{\text{sur}}(t) = \begin{pmatrix}u_{sur,1}(t) & u_{sur,2}(t)\end{pmatrix}^\top$. Through a dynamic linerization algorithm the following controller is obtained [23]

$$\begin{aligned} \dot{\xi}(t) &= v_1(t)\cos\theta(t) + v_2(t)\sin\theta(t) \\ u_{sur,1}(t) &= \xi(t) \\ u_{sur,2}(t) &= \frac{v_2(t)\cos\theta(t) - v_1(t)\sin\theta(t)}{\xi(t)}, \end{aligned} \tag{8}$$

being $\xi \in \mathbb{R}$ the state of a one-dimensional dynamic compensator, and $v_1(t) = k_{p1}(x_{\text{im}} - x(t)) - k_{d1}\dot{x}(t)$ and $v_2(t) = k_{p2}(y_{\text{im}} - y(t)) - k_{d1}\dot{y}(t)$, where $q_{\text{im}} = \begin{pmatrix}x_{\text{im}} & y_{\text{im}} & 0 & 0\end{pmatrix}^\top$ is the point where the data diver should do the immersion, i.e., $q(t_k^{\text{im}})$. $k_{p1}, k_{d1}, k_{p2}, k_{d2} \geq 0$ should satisfy the constrains imposed in [23].

B. Underwater control

It is easy to see that the estimated underwater dynamics (6b) converges and asymptotically to $d^\star$ with the control law $\hat{u}_{\text{sub}}(t) = -\hat{\nu}_z(\hat{q}(t),t) + k(d^\star - \hat{z}(t))$, where $k > 0$ is a feedback gain. However, note that the evolution of the actual system (2b) may be different depending on the error $e_{\nu_z}(t) = \nu(q(t),t) - \hat{\nu}(\hat{q}^\star(t),t)$, where $\hat{q}^\star(t) = \begin{pmatrix}\hat{x}(t) & \hat{y}(t) & z(t) & \theta(t)\end{pmatrix}^\top$. Note that the definition of $\hat{q}^\star(t)$

is necessary because we know the exact depth of the data diver at every moment and we can use this information in the control law. Nevertheless, we still need an estimation of the depth to compute the estimated trajectory, since this estimation will be done before the knowledge of the actual estimation. In contrast, we have to use always the estimations on the xy-plane because its position on that plane is unknown while the data diver is underwater. With this information, let us consider the following state feedback control law

$$u_{\text{sub}}(t) = -\hat{\nu}_z(\hat{q}^\star(t), t) + k(d^\star - z(t)). \tag{9}$$

Asymptotic stability cannot be proved for the actual underwater dynamics since we have an unknown error $e_{\nu_z}(t)$. In spite of that, ISS stability respect to $e_{\nu_z}(t)$ is guaranteed.

Proposition 3.1: Consider the underwater subsystem (2b) with control law (9). Then, (2b) is ISS respect to e_{ν_z}.

The importance of Proposition 3.1 is that the more we submerge the data diver and we obtain new information of the current flow, the more accurate estimation we have and, consequently, we can guarantee that the closer to $d^\star$ the data diver will be in the next immersions.

C. Coverage optimization

In this section, we describe the strategy to optimize the coverage of the desired surface. To obtain the effective coverage of $\tilde{q}$ we should make the following assumption:

Assumption 3.2: Once the data diver has emerged, it is possible to reconstruct the followed trajectory.

Assumption 3.2 is acceptable in view of works like [20], where the flow field is successfully reconstructed. Hence, due to Assumption 3.2, we can compute the effective coverage of $\tilde{q}$ performed by the data diver, from the beginning until the instant of the next immersion t_k^{im}, i.e.,

$$S_k(\tilde{q}) = \int_0^{t_k^{\text{im}}} s(\tilde{q}, q(t))dt. \tag{10}$$

Analogously, the estimation of the effective coverage of $\tilde{q}$ after the immersion is

$$\hat{S}_k(\tilde{q}, T_{k,\text{sub}}, q(t_k^{\text{im}})) = \int_{t_k^{\text{im}}}^{t_k^{\text{im}}+T_{k,\text{sub}}} \hat{s}(\tilde{q}, \hat{\Phi}_{\hat{f}}^t(q(t_k^{\text{im}}))dt + S_k(\tilde{q}), \tag{11}$$

where $T_{k,\text{sub}} = t_{k+1}^{\text{em}} - t_k^{\text{im}}$ is the time that the data diver is submerged in the immersion k and $q(t_k^{\text{im}})$ is the position of that immersion. Note that (10) depends only on $\tilde{q}$ because it is simply the coverage of $\tilde{q}$ obtained until now, i.e., it is the coverage initial condition for the next immersion. In contrast, the future coverage (11) depends on the future position of immersion and the time associated to this immersion. Let $C(\tilde{q})$ be the desired effective coverage of $\tilde{q}$ such that once the coverage of $\tilde{q}$ has reached $C(\tilde{q})$, we consider that $\tilde{q}$ has been perfectly covered. Then, it is possible to define a penalty function p, positive definite, which penalizes both the lack of coverage of $\tilde{q}$ if it is less than $C(\tilde{q})$ and the excess of coverage it is larger than $C(\tilde{q})$. An example can be

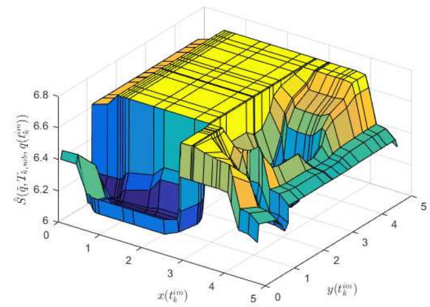

Fig. 2. Coverage estimation depending on the position of immersion for a fixed $T_{k,\text{sub}}$ when the data diver is submerged in a simple laminar flow.

$$p(C(\tilde{q}), \hat{S}_k(\tilde{q}, T_{k,\text{sub}}, q(t_k^{\text{im}}))) = -\frac{\log\left(e^{-\lambda \hat{S}_k(\tilde{q}, T_{k,\text{sub}}, q(t_k^{\text{im}}))} + e^{-\lambda C(\tilde{q})}\right)}{\lambda}, \tag{12}$$

which is a smooth function which satisfies that the larger $\lambda > 0$ is, the closer the approximation is to the min function. With these considerations, the total estimated effective coverage of region $\mathcal{D}$ after immersion k is

$$Q(T_{k,\text{sub}}, q(t_k^{\text{im}})) = \int_{\mathcal{D}} p(C(\tilde{q}), \hat{S}_k(\tilde{q}, T_{k,\text{sub}}, q(t_k^{\text{im}})))d\tilde{q}. \tag{13}$$

Note that if $\hat{S}(\tilde{q})$ increases, then Q increases but once $\hat{S}(\tilde{q})$ reaches $C(\tilde{q})$, new coverage of $\tilde{q}$ are penalized due to (12). Finally, the optimization problem to solve is

$$\underset{T_{k,\text{sub}} \in \mathbb{R}_{\geq 0}, q(t_k^{\text{im}}) \in \mathbb{R}^4}{\text{minimize}} \quad -Q(T_{k,\text{sub}}, q(t_k^{\text{im}})). \tag{14}$$

Solving (14) implies the optimization of the estimated coverage which guarantees at least the same level of coverage in the actual immersion.

Remark 3.3: Since (13) is a nondecreasing function of $T_{k,\text{sub}}$, it might be interesting adding constraints to minimize $T_{k,\text{sub}}$ to guarantee that in case of similar coverage, the diver follows the trajectory which covers more area in less time.

The function $-Q$ might be non-convex depending on the flow and the sensing and penalty functions. In addition, the more parts of $\mathcal{D}$ are covered, more local minimums appear (See Figure 2 as an example). For this reason, a two-part algorithm is designed to solve the optimization problem. First, an heuristic search of non-covered regions is carried out with a genetic algorithm [24]. Random time and position of immersion form each chromosome of the population. The crossover and mutation cause the evolution toward non-covered regions. Second, the result of the genetic algorithm is used as the initial condition for a Newton's method to obtain a local optimized value.

The application of the Newton's method needs the computation of the second order partial derivatives of (13). The following propositions provide the conditions to guarantee the existence of these derivatives.

Proposition 3.4: If $\hat{f}_{\text{sub}}$ is a twice differentiable function, then $\partial\Phi^t_{\hat{f}}(q_0)/\partial q_0$ and $\partial^2\Phi^t_{\hat{f}}(q_0)/\partial q_0^2$ exist in the interval $t \in [t_k^{\text{im}}, t_{k+1}^{\text{em}})$.

Proposition 3.5: If $\hat{s}$ and $\hat{f}_{\text{sub}}$ are Riemann integrable functions, p and $\hat{s}$ are twice differentiable functions, and Proposition 3.4 is satisfied, and hence Newton's method can be implemented to find a local optimum with

$$\nabla Q(T, q_0) = \begin{bmatrix} \int\limits_{\mathcal{D}} \frac{\partial p(C(\tilde{q}),\hat{S}(\tilde{q},T,q_0))}{\partial \hat{S}} \frac{\partial \hat{S}(\tilde{q},T,q_0)}{\partial T} d\tilde{q} \\ \int\limits_{\mathcal{D}} \frac{\partial p(C(\tilde{q}),\hat{S}(\tilde{q},T,q_0))}{\partial \hat{S}} \frac{\partial \hat{S}(\tilde{q},T,q_0)}{\partial q_0} d\tilde{q} \end{bmatrix}, \quad (15)$$

$$HQ(T, q_0) = \begin{bmatrix} H_1 & H_2 \\ H_2^\top & H_3 \end{bmatrix}, \quad (16)$$

where (15)-(16) are detailed in (17)-(19).

Once all the elements have been introduced, we propose Algorithm 1 to complete the coverage of the surface $\mathcal{D}$.

IV. Simulation

The algorithm developed in Section 3 has been tested in different simulations. Consider the model described by (2a)-(2b) under control law (8)-(9) and assume the current flow

$$\nu(q(t), t) = \begin{pmatrix} -0.84 \cdot 10^{-3} y(t) - 0.015 \\ -1.37 \cdot 10^{-3} x(t) \\ 0 \end{pmatrix} \quad (20)$$

while the available model to do the estimation is

$$\hat{\nu}(\hat{q}(t), t) = \begin{pmatrix} -10^{-3} \hat{y}(t) - 0.01 \\ -10^{-3} \hat{x}(t) \\ 0 \end{pmatrix}. \quad (21)$$

The surface to be covered is a square region of 5 m and it is located at $d = 12$ m. Hence, the reference depth for the data diver is $d^\star = 11.9$ m. The parameters of the camera sensor (5) are $h = 0.2$ m and $r = 0.5$ m. From (20)-(21), we obtain $\bar{e}^2_{1,k}(t - t_k^{\text{im}}, q(t_k^{\text{im}})) = (\bar{x}_0 |e^{c_1(t-t_k^{\text{im}})} - e^{c_2(t-t_k^{\text{im}})}| + |c_3(1 - e^{c_1(t-t_k^{\text{im}})}) + (1 - e^{c_2(t-t_k^{\text{im}})})|)^2 + \bar{y}_0^2 (e^{c_1(t-t_k^{\text{im}})} - e^{c_2(t-t_k^{\text{im}})})^2$ and $\bar{e}_{2,k}(t - t_k^{\text{im}}, q(t_k^{\text{im}})) = 0$ with $c_1 = 0.0011$, $c_2 = 0.0010$, $c_3 = 1.2455$, and $\bar{x}_0 = 10$ and $\bar{y}_0 = 10$ the maximum expected initial conditions in xy-plane. By simplicity, we can take $\bar{e}_1(t, t_1^{\text{im}}, ..., t_k^{\text{im}}) \leq 0.01(t - t_k^{\text{im}})$ for $t - t_k^{\text{im}} \leq 1300$ s, which is acceptable approximation for the case of study. Figure 3 shows a comparison of the two-step algorithm w.r.t. the Newton's method and the genetic algorithm separately considered and w.r.t. the coverage with random immersions. Naturally, the combination of the Newton's method with the genetic algorithm provides better results than if we consider them independently. In Figure 4, we can observe as the data diver covers the surface following the two-step algorithm.

Algorithm 1: Coverage of an underwater surface.

Step 1 Set $k = 1$, $S(\tilde{q}) = 0$.
Step 2 Execute the genetic algorithm to obtain a first approximation for $T_{k,\text{sub}}$ and $q(t_k^{\text{im}})$.
Step 3 Use the values of Step 2 as initial conditions of the Newton's method to obtain $T_{k,\text{sub}}$ and $q(t_k^{\text{im}})$.
Step 4 Update (8) to reach the desired position $q(t_k^{\text{im}})$.
Step 5 Once the data diver reaches $q(t_k^{\text{im}})$, submerge it to a depth of $d^\star$ [units of length] using (9).
Step 6 Keep the data diver underwater $T_{k,\text{sub}}$ [units of time].
Step 7 Return the data diver to the water surface and rebuild its trajectory. If $C(\tilde{q}) = S_k(\tilde{q})$, $\forall \tilde{q}$, exit. Otherwise update $k = k + 1$ and $S(\tilde{q})$, and go to Step 2.

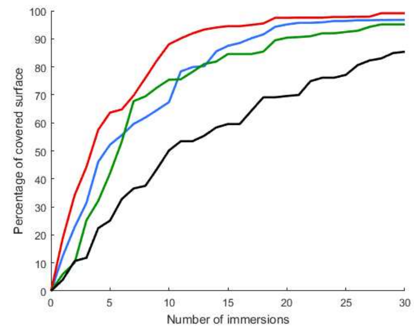

Fig. 3. Comparative of the covered area by the different algorithms. (Red) Newton's method + Genetic algorithm. (Green) Newton's method. (Blue) Genetic algorithm. (Black) Random position and time of immersion.

V. Conclusions

We have proposed an algorithm for coverage optimization of an underwater surface. The unawareness of the position of the data diver and the lack of control while underwater imply that the trajectory should be estimated depending on the position and time of immersion. The optimization problem is solved using a combination of genetic algorithm and Newton's method. The solution provides a minimum expectation of the coverage that will be carried out during the immersion. Future work will include the estimation and improvement of the current flow model using the data obtained by the vehicles, the extension to more complex notions of coverage that incorporate requirements from computer vision to better reconstruct the ocean floor, and the investigation of scenarios with cooperative multi-agent systems.

References

[1] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
[2] J. Cortes, S. Martinez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 11, no. 4, pp. 691–719, 2005.
[3] S. Martinez, "Distributed interpolation schemes for field estimation by mobile sensor networks," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 491–500, 2010.
[4] L. C. Pimenta, V. Kumar, R. C. Mesquita, and G. A. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *47th IEEE Conference on Decision and Control (CDC)*, 2008, pp. 3947–3952.
[5] C. H. Caicedo-Nunez and M. Zefran, "Performing coverage on nonconvex domains," in *IEEE International Conference on Control Applications (CCA)*, 2008, pp. 1019–1024.

$$H_1 = \int\limits_{\mathcal{D}} \left(\frac{\partial^2 p(C(\tilde q),\hat S(\tilde q,T,q_0))}{\partial \hat S^2} \left(\frac{\partial \hat S(\tilde q,T,q_0)}{\partial T} \right)^2 + \frac{\partial p(C(\tilde q),\hat S(\tilde q,T,q_0))}{\partial \hat S} \frac{\partial^2 \hat S(\tilde q,T,q_0)}{\partial T^2} \right) d\tilde q \tag{17}$$

$$H_2 = \int\limits_{\mathcal{D}} \left(\frac{\partial^2 p(C(\tilde q),\hat S(\tilde q,T,q_0))}{\partial \hat S^2} \left(\frac{\partial \hat S(\tilde q,T,q_0)}{\partial T} \right) \left(\frac{\partial \hat S(\tilde q,T,q_0)}{\partial q_0} \right) + \frac{\partial p(C(\tilde q),\hat S(\tilde q,T,q_0))}{\partial \hat S} \frac{\partial^2 \hat S(\tilde q,T,q_0)}{\partial T \partial q_0} \right) d\tilde q \tag{18}$$

$$H_3 = \int\limits_{\mathcal{D}} \left(\frac{\partial^2 p(C(\tilde q),\hat S(\tilde q,T,q_0))}{\partial \hat S^2} \left(\frac{\partial \hat S(\tilde q,T,q_0)}{\partial q_0} \right)^\top \left(\frac{\partial \hat S(\tilde q,T,q_0)}{\partial q_0} \right) + \frac{\partial p(C(\tilde q),\hat S(\tilde q,T,q_0))}{\partial \hat S} \frac{\partial^2 \hat S(\tilde q,T,q_0)}{\partial q_0^2} \right) d\tilde q \tag{19}$$

where

$$\frac{\partial \hat S(\tilde q,T,q_0)}{\partial T} = \hat s(\tilde q, \Phi^T_{\hat f}(q_0)), \qquad \frac{\partial \hat S(\tilde q,T,q_0)}{\partial q_0} = \int\limits_{t_k^{\mathrm{im}}}^{t_k^{\mathrm{im}}+T} \frac{\partial \hat s(\tilde q, \Phi^t_{\hat f}(q_0))}{\partial \Phi^t_{\hat f}} \frac{\partial \Phi^t_{\hat f}(q_0)}{\partial q_0} dt,$$

$$\frac{\partial^2 \hat S(\tilde q,T,q_0)}{\partial T^2} = \frac{\partial \hat s(\tilde q, \Phi^T_{\hat f}(q_0))}{\partial \Phi^T_{\hat f}} \hat f(\Phi^T_{\hat f}), \qquad \frac{\partial^2 \hat S(\tilde q,T,q_0)}{\partial q_0 \partial T} = \frac{\partial^2 \hat S(\tilde q,T,q_0)}{\partial q_0 \partial T} = \frac{\partial \hat s(\tilde q, \Phi^T_{\hat f}(q_0))}{\partial \Phi^T_{\hat f}} \frac{\partial \Phi^T_{\hat f}(q_0)}{\partial q_0},$$

$$\frac{\partial^2 \hat S(\tilde q,T,q_0)}{\partial q_0^2} = \int\limits_{t_k^{\mathrm{im}}}^{t_k^{\mathrm{im}}+T} \left(\left(\left(\frac{\partial \Phi^t_{\hat f}(q_0)}{\partial q_0} \right)^\top \otimes \mathbb{I}_1 \right) \frac{\partial}{\partial q_0} \frac{\partial \hat s(\tilde q, \Phi^t_{\hat f}(q_0))}{\partial \Phi^t_{\hat f}} + \left(\mathbb{I}_2 \otimes \frac{\partial \hat s(\tilde q, \Phi^t_{\hat f}(q_0))}{\partial \Phi^t_{\hat f}} \right) \frac{\partial}{\partial q_0} \frac{\partial \Phi^t_{\hat f}(q_0)}{\partial q_0} \right) dt$$

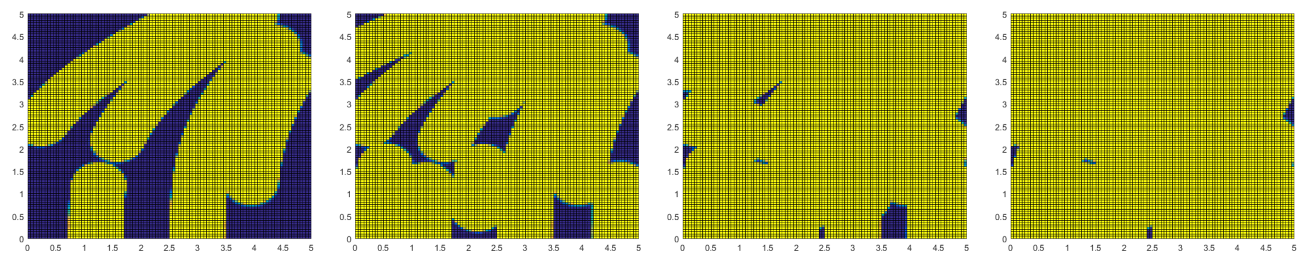

Fig. 4. Evolution of the coverage of the region $\mathcal{D}$ after 5, 10, 20 and 30 immersions, respectively.

[6] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, and D. Rus, "Voronoi coverage of non-convex environments with a group of networked robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 4982–4989.

[7] A. Arsie and E. Frazzoli, "Efficient routing of multiple vehicles with no communication," in *American Control Conference (ACC)*, 2007, pp. 449–454.

[8] P. F. Hokayem, D. Stipanovic, and M. W. Spong, "Dynamic coverage control with limited communication," in *American Control Conference (ACC)*, 2007, pp. 4878–4883.

[9] G. M. Atinç, D. M. Stipanović, P. G. Voulgaris, and M. Karkoub, "Swarm-based dynamic coverage control," in *53rd IEEE Conference on Decision and Control (CDC)*, 2014, pp. 6963–6968.

[10] D. Panagou, D. M. Stipanovic, and P. G. Voulgaris, "Distributed dynamic coverage and avoidance control under anisotropic sensing," *IEEE Transactions on Control of Network Systems*, 2016, early access.

[11] S. Miah, B. Nguyen, A. Bourque, and D. Spinello, "Nonuniform coverage control with stochastic intermittent communication," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1981–1986, 2015.

[12] J. R. Stack and C. M. Smith, "Combining random and data-driven coverage planning for underwater mine detection," in *Proceedings OCEANS 2003*, vol. 5, 2003, pp. 2463–2468.

[13] S. Alam and Z. J. Haas, "Coverage and connectivity in three-dimensional underwater sensor networks," *Wireless Communications and Mobile Computing*, vol. 8, no. 8, pp. 995–1009, 2008.

[14] E. Cayirci, H. Tezcan, Y. Dogan, and V. Coskun, "Wireless sensor networks for underwater surveillance systems," *Ad Hoc Networks*, vol. 4, no. 4, pp. 431–446, 2006.

[15] Y.-S. Jung, K.-W. Lee, S.-Y. Lee, M. H. Choi, and B.-H. Lee, "An efficient underwater coverage method for multi-auv with sea current disturbances," *International Journal of Control, Automation and Systems*, vol. 7, no. 4, pp. 615–629, 2009.

[16] L. Paull, S. Saeedi, M. Seto, and H. Li, "Sensor-driven online coverage planning for autonomous underwater vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 6, pp. 1827–1838, 2013.

[17] E. Galceran, R. Campos, N. Palomeras, D. Ribas, M. Carreras, and P. Ridao, "Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles," *Journal of Field Robotics*, vol. 32, no. 7, pp. 952–983, 2015.

[18] H. Khalil, *Nonlinear systems (3rd ed.)*. Prentice Hall, 2002.

[19] Apium, "Data diver — apium, swarm robotics," 2016, [Accessed 2-March-2018]]. [Online]. Available: http://apium.com/data-diver/

[20] H. Fang, R. A. De Callafon, and J. Cortés, "Simultaneous input and state estimation for nonlinear systems with applications to flow field estimation," *Automatica*, vol. 49, no. 9, pp. 2805–2812, 2013.

[21] H. Fang, R. A. Callafon, and P. J. Franks, "Smoothed estimation of unknown inputs and states in dynamic systems with application to oceanic flow field reconstruction," *International Journal of Adaptive Control and Signal Processing*, vol. 29, no. 10, pp. 1224–1242, 2015.

[22] I. I. Hussein and D. M. Stipanovic, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 642–657, 2007.

[23] A. De Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot motion planning and control*, J.-P. Laumond, Ed. Springer, 1998, pp. 171–253.

[24] K. Deb, *Optimization for engineering design - algorithms and examples*. PHI Learning Private Limited, 2012.