CrossMark

Real-time adaptive multi-robot exploration with application to underwater map construction

Athanasios Ch. Kapoutsis[1,2] · **Savvas A. Chatzichristofis**[1,2] · **Lefteris Doitsidis**[3] ·
João Borges de Sousa[4] · **Jose Pinto**[4] · **Jose Braga**[4] · **Elias B. Kosmatopoulos**[1,2]

Abstract This paper deals with the problem of autonomous exploration of unknown areas using teams of Autonomous X Vehicles (AXVs)—with X standing for Aerial, Underwater or Sea-surface—where the AXVs have to autonomously navigate themselves so as to construct an accurate map of the unknown area. Such a problem can be transformed into a dynamic optimization problem which, however, is NP-complete and thus infeasible to be solved. A usual attempt is to relax this problem by employing greedy (optimal one-step-ahead) solutions which may end-up quite problematic. In this paper, we first show that optimal one-step-ahead exploration schemes that are based on a *transformed optimization criterion* can lead to highly efficient solutions to the multi-AXV exploration. Such a transformed optimization criterion is constructed using both theoretical analysis and experimental investigations and attempts to minimize the "disturbing" effect of deadlocks and nonlinearities to the overall exploration scheme. As, however, optimal one-step-ahead solutions to the transformed optimization criterion cannot be practically obtained using conventional optimization schemes, the second step in our approach is to combine the use of the transformed optimization criterion with the cognitive adaptive optimization (CAO): CAO is a practicably feasible computational methodology which adaptively provides an accurate approximation of the optimal one-step-ahead solutions. The combination of the transformed optimization criterion with CAO results in a multi-AXV exploration scheme which is both practically implementable and provides with quite efficient solutions as it is shown both by theoretical analysis and, most importantly, by extensive simulation experiments and real-life underwater sea-floor mapping experiments in the Leixes port, Portugal.

Keywords Path planning for multiple mobile robot systems · Trajectory generation · Cognitive robotics · Mapping · Marine robotics

✉ Athanasios Ch. Kapoutsis
  athakapo@iti.gr

  Savvas A. Chatzichristofis
  schatzic@iti.gr

  Lefteris Doitsidis
  ldoitsidis@chania.teicrete.gr

  João Borges de Sousa
  jtasso@fe.up.pt

  Jose Pinto
  zepinto@fe.up.pt

  Jose Braga
  jmab@fe.up.pt

  Elias B. Kosmatopoulos
  kosmatop@iti.gr

[1] Democritus University of Thrace, Xanthi, Greece

[2] Information Technologies Institute, CERTH, Thessaloniki, Greece

[3] Department of Electronic Engineering, Technological Educational Institute of Crete, Chania, Greece

[4] Faculdade de Engenharia da Universidade do Porto, Porto, Portugal

1 Introduction

Recent technological advances have made the usage of teams of Autonomous X Vehicles (AXVs)—where X can stand for Aerial, Underwater or Sea-surface—more appealing in a variety of missions (Birk et al. 2012), which may include harbor security (Reed et al. 2010; Rodningsby and Bar-Shalom 2009; Kessel and Hollett 2006), post-disaster infrastructure inspection (Murphy et al. 2009), underwater archaeology (Roman and Mather 2010), continuous infrastructure moni-

toring to prevent accidents (DeVault 2000), habitat mapping (Blondel 2008), spy missions, unmanned weapon and other military activities (Khurshid and Bing-rong 2004) as well as agricultural activities such as example paddy monitoring and spraying (Samad et al. 2013). In all the aforementioned missions there are several factors that affect the performance of the AXV (in the case of a single vehicle) or of the overall team (in the case of a team of AXVs operating simultaneously). These are related with the technological limitations of the hardware which is used and the methodologies that process and fuse data to obtain valid conclusions related with the actual AXV performance. A key element of success in almost every mission is the ability to produce valid maps by utilizing all the available resources.

There are, basically, two different problems that the team of AXVs faces when deployed in missions such as the ones mentioned above. The first of these problems has to do with the ability of the AXVs to process their sensor measurements so as they create accurate maps of the environment. As creation of accurate maps requires the AXVs to "know where they are", such a problem is also known as the Simultaneous Localization and Mapping (SLAM) problem, i.e., the problem of processing the AXVs sensor measurements so as to simultaneously identify "where they are" and create the map of the external environment. The second of the problems has to do with "which trajectories the AXVs have to follow", i.e., the problem[1] of *trajectory generation* for the AXVs so as to maximize SLAM efficiency.

Most of the research work has concentrated on the problem of SLAM (in case of single-AXVs) or Cooperative SLAM (C-SLAM) in case where a team of AXVs is deployed. Very powerful SLAM and C-SLAM methodologies have been proposed recently and have successfully demonstrated in real-life situations. Despite, however, these advances, the vast majority of missions rely on pre-specified AXV trajectories. In other words, the trajectory the AXV has to follow is designed off-line, before its actual deployment. As the AXV is called to map a partially known or, in some cases, a totally unknown area, off-line designing of the AXV trajectories may become quite problematic: first of all, the off-line design is quite likely to "miss" areas of crucial information; moreover, it may lead the AXV to "waste" time mapping areas of little information. For this reason, in practice, AXV-based mapping is accomplished by employing a costly and tedious repetitive procedure: firstly, an original trajectory is designed off-line, the AXV is then deployed and maps the area following to the off-line designed trajectory, then based on the created map a new trajectory is designed off-line

again, the AXV is deployed according to this new trajectory, and so on. Apart from the fact that such a procedure is costly and tedious, it renders prohibitive the deployment of AXV in time-critical mapping missions or in cases where there are limited resources available, such as detection of sunken drums leaking chemicals or search-and-rescue missions. Most importantly, off-line generation of the AXV trajectories cannot take advantage and exploit the cooperative capabilities in case a team of AXVs is employed. Typically, multi-AXV deployment for mapping purposes employ again pre-specified trajectories with no or little interaction between the AXVs or, in the best case, the AXVs communicate with each other so as to improve their localization estimates and/or to make sure that they are moved in certain formation. However, full exploitation of the cooperative capabilities of a multi-AXV system cannot be accomplished by having the AXVs moving along pre-specified trajectories or in formation: the cooperation between more than one AXVs can speed up considerably the overall mapping process, by having the AXVs coming closer in areas of high importance and by having the AXVs sharing sensor measurements and mapping information.

To overcome the shortcomings of off-line trajectory generation, many different approaches have been proposed which attempt to generate in real-time the AXVs trajectories so as to maximize the overall C-SLAM efficiency, see Sect. 2 for more details. There are, however, several theoretical and practical limitations that prevent these approaches from becoming a generic and practicable tool that will provide efficient trajectory generation: the fact that trajectory generation for maximizing SLAM efficiency is a difficult-to-be-solved optimization problem, the strong reliance of trajectory generation to the particular SLAM methodology employed, the highly non-linear nature of sensor noise and the limited communication capabilities of the AXVs are among the most important of such limitations. In this paper, we propose and evaluate both using theoretical analysis and simulations as well as real-life experiments a new methodology that attempts to overcome such limitations.

One of the most severe limitations of multi-robot trajectory generation for maximizing SLAM efficiency is the fact that such a problem is an NP-hard optimization problem. Most of the existing approaches employ one-step-ahead optimization or relaxed versions of the NP-hard trajectory generation optimization problem to overcome such a limitation. Such approaches, however, may end-up being quite problematic. First of all, the closed-form (i.e., analytical mathematical form) that relates the SLAM efficiency to the overall multi-robot team dynamics is not easy to be calculated. However, calculating of the analytical form of SLAM efficiency is the least of the problems encountered: the most severe problem is due to the fact that optimizing the SLAM efficiency may lead to severe *deadlocks* or, mathematically speaking, to getting

---

[1] The problem of multi-robot *trajectory generation for maximizing SLAM efficiency* is also referred in the literature as *exploration* or *optimal motion strategy*. In the rest of this paper, these terms will be used interchangeably.

stuck into local maxima. As a matter of fact, as we report in the next sections, one-step-ahead optimization of the SLAM efficiency can lead to situations where the AXVs get stuck to deadlocks even after they have accomplished only 10–20 % of their mapping mission. A similar situation is also present if relaxations to the original NP-hard problem are employed. Moreover, any approach used for optimizing the SLAM efficiency has to deal with another problem: as, typically SLAM algorithms are based on linearized or approximate models for the sensor dynamics, optimizing the SLAM efficiency does not guarantee that poor performance or, even, divergence of the SLAM procedure is avoided.

To overcome all the above shortcomings and limitations, a new approach is employed and analyzed in this paper. According to this new approach, the AXVs trajectories are calculated so as to optimize a transformed version of SLAM efficiency: such a transformation guarantees that deadlocks are avoided and, moreover, that the AXVs move towards minimizing the effect of sensor non-linearities which, in turn, implies minimizing the problem of poor performance/divergence of the SLAM procedure. Theoretical analysis establishes these properties and, moreover, simulation experiments exhibit that the use of such a transformed version can improve significantly the performance of the exploration scheme.

As in the case of SLAM efficiency, it is difficult—if not possible—to obtain an analytical form for the transformed version of the SLAM efficiency. To overcome such a problem, we employ Cognitive Adaptive Optimization methodology (CAO), an adaptive optimization approach that does not require the availability of analytical forms of the function to be optimized (Kosmatopoulos et al. 2007; Kosmatopoulos 2009; Kosmatopoulos and Kouvelas 2009). It has to be emphasized that CAO—successfully implemented to the problem of multi-robot optimal surveillance coverage (Renzaglia et al. 2012)—is computationally simple and thus scalable. Both simulation experiments and theoretical analysis establish that the use of CAO, combined with the transformed version of the SLAM efficiency, guarantees efficient performance of the overall exploration.

Apart from the limitations of existing approaches that have to do with the complexity of the overall exploration problem, it has to be emphasized that most of these approaches suffer from other shortcomings that render their application in real-life quite difficult: for instance, some of these approaches impose certain requirements for the AXV communication/sensing system and/or they strongly rely on a particular SLAM method. The development of the proposed approach is performed so as to avoid the above-mentioned shortcomings. As a matter of fact, rather than imposing requirements in the communication/sensing system and the SLAM methodology employed, the philosophy of the proposed approach is "to do the best it can" given the communication/sensing/SLAM system, allowing even cases where the multi-robot team comprises AXVs with mutually different sensing capabilities or operating different SLAM algorithms. In such a way, the addition/removal of an AXV (with probably different sensing capabilities than the existing ones or operating a different SLAM technique than the other AXVs) is performed "automatically".

We close this section by mentioning that real-life underwater sea-floor mapping experiments were conducted using two AUV (Autonomous Underwater Vehicles) in the port of Porto, Portugal. The experiments exhibit the practicability and "ease-to-operate" of the proposed approach. Despite the fact that, the theoretical approach (Sect. 3) along with the simulation results have been made under communication constraints, where at each time-step, every AXV has to be at most a maximum distance from its closest AXV (Martijn 2007), in real-life experiments, it is assumed that the AXVs modems can transmit/receive from the whole operation area. This limitation is not unrealistic (at least until a specific order of magnitude in the square meters of the operation area), taking for granted that in the exhibited real-life experiments, the AXVs are able to transmit in each time-step their measurements back to central station. Additionally, the communication protocol that is used is well established in terms of the modern communication methods (Pfingsthorn et al. 2010; Johnson et al. 2009; Rajala and Edwards 2007) available bandwidth. The last remark regarding to the communications is that the theoretical approach along with the simulation results have been made independently on the place where the new waypoints calculation is taking place (either on a central command station or assuming a processor on one of the AXVs). This feature leverage the proposed approach with the ability of a broader appliance in AXVs mapping missions.

The rest of the paper is organized as follows. In Sect. 2 we provide a short literature review concerning the methodologies used to generate trajectories for the task of multi-robot exploration, while in Sect. 3 we attempt to provide a description of the set-up of map construction using a team of AXVs. In Sect. 4 we formulate the problem of autonomous navigation/exploration of a team of AXVs as an optimization problem, while in Sect. 5 we present the proposed CAO based approach which allows the autonomous navigation/exploration of a team of AXVs. Finally details simulations that present the applicability of our approach are presented in Sect. 6 while in Sect. 7 we present how our approach was implemented as a plug-and-play tool for the navigation of a set of real autonomous underwater vehicles in the area of Oporto's Harbor in Portugal. Some concluding remarks and ideas for future research are discussed in Sect. 8.

2 Related work and positioning

2.1 Optimal control/dynamic-programming techniques

Mathematically speaking, single- or multi-robot exploration, i.e., the on-line generation of robot trajectories so as to maximize SLAM accuracy and efficiency is an NP-hard (Donald et al. 1993; Pasqualetti et al. 2012) or, equivalently, a non-convex dynamic optimization problem. As a result, any attempt to generate the globally optimal solution is not possible to end up with a computationally and practically feasible solution. For this reason, many different classes of approaches have been proposed whose main idea is based on solving, instead of the original NP-hard problem, a simplified or relaxed version of it. In such a way, a computationally feasible solution is possible to be constructed at the expense, of course, of sacrificing global optimality. For instance, there are several methodologies which are based on relaxed or linearized version of the exploration problem when it is formulated as an *optimal-control* or *dynamic-programming* problem, see e.g., Ny and Pappas (2009), Milam et al. (2000), Kelly and Nagy (2003), Low et al. (2011), and Tabuada and Pappas (2005). In all these approaches, the original optimal control or dynamic programming problem for optimizing the SLAM accuracy/efficiency is simplified, relaxed and/or linearized so as to come up with a—suboptimal—solution that is computationally feasible.

2.2 Optimal one-step-ahead/Greedy approaches

In another class of approaches, instead of relaxing the optimal control or the dynamic programming problem over a time horizon, relaxed versions of the optimal one-step-ahead optimization problem are pursued. In this family of approaches, the next AXVs poses are chosen greedily so as to optimize an appropriately defined quality/objective function that is related to the SLAM accuracy and efficiency. Some of these approaches are constructed by analyzing the EKF equations (Msechu et al. 2008; Zhou and Roumeliotis 2011) or by relaxing the mathematical models (Zhou and Roumeliotis 2011) that carry out the aspects of the trajectories generation problem (e.g., sensor model). In this subgroup a common practice (Feder et al. 1999; Bourgault et al. 2002; Stachniss and Burgard 2003) is to choose the next action χ that maximizes the expected information gain, e.g., $\text{argmin}_\chi (tr (\mathcal{P}))$, where $tr (\mathcal{P})$ is the trace of the EKF error covariance matrix. Another subset of approaches (John 2003; Beard et al. 2002) optimizes a *quality function* that en-captures, in the best possible way, the goals of the trajectory generation problem without any prior knowledge about the terrain morphology or the underlying dynamics. The trajectories are chosen so as to optimize this quality function, using in most of the cases *model-free Gradient-decent*-like approaches (Nesterov

2007). The selection of the adequate quality function is not trivial.

Both dynamic programming/optimal control and optimal one-step-ahead approaches suffer from the following drawbacks:

– it is not easy to be evaluated, whether the relaxed solutions do indeed provide a solution to the robot exploration problem that is actually efficient.
– the nonlinearities (such as in cases where the noise does not follow the Additive Gaussian White Noise model) may give rise to undesirable estimator's divergence.
– in many cases, they get stuck to deadlocks or, equivalently, to local minima (see our discussion in Sect. 4).

Researchers of the field, perceiving these disadvantages, have established and successful evaluated using numerous of experiments results, deadlock recovery mechanisms for greedy approaches (eg Martijn (2007) and Rathnam and Birk (2013)) which are resistant to the aforementioned shortcomings

Unfortunately, these are not the only problems, and this is where this work aims to contribute. In order to attack the problem at hand, the vast majority of the optimal/dynamic programming approaches do not take into account the nonlinear characteristics of the AXVs sensors. For example, it is usually considered that an AXV can accurately estimate the positions of a landmark/cell as soon as it "sees" it. This simplifying assumption is crucial for most of the existing approaches so as to be able to calculate in real-time the cost function to be optimized as well as the next AXV locations that greedily optimize such a cost function. Moreover, such an assumption is crucial for overcoming deadlocks (local minima) which are frequently encountered when greedy approaches are employed. The main contribution of the proposed approach is to alleviate the above-mentioned simplifying assumption by introducing a more realistic one that takes the nonlinear sensor characteristics into account (in which case, for instance, the accurate estimation of a landmark depends on the distance and the time the AXV "spends" when sensing it). The removal of the above-mentioned assumption renders the existing approaches not practicable as it is not possible anymore to calculate in real-time the cost function as well as the locations of the AXVs that greedily optimize it. Moreover, it renders not practicable the use of the techniques used in the existing approached for avoiding or escaping deadlocks. For this reason, the proposed approach introduces a new methodology for (a) estimating in real-time the cost function and (b) avoid deadlocks.

A large worth-mentioned variation of this *optimal one-step-ahead*, class exploits the idea of the frontier cell navigation (Fox et al. 2006; Martijn 2007; De Hoog et al. 2009; Rathnam and Birk 2013; Freda and Oriolo 2005; Bur-

gard et al. 2000) firstly introduced by Yamauchi (1998). The majority of the one-step-ahead (greedy), path planning approaches utilize that fundamental idea and in the most cases they augment their solutions upon the frontier cell concept. Our approach is not an exception, as the omission of this concept would seriously compromise the overall successfulness of the mission. Over-and-above, the proposed approach introduces a supplementary improvement, regarding to frontier cell concept. In the proposed approach the next AXVs positions are at most a predefined distance (which is directly related to the maximum possible speed of the AXVs). This constraint in AXVs movement adds an extra layer of efficiency in the sense that it strictly bounds the waiting time for each AXV, from the moment it reaches the desired waypoint until the new one. Additionally, and in correspondence to the majority of the members of this class, we further investigate the construction of the objective function (Sect. 4), adding secondary terms in order to avoid undesirable deadlocked situations. But the development of another *optimal one-step-ahead* technique is not the main contribution of the paper. This paper attempts to prove that even more complicated and unknown problems of the ones that belong to the target group of that class can be adequately addressed, problems where the objective/reward function is not a metric that can be computed a-priori. Overall, the theoretical contribution of this paper should be classified as an *approximately optimal one-step-ahead approach* and in particular in the model-free class.

2.3 Sampling-based approaches

Another widely-spread trajectory generation approach, is the *sampling-based* ones, including the Probabilistic RoadMaps (PRMs) (Kavraki et al. 1996, 1998; Prentice and Roy 2009) and Rapidly-exploring Random Trees (RRTs) (Kuffner and LaValle 2000; LaValle 2006). These approaches construct a graph (the roadmap), which represents a random set of collision-free (obstacle avoidance) trajectories, and then by computing the shortest path that connects the initial state with a final state through the roadmap, provides the near-optimum trajectory. Given this graph, a feasible, collision-free path can be found using a standard graph search algorithm from the start node to the goal node. Besides the fact that the trajectories are almost optimal, the above approach has some vital drawbacks:

- The whole process strongly relies on the prior knowledge about the morphology, rendering the method unsuitable for exploration of unknown territories.
- These algorithms provide the best trajectories strictly among the starting and ending point.

- Both theoretical analysis and real-world experiments (Valencia et al. 2011) are limited mostly to single-robot approaches.

An interesting variation (Birk and Carpin 2006) of this class, utilizes the occupancy-grid concept (Burgard et al. 2000) under real-life communication constraints, to achieve cooperative exploration.

2.4 Simulation-based approaches

Finally, we mention that another class of approaches for the development of robot exploration approaches employs *simulation-based* optimization, see e.g. Kollar and Roy (2008), Martinez-Cantin et al. (2009), Peng et al. (2005), and Doucet (1998). In simple words, the idea behind these approaches is the following: first a parametrized decision-making mechanism is devised for generating on-line the robot trajectories, with different choices for its parameters leading to different exploration decision-making mechanisms. Then, realistic simulations or similar tools are used in order to optimize the parameters of the decision-making mechanism. Thus, conceptually, many of the optimization computations that otherwise would take place on-line are "moved" off-line. The problem with such approaches is that, first, the simulations need to cover a wide range of different realistic scenarios (and thus they may become "expensive") and, moreover, since the dimensionality of the optimization problem is quite high, a large number of parameters is needed in order to come up with an efficient decision-making mechanism. To overcome such problems, most of the proposed approaches assume sufficient *a priori* knowledge of the map to be explored, in which case practicable solutions can be devised.

3 The set-up

In this section, we describe with the set-up assumed in this paper, for map construction using a multi-robot team of N_R AXVs, where N_R denotes the number of AXVs.

3.1 Optimal quantized map

Without loss of generality, let us assume that the area to be mapped is constrained within a rectangle in the (x, y)-coordinates, i.e., the AXVs are called to map the area constrained in the (x, y)-coordinates as follows:

$$\mathcal{U} = \left\{ x, y : x \in [x_{min}, x_{max}], y \in [y_{min}, y_{max}] \right\}$$

where $x_{min}, x_{max}, y_{min}, y_{max}$ are real numbers that define the "borders" of the area to be mapped. Using the definition of $\mathcal{U}$, the area can be then defined as a function that corresponds each point $(x, y) \in \mathcal{U}$ to a point $z = z(x, y)$ [height of map at (x, y)]. Let us also consider a *fixed* set of N_L pairs $(x_\ell, y_\ell), \ell = 1, \ldots, N_L$ that are distributed in $\mathcal{U}$ [typically (x_i, y_i) are uniformly distributed in $\mathcal{U}$] and let $z(x_\ell, y_\ell)$ denote map's value at the point (x_ℓ, y_ℓ). Typically, the map construction problem can be transformed to the one of finding the "best grid" $(x_\ell, y_\ell, z_{L,\ell}), \ell = 1, \ldots, N_L$ such that the *quantized map*

$$z_q(x, y) = \sum_{\ell=1}^{N_L} z_{L,\ell} \phi_i(x, y, x_\ell, y_\ell)$$

approximates the actual map $z(x, y)$ as accurately as possible, i.e., the map construction problem is transformed into the problem of finding the parameters $z_{L,\ell}, \ell = 1, \ldots, N_L$ that minimize the following criterion:

$$\int_{(x,y)\in\mathcal{U}} \| z(x, y) - \sum_{\ell=1}^{N_L} z_{L,\ell} \phi_\ell(x, y, x_\ell, y_\ell) \|^2 \, dx dy \quad (1)$$

The functions $\phi_\ell(\cdot) \in [0, 1]$ in the above equation correspond to the so-called *basis functions*. Typical choices for $\phi_\ell(\cdot)$ is the piecewise constant function:

$$\phi_\ell(x, y, x_\ell, y_\ell)$$
$$= \begin{cases} 1 & \text{if } \sqrt{(x - x_\ell)^2 + (y - y_\ell)^2} \\ & = \min_{j=1,\ldots,L} \sqrt{(x - x_j)^2 + (y - y_j)^2} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

or, the Gaussian function:

$$\phi_\ell(x, y, x_\ell, y_\ell) = \exp(-(x - x_\ell)^2 - (y - y_\ell)^2) \quad (3)$$

Let also

$$\mathbf{X}^L = \begin{bmatrix} x_1, y_1, z_{L,1}(x_1, y_1) \\ \vdots \\ x_{N_L}, y_{N_L}, z_{L,N_L}(x_{N_L}, y_{N_L}) \end{bmatrix}$$

Hereafter, we will define $\mathbf{X}^L$ to be the matrix of *map landmarks* associated with the map $z(x, y)$.

We close this section by mentioning that in the experiments described in this paper, the basis functions were chosen to be the ones given in (3).

3.2 AXVs sensors

Before we continue on the map construction problem, let us first provide with some necessary preliminaries as far as

the AXV sensors are concerned. The AXVs are equipped with sensors that provide proprioceptive measurements (e.g., from GPS labels or inertial sensors) to propagate their state (position and orientation) estimates as well as exteroceptive measurements (e.g., cameras, sonars, bathymeters, etc) that enable them to measure their distance or bearing from points of interest. Let x_i^R denote the position(in a 3-D space) of the ith AXV [2] and

$$\mathbf{X}^R = \begin{bmatrix} x_1^{R\tau} \\ \vdots \\ x_{N_R}^R{}^\tau \end{bmatrix}$$

denote the matrix of all AXVs positions (*team configuration*). Furthermore, let $\mathbf{Y}$ denote the vector of all AXVs' sensor measurements. In the most general case, the sensor measurements are related to the matrices $\mathbf{X}^L$ and $\mathbf{X}^R$ through a nonlinear function that admits the form

$$\mathbf{Y} = H(\mathbf{X}^L, \mathbf{X}^R, \varXi)$$

where H is the nonlinear vector sensor function and $\varXi$ is the sensor measurement noise vector.

The design for map construction using AXVs will have to take into account the—sometimes severe—limitations of the environment the AXVs operate on: nonlinear sensor noise characteristics, limited communication range, and limited visibility of the AXVs sensors are some of the limitations that render multi-AXV map construction a very challenging task. Below, these limitations are described in more detail:

(NL-Noise) The typical assumption made in most robotic applications that the sensor noise is additive Gaussian noise is very restrictive and not realistic in many AXV applications as the sensor noise affects the sensor measurements in a Non-linear fashion. For instance, in the case of sonar or vision sensors, the noise affecting such sensors is proportional to the sensor-to-sensing point distance, i.e., the larger is the AXV-to-sensing point distance, the larger is the sensor noise. Similarly, in the case of localization-related sensors, the larger is the time the AXV is operating without GPS recalibration, the larger is the localization noise. As a result, it is more realistic to assume a multiplicative sensor noise model that takes the form:

$$y = h(x^R, q) + h_\xi(x^R, q, t_{uw})\xi \quad (4)$$

where y is the sensor measurement, x^R, q are the positions of the AXV and the sensing point, respectively, $h(x^R, q)$

---

[2] For simplicity, we assume that the orientation of the AXVs is fixed and constant all the time. All the results of this paper can be easily extended in the case where the orientation changes by the navigation algorithm.

is the sensor model in the noise-free case, $h_\xi(x^R, q, t_{uw})$ is the multiplicative sensor noise term, t_{uw} is the time the AXV is without GPS connection and ξ is a standard Gaussian noise.

(LimComRange) In order to establish the required communication connection for the data transferring between the AXVs, an additional constraint is needed. For ease of comprehension, and without loss of generality, it can be considered that all AXVs have the same communication range, represented as com_R. As a result, an AXV must make sure—at each time-instant—that there is at least one other AXV that is within com_R distance from it.

(LimVis) In addition to the aforementioned limitations, the AXV exteroceptive sensors are of limited visibility. As a result, additionally to the nonlinear sensor noise assumption (4), the sensor model for the exteroceptive sensors should be augmented to count for the limited visibility constraint. Moreover, the sensor model must be augmented to count for the case where there is no line-of-sight between the AXV and the sensing point (e.g., there is an obstacle in between). As a result, the actual sensor model becomes:

$$y_{x^R-q} = \begin{cases} \text{undefined} & \text{if } \|x^R - q\| \geq \text{thres} \\ \text{undefined} & \text{if there is no line-of-} \\ & \text{sight between } x^R \text{ and } q \\ h(x^R, q) + \\ h_\xi(x^R, q, t_{uw})\xi & \text{otherwise} \end{cases} \quad (5)$$

where y_{x^R-q} denotes the sensor measurement from an AXV at position x^R to a sensing point at position q, $thres$ denotes the visibility threshold beyond which the sensor does not "see" and $\|\cdot\|$ denotes the Euclidean norm.

3.3 Aperiodic AXV navigation/communication under communication constraints

Having defined the limitations the AXV sensors are facing, we now proceed by describing the way that the multi-robot team of AXVs is operating while performing the map construction task. More precisely, consider that the AXVs are initially deployed at the time instant t_0 with the AXVs initial positions being equal to $x_1^R(t_0), \ldots, x_{N_R}^R(t_0)$ [or, in a more compact notation, the overall multi-robot of AXVs initial position is $\mathbf{X}^R(t_0)$]. Then, by employing an autonomous navigation/exploration algorithm (to be described in next sections), the *next desired location* (waypoint) for each AXV is calculated and each AXV is navigated to this next desired location. Apparently, the time needed for each AXV to reach its desired location is not the same for all AXVs. As a result, some of the AXVs will have to wait (after they have reached their desired location), so as for the rest of AXVs to reach their respective next desired location. Please note that while executing the task of navigating to the next desired location,

the different types of AXV sensors operate using different *activation frequencies*: for instance, the IMU sensors may be activated many times while the AXVs are accomplishing their navigation task, while the exteroceptive sensors, such as sonars, cameras and bathymeters are typically activated once and as soon as the AXVs reach and stabilize at their desired location. Let $\Delta t_{1,nav}$ denote the time required for *all* of the AXVs to reach their next desired location as well as to accomplish the activation of their exteroceptive sensors. Then, as soon as activation of all exteroceptive sensors is accomplished, the AXVs communicate their sensor measurements—typically after some pre-processing—to a central station and receive back their next desired locations. Let $\Delta t_{1,com}$ denote the time required for all AXVs to communicate their sensor measurements and to receive back their next desired locations.

Using the above procedure we have that all of the AXVs have accomplished their navigation, sensing and communication tasks at the time-instant $t_1 = t_0 + \Delta t_{1,nav} + \Delta t_{1,com}$. At this time instant, the navigation/exploration algorithm calculates the new desired locations for the AXVs and the above-described procedure is repeated again. In a nutshell, the AXVs receive at the time-instances $t_1, t_2, t_3, \ldots$ their new desired locations (waypoints) where $t_1 = t_0 + \Delta t_{1,nav} + \Delta t_{1,com}$, $t_2 = t_1 + \Delta t_{2,nav} + \Delta t_{2,com}$, $t_3 = t_2 + \Delta t_{2,nav} + \Delta t_{2,com}$, ... As explained above, the time-intervals $[t_0, t_1), [t_1, t_2), [t_2, t_3), \ldots$ are not of the same length.

Please also note, that the overall exploration/ navigation process must be accomplished under *severe communication constraints*: due to these constraints, not all of the AXV sensor measurements can be communicated to the rest of the team. Moreover, due to these constraints, one or more AXVs must not be able to communicate with the others (and, with the central station) at certain time-instances, which renders the overall exploration/navigation problem more challenging. For these reasons, the information to be communicated by each AXVs must be such that :

(C1) *It exploits to the maximum the cooperation capabilities of the system* by making sure that mapping information received by one AXV that is useful to the other AXVs must be communicated.

(C2) Moreover, it must be able to efficiently operate when communication of one or more AXVs is lost at some time-instances or when the addition of a new member of the team is required "on-the-fly", i.e., when the team is in operation.

In the next sections, we will describe how we address these two issues that are related to the severe communication constraints of the operation environment.

3.4 Distributed cooperative estimation (SLAM) under communication limitations

The proprioceptive and exteroceptive sensor measurements are processed so as to perform both the map construction as well as the localization of the AXVs. Accurate localization is a prerequisite for accurate map construction and, as a result, the overall estimation procedure requires the efficient simultaneous localization and mapping (SLAM). As already mentioned in the Introduction, there exists quite powerful methodologies for *single-AXV SLAM*. Our approach is to develop a system that can take *any of these single-AXV* approaches and appropriately enhance them so as to develop a *cooperative-AXV SLAM* which will substantially increase the efficiency of single-AXV SLAM by embedding within it with cooperative information while, of course, making sure that the two objectives (C1), (C2) related to the communication constraints are satisfied. More precisely, our approach comprises enhancing single-AXV SLAM approaches by providing them with the "optimal" possible information received by the other AXVs. Below, we describe how such an approach is embedded within the proposed system.

Let us fix the number N_L of landmarks and let $\hat{\mathbf{X}}_L(0)$ denote an initial estimation of the map landmarks. Let also $\hat{\mathbf{X}}_R(0)$ denote the estimate of the initial AXV positions. Then, in general, a single-AXV SLAM system can be mathematically represented as follows:

$$
\begin{aligned}
&\left[\widehat{\mathbf{X}}_L^{(j)}(t_{i+1}), c_L^{(j)}(t_{i+1}), \hat{x}_R^{(j)}(t_{i+1}), c_R^{(j)}(t_{i+1}) \right] \\
&= \mathrm{EST}_j \left(\widehat{\mathbf{X}}_L^{(j)}(t_i), c_L^{(j)}(t_i), \hat{x}_R^{(j)}(t_i), c_R^{(j)}(t_i), y^{(j)}(t_{i+1}) \right)
\end{aligned}
\tag{6}
$$

where

- $\mathrm{EST}_j(\cdot)$ denotes the overall dynamics of the single-AXV SLAM system for the j-th AXV.
- $\hat{\mathbf{X}}_L^{(j)}(t_{i+1})$ denotes the estimated landmarks as generated by the single-AXV SLAM system for the j-th AXV.
- $c_L^{(j)}(t_{i+1})$ is an N_L-dimensional vector that corresponds to the *confidence level* of estimation of landmarks, i.e., the ℓ-th entry of the vector $c_L^{(j)}(t_{i+1})$ indicates the degree to how accurately the ℓ-th landmark has been estimated. For instance, in the case an Extended Kalman filter (EKF) is employed for performing the landmark estimation, the confidence level vector typically corresponds to the diagonal elements of the EKF error covariance matrix.
- $\hat{x}_R^{(j)}(t_{i+1})$ denotes the position estimate of the j-th AXV.
- $c_R^{(j)}(t_{i+1})$ is a positive number indicating the confidence level of estimation of the AXV position.
- $y^{(j)}(t_{i+1})$ is a vector comprising all sensor measurements that are available to the j-th AXV.

Note that in case of single-AXV missions, the sensor measurement vector $y^{(j)}(t_{i+1})$ comprises of the proprioceptive and exteroceptive sensor measurements of the single-AXV. In the case, however, of cooperative AXV missions, this vector can be augmented by using information that is communicated by the other AXVs. In other words, in the case of cooperative AXV missions the sensor measurement vector $y^{(j)}(t_{i+1})$ can be augmented as follows:

$y^{(j)}(t_{i+1}) = [$*local sensor info of the* j-*th AXV, sensor info from the other AXVs communicated to the j-th AXV* $]$.

The problem at hand becomes, then, to design the signals contained in *sensor info from the other AXVs communicated to the j-th AXV* so as to substantially improve the estimator's (6) efficiency by fully exploiting the cooperation capabilities [objective (C1)] by taking into account the restrictions and possible malfunctions [objective (C2)] of the communication system. Before, we proceed on how to design the signals contained in *sensor info from the other AXVs communicated to the j-th AXV*, we need some definitions, provided next.

Definition 1 We say that the ℓ-th landmark $\mathbf{X}_\ell^L = (x_\ell, y_\ell, z_\ell)$ is *visible* if there exists at least one AXV so that

- the AXV and $\mathbf{X}_\ell^L$ are connected by a line-of-sight;
- the AXV and the point $\mathbf{X}_\ell^L$ are at a distance smaller than the threshold value $thres$ defined in Eq. (5) [which corresponds to the maximum distance the AXVs' exteroceptive sensors can "see"].

Given a particular team configuration $\mathbf{X}^R(t_i)$, we let $\mathcal{V}(\mathbf{X}^R(t_i))$ denote the set of all indices ℓ for which $\mathbf{X}_\ell^L$ is visible at time-instant t_i. Similarly, we let $\mathcal{V}^{(j)}(\mathbf{X}^R(t_i))$ to denote the set of all landmarks that are visible by the j-th AXV.

In simple words, the subsets $\mathcal{V}^{(j)}(\mathbf{X}^R(t_i))$ and $\mathcal{V}(\mathbf{X}^R(t_i))$ provide us with information on the number of landmarks that are currently visible by the j-th AXV and by the overall AXV team, respectively.

Definition 2 We say that the ℓ-th landmark $\mathbf{X}_\ell^L(t_i)$ is *accurately-estimated* at the time-instant t_i, if the ℓ-th entry of the confidence level vector $c_L^{(j)}(t_{i+1})$ of at least one AXV is less than a given threshold[3]. Moreover, we define as $\mathcal{A}^L(t_i)$ the set of all indices ℓ for which $\mathbf{X}_\ell^L$ is accurately-estimated at time-instant t_i. Similarly, we define $\mathcal{A}^{(L,j)}(t_i)$ as the set of all indices ℓ for which $\mathbf{X}_\ell^L$ is accurately-estimated at time-instant t_i by the j-th AXV, i.e., $\mathcal{A}^{(L,j)}(t_i)$ corresponds to the

---

[3] Additionally, it might be useful to set an upper limit (big enough) in the times that a landmark can be estimated by at least one AXV with any accuracy. This limit will serve as deadlock avoidance meachnism in cases of a landmark cannot be accurately estimated, due to the local morphology of the area to be mapped. We would like to thank one of the reviewers who pointed that out.

landmarks that have become accurately estimated due to the j-th AXV.

Definition 3 We say that the ℓ-th landmark $\mathbf{X}_\ell^L(t_i)$ is *inaccurately-estimated but visible* at the time-instant t_i, if $\mathbf{X}_\ell^L(t_i) \in \mathcal{V}(\mathbf{X}^R(t_i))$ and $\mathbf{X}_\ell^L(t_i) \notin \mathcal{A}^L(t_i)$. Moreover, we define as $\mathcal{B}^L(t_i)$ the set of all indices ℓ for which $\mathbf{X}_\ell^L$ is inaccurately-estimated but visible at time-instant t_i. Similarly, we define $\mathcal{B}^{(L,j)}(t_i)$ as the set of all indices ℓ for which $\mathbf{X}_\ell^L$ is currently visible by the j-th AXV but have not yet become accurately estimated.

Definition 4 We say that the AXV $\mathbf{X}_j^R(t_i)$ is *accurately-estimated* at the time-instant t_i, if the confidence level $c_R^{(j)}(t_{i+1})$ is less than a given threshold. Moreover, we define as $\mathcal{A}^R(t_i)$ the set of all indices j for which $\mathbf{X}_j^R$ is accurately-estimated at time-instant t_i.

In simple words, the subsets $\mathcal{A}^L(t_i)$, $\mathcal{A}^R(t_i)$ provide us with information on the number of landmarks and AXVs, respectively, whose locations (positions) are currently accurately estimated, while the subset $\mathcal{B}^L(t_i)$ provides with information on the number of landmarks that are currently visible but have not yet been accurately estimated. Please note that, in the case of the subset $\mathcal{A}^L(t_i)$, if a particular index ℓ becomes a member of this set at some time, then it remains in this set for ever (once a landmark becomes accurately estimated, it remains accurately estimated for ever). This is not true for the set $\mathcal{A}^R(t_i)$ as an AXV whose position is currently accurately estimated may become inaccurately estimated later on. Moreover, please note that it suffices for at least one AXV to estimate accurately a landmark. It is worth noticing that the aforementioned concepts are graphically illustrated in Fig. 3.

We now return back to the problem of designing the signals contained in *sensor info from the other AXVs communicated to the j-th AXV*. As a first point, please notice that at the time-instant t_i, the j-th AXV SLAM estimator may produce a number of new accurately estimated landmarks as well as a number of new visible but not accurately estimated landmarks. Mathematically speaking, during the aforementioned time-instant the j-th AXV accurately estimates the landmarks that belong to the set $\mathcal{A}^{(L,j)}(t_i) \setminus \mathcal{A}^{(L,j)}(t_{i-1})$ while the landmarks that belong to the set $\mathcal{B}^{(L,j)}(t_i)$ are the ones that become visible but not accurately estimated by the j-th AXV. In the ideal case of infinite communications resources, the following information should become available from the j-th AXV to the rest of the team:

– The set of landmarks that have become accurately estimated at time-instant t_i by the j-th AXV, i.e., the set $\mathcal{A}^{(L,j)}(t_i) \setminus \mathcal{A}^{(L,j)}(t_{i-1})$. Information of this set is needed by the other AXVs so as to know which landmarks have been accurately estimated by the j-th AXVs so as for

them not to spend time estimating landmarks already accurately estimated. In other words, *this information is needed by the AXVs so as not to spend time exploring areas that have been already efficiently explored by some other AXV.*

– The current position estimate $\hat{x}_R^{(j)}(t_i)$ of the j-th AXV, the confidence level $c_R^{(j)}(t_i)$ as well as the sensor measurements that correspond to the $\mathcal{B}^{(L,j)}(t_i)$. Note that if this information were available to the other than the j-th AXV, they could include in their SLAM scheme sensor measurements acquired by the j-th AXV. In other words, if this information were available *each of the AXVs could use sensor information received by other AXVs as if it has been received by its own.* In such a way, the cooperative capabilities of the team could be exploited to the maximum extend.

As the team of AXVs has to operate under limited communication resources, exchanging the information described above is not possible. However, one can attempt to reduce the amount of the information described above by (i) tailoring this information so as it meets the communication requirements and (b) keep the information that can have the most significant impact to the SLAM estimators. This is done within the proposed system as follows:

– As a first step, for each AXV, the members of the team that belong to the *reachable exploration area of the particular AXV* are identified. The reachable exploration area for each AXV is estimated as follows: take any two AXVs, say the j-th and the k-th one. Let $\mathcal{P}$ denote the closest landmark [in the (x, y)-plane] to the k-th AXV that is currently visible by the j-th AXV. Then, the k-th AXV belongs to the reachable exploration area of the j-th AXV, if the distance in the (x, y)-plane between $\mathcal{P}$ and the position estimate of the k-th AXV is less than a threshold which corresponds to the distance the k-th AXV can travel in the time interval $[t_i, t_{i+1}]$. In other words, *if the k-th AXV does not belong to the reachable exploration area of the j-th AXV, then it is not possible for the k-th AXV to "see" any of the landmarks currently "seen" by the j-th AXV.*

– The landmarks that belong to the sets $\mathcal{A}^{(L,j)}(t_i) \setminus \mathcal{A}^{(L,j)}(t_{i-1})$ and $\mathcal{B}^{(L,j)}(t_i)$ are sorted in ascending order according to their distance to the AXVs that belong to the reachable area of the j-th AXV. Then, the j-th AXV communicates to the rest of the system its position estimate and the sorted landmark information up to the maximum communication capacity [in case where there are—estimations of—landmarks that are very close to each other, e.g., their distance is less than a pre-specified threshold, then only one of these landmarks is kept in the

sorted list so as to avoid sending landmarks that contain similar information]. In return, the j-th AXV receives its next way point, the position estimates of the AXVs that belong to its reachable exploration area as well as the sorted landmark information from the other AXVs up to the maximum communication capacity, where the sorting of the landmarks is done with respect to their (x, y)-distance from the j-th AXV (and quantized by removing landmarks that carry similar information). By "landmark information" we mean (i) the index of the landmark in case this landmark belongs to the set $\mathcal{A}^L$ (i.e., in case it is an accurately estimated landmark), its estimated value $(\hat{z}_{L,\ell})$ and a binary signal indicating that it corresponds to an accurately estimated landmark (b) the index of the landmark and the respective sensor measurement in case this landmark is a currently visible but not accurately estimated landmark and a binary signal indicating that it corresponds to a visible but not accurately estimated landmark. Moreover, by "up to the maximum communication capacity" we mean that the packet to be sent or received is filled with as many landmarks as its maximum capacity allows.

By using the above logic for designing the signals to be communicated, useless sensor information is not communicated (i.e., information about landmarks that it not possible for the AXVs to "see" in the next time-step as these landmarks are quite far from the current positions of the AXVs) while sensor information that can and should be used by the AXVs is quantized and prioritized by giving priority to landmarks that are more likely to be "seen" and, moreover, by removing information about landmarks that carry similar information.

We close this section by mentioning that in the simulation and experimental results detailed in later sections, the particular single-SLAM estimator used was a standard non-linear least-squares solver for estimating the landmarks and a cooperative EKF-based scheme for localization.

4 Autonomous multi-robot AXV navigation/exploration as an optimization problem

The distributed SLAM procedure as described in the previous section can be successful only if the trajectories of the AXVs are designed in real-time, so as to optimize the estimators' performance. In other words, a real-time navigation procedure for the multi-AXV system is needed which will optimize the performance of the overall estimation procedure. Apart from that, special emphasis must be given so as the navigation procedure is fault-tolerant, i.e., it is still working in case where one or more AXV has lost connection to the rest of

the team. In the next sections, we describe and analyse the navigation procedure developed within the proposed system.

4.1 Optimal navigation/exploration

By using all the preliminaries and definitions described previously, the optimal AXV team navigation/exploration problem can be cast as a dynamic optimization problem as follows:

$$\max_{\mathbf{X}^R(t_1), \mathbf{X}^R(t_2), \ldots, \mathbf{X}^R(t_N), N} \sum_{i=1}^{N} J(t_i)$$

$$s.t. \quad C(\mathbf{X}^R(t_i)) \leq 0, i = 1, \ldots, N \tag{7}$$

$$J(t_i) = \frac{\left|\mathcal{A}^L(t_i)\right| - \left|\mathcal{A}^L(t_{i-1})\right|}{(t_i - t_{i-1})} \tag{8}$$

where N denotes the time-instant where all landmarks have been accurately estimated, $|\mathcal{A}|$ denotes the *cardinality* of the set $\mathcal{A}$ and $C(\cdot)$ is a non-linear function of the AXV positions. The incorporation of the non-linear function $C(\cdot)$ is used in order to constrain the AXV waypoints $\mathbf{X}^R(t_i))$ to the rectangle $\mathcal{U}$ as well as to incorporate obstacle avoidance and maximum speed constraints. Standard algebraic manipulations can be employed to cast all these constraints in the form $C(\mathbf{X}^R(t_i)) \leq 0$. Note that the instantaneous cost $J(t_i)$ corresponds to the rate of number of landmarks that are accurately estimated per time unit. It is not difficult for someone to see that maximizing the criterion $\sum_{i=1}^{N} J(t_i)$ is equivalent of minimizing the time to accomplish the map construction procedure. It is worth noticing that the above formulation is not unique and other, different formulations can be applied as well. All the results of the proposed approach can be easily extended to the case where different formulations are used than the one above [by simply replacing the term $J(t_i)$ in Eq. 7 by the respective term of the different formulation]. Also, note that the accurate estimation of the AXV positions (localization) is not directly used in the above formulation. However, accurate estimation of the AXV positions is a prerequisite for accurate map construction and, as a result, the optimal solution of (7) requires that optimal waypoints $\mathbf{X}^R(t_1), \mathbf{X}^R(t_2), \ldots, \mathbf{X}^R(t_N)$ are chosen so that AXVs are optimally localized whenever it is necessary.

4.2 Optimal one-step-ahead navigation/exploration

There are many different reasons that render the solution of the dynamic optimization problem (7–8) practically infeasible. First of all, the complexity of the overall system dynamics renders practically very difficult to obtain closed-form (analytic) expressions for most of the terms in (7–8) which is a prerequisite, for applying most of the available tools from optimization theory. Most importantly and even if

it were possible to obtain closed-form expressions, still the problem (7–8) would be impossible to be practically solved as it is a NP-complete problem. To overcome these two problems we adopt a two step approach:

Step 1 In the first step, we assume that it is possible to obtain closed-form expressions for the terms in (7–8) and then attempt to modify the overall problem so as to come up with an approach that overcomes the NP-complete issue, on the one hand, but provides an efficient solution to the navigation/exploration problem, on the other. To do so, we adopt an approach where the problem is to construct—based on the dynamic optimization problem (7–8)—an appropriate objective function $\mathcal{J}(t_i)$—different than $J(t_i)$ in Eq. (7)—so that *optimizing one-step-ahead* the function $\mathcal{J}(t_i)$ leads to an efficient solution to the navigation/exploration problem. In other words, we attempt to construct a function $\mathcal{J}(t_i)$ such that, choosing the waypoints $\mathbf{X}^R(t_{i+1})$ that optimize $\mathcal{J}(t_{i+1})$, provides an efficient solution to navigation/exploration problem. Both theoretical analysis and simulation experiments are used towards such a purpose. In the simulation experiments, to overcome the problem that closed-form expressions are not available, we employ the *optimal-one-step-ahead random semi-exhaustive search (OSARSES)*, an approach which approximates the optimal-one-step-ahead exhaustive search algorithm. Please note that OSARSES is a very "heavy" computational algorithm which cannot be implemented in real-life applications and is used only for analysis purposes.

Step 2 In order to overcome the problem that closed-form solutions of the terms involved in the optimization problem, we employ the Cognitive Adaptive Optimization (CAO) algorithm initially introduced in Kosmatopoulos et al. (2007), Kosmatopoulos (2009), and Kosmatopoulos and Kouvelas (2009). CAO has successfully been implemented to a similar—but significantly less complex—problem, this of navigating a team of autonomous robots when they are deployed to perform optimal surveillance coverage (Renzaglia et al. 2012; Doitsidis et al. 2012; Renzaglia et al. 2013). As in the case of navigation/exploration for map construction, the problem of optimal surveillance coverage involves optimization of terms for which closed-form expressions are not available. However, the problem of optimal surveillance coverage is a *static optimization* problem and, thus, less complex than the dynamic optimization problem of exploration treated here.

4.3 Transforming the optimization problem

An iterative approach was used in order to accomplish the first step described above:

Candidates for the objective function $\mathcal{J}$ were evaluated by employing multi-AXV exploration using OSARSES for two different simulation environments. These two different simulation environments used two different quite complex maps (referred hereafter **Map #1** and **Map #2**). The first map (Map #1) depicts an area located in Zurich, Switzerland. This map was generated using a state-of-the-art visual-SLAM algorithm (Doitsidis et al. 2012) which tracks the pose of the camera while, simultaneously and autonomously, building an incremental map of the surrounding environment. The second map (Map #2) is an artificially generated one, constructed using S-plines interpolation in such a way to represent sharp morphological variations in order to be used as worst case mapping scenario. Both maps are illustrated in Fig. 1

For each of the candidate forms of $\mathcal{J}$ and for both of the simulation environments, the AXVs were navigated so as to optimize $\mathcal{J}$ by employing OSARSES: at each time-instant t_i, different sets of feasible candidate next waypoints $\mathbf{X}^{R,cand}(t_{i+1})$ for the AXVs were randomly generated and the overall system was simulated until the time-instant t_{i+1}; several such candidate sets were generated/simulated and the best [i.e., the one that provides the best $\mathcal{J}(t_{i+1})$] is chosen to be the next waypoints for the AXVs. The overall procedure of applying OSARSES is exhibited in Algorithm 1.

Algorithm 1 One time-stamp of OSARSES Algorithm

Randomly Generate $\mathbf{X}^{R,cand}(t_{i+1})$
Exclude the ones that violates the environmental and communication constraints $\mathbf{X}^{R,cand}_{valid}(t_{i+1}) \subseteq \mathbf{X}^{R,cand}(t_{i+1})$
$c \leftarrow 0$
while $c \leq |cand|$ **do**
  Deploy the action $\mathbf{X}^{R,cand(c)}_{valid}(t_{i+1})$
  $\mathcal{J}_c \leftarrow \mathcal{J}(t_{i+1})$ //Calculate the objective function value for this valid movement
  $c \leftarrow c + 1$
end while
$imax \leftarrow \arg\max(\mathcal{J})$
$\mathbf{X}^{R,final}(t_{i+1}) \leftarrow \mathbf{X}^{R,cand(imax)}_{valid}(t_{i+1})$

Using the above described procedure, an iterative design procedure was adopted for the design of $\mathcal{J}$: at each step of this iterative procedure, the function $\mathcal{J}$ was modified, based on both observations on the system simulated performance and theoretical analysis.

In order to evaluate the performance under different choices for $\mathcal{J}$ we performed two different types of comparison. First, we compared the performance among these different choices as explained in the previous subsection. Secondly, we used an approximate solution to the dynamic

Fig. 1 The simulation environment. **a** Map #1. **b** Map #2

(a)

(b)
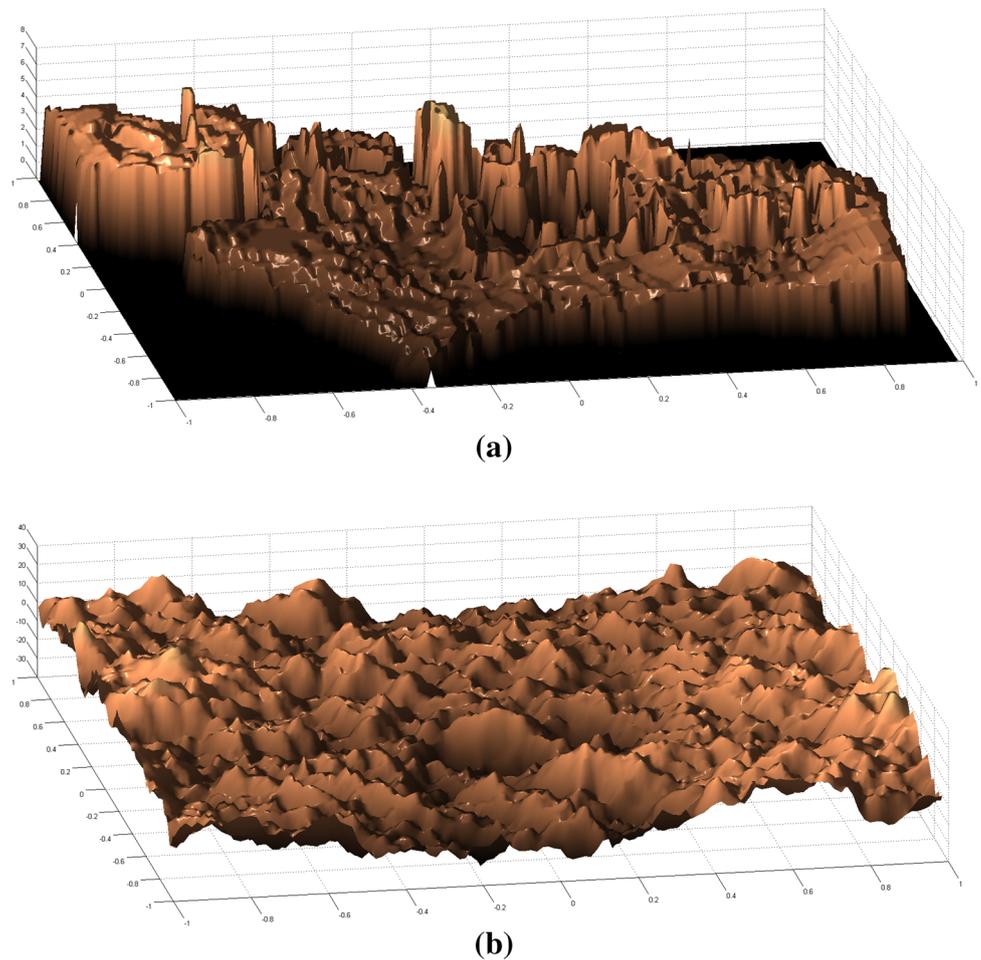
optimization problem (7–8) and compared its performance with that of the proposed approach. More precisely, by employing the so-called The parametrized cognitive adaptive optmization (PCAO)— approach was employed in order to approximately solve the dynamic optimization problem (7–8). By doing so, we concluded that the globally optimal—but practically not feasible— navigation/exploration algorithm for Map #1 can accomplish the overall map construction mission in less than 400 time-units, while the globally optimal navigation/exploration algorithm for Map #2 can accomplish the overall map construction mission in less than 450 time-units. It is worth noticing that the PCAO approach can provide a non-practically feasible solution as the number of computations required for implementing such a solution is huge. Moreover, as the exact solution to the dynamic optimization problem (7–8) is not possible to be obtained, these performance numbers correspond to upper bounds for the performance of the globally optimal solution.

However, as PCAO did not exhibit and any further significant improvement on these numbers as its approximation accuracy is increased, these upper bounds seem to be quite close to the performance bounds of the globally optimal

performance. Having this in mind, all of the simulation experiments for both Map #1 and Map #2 were executed for 500 time-units and the performance index we adopted for evaluating the different choices for $\mathcal{J}$ as well as the proposed CAO-based approach was the *map error at t=500* (where by "map error" we define the percentage of landmarks that have not been accurately estimated). A solution that provides a small map error at $t = 500$, although it is not the globally optimal, is very close to it.

Below, we describe the iterative procedure used for devising the objective function $\mathcal{J}$. Initially, the function $\mathcal{J}(t_i)$ was selected to be the instantaneous cost $J(t_{i+1})$ for the dynamic optimization problem (7–8). This is a typical choice in many different practical applications and it is the most straightforward choice for $\mathcal{J}$. However, the use of such a choice exhibits a very poor performance as shown in Table 1. More precisely and as shown in Table 1, this particular choice for $\mathcal{J}$ leads to a very poor performance (less than 15 %[4] of the total number of landmarks are accurately estimated, for all different cases).

---

[4] Table 1 presents the performance using the average percentage of the *Non-Accurately* estimated landmarks, so as to be in-line with the upcoming results.

Table 1 Average percentage of non-accurately estimated landmarks at $t = 500$ for $\mathcal{J}(t_i) = J(t_{i+1})$

Landmarks	1100			All	
Candidates	5	50	500	5	70
Map #1 (%)	91.9	88.9	85.3	84.9	85
Map #2 (%)	87.5	89.6	89.8	87.8	86.3

The poor performance of the choice $\mathcal{J}(t_i) = J(t_{i+1})$ is due to the fact that such a choice leads the overall algorithm to get stuck in *deadlocks* (or, mathematically speaking, the algorithm gets stuck in *local maxima*). To avoid such an unexpected situation, we modify the $\mathcal{J}(t_i) = J(t_{i+1})$ by adding more terms, the maximization of which lead to a better performance. By employing the above-described iterative approach, the results of which are presented in the following subsections, the final form of the function $\mathcal{J}(t_i)$ is shown below:

$$\mathcal{J}(t_i) = J(t_{i+1}) + J_1(t_{i+1}) + J_2(t_{i+1}) + J_3(t_{i+1}) \quad (9)$$

This enhanced version of the objective function retains only the terms that actually attribute to the general objective of the problem (see Sect. 4.3.4 and Fig. 2). Below, we describe each of the terms in (9), as well as the reasoning behind choosing these terms.

4.3.1 Move towards the closest unexplored areas

The choice for the term J_1 is based on the observation that when there are no further landmarks that can be accurately estimated by some AXVs, the objective function can be augmented so as to "motivate" these AXVs to increase the number of *visible but non-accurately estimated* landmarks. The idea behind such an augmentation is simple: whenever some AXVs cannot estimate further landmarks, then they are "moved" to the closest unexplored areas. The particular form for J_1 that realizes such a reasoning is as follows:

$$J_1(t_i) = \sum_{j=1}^{N_R} s_{1,j}(t_i) \frac{\left| \mathcal{B}^{L,j}(t_i) \right|}{t_i - t_{i-1}} \quad (10)$$

where $\mathcal{B}^{L,j}(t_i)$ denotes the set of indices of visible, but non-accurately estimated landmarks by the j-th AXV—see Definition 3—and the term $s_{1,j}$ denotes a switching function that is zero when there are no deadlocks and becomes equal to a user-defined parameter K_1, otherwise:

$$s_{1,j}(t_i) = \begin{cases} 0 & \text{if } \left| \mathcal{A}^{L,j}(t_{i-1}) \right| - \left| \mathcal{A}^{L,j}(t_{i-2}) \right| \leq \epsilon_1 \\ K_1 & \text{otherwise} \end{cases} \quad (11)$$
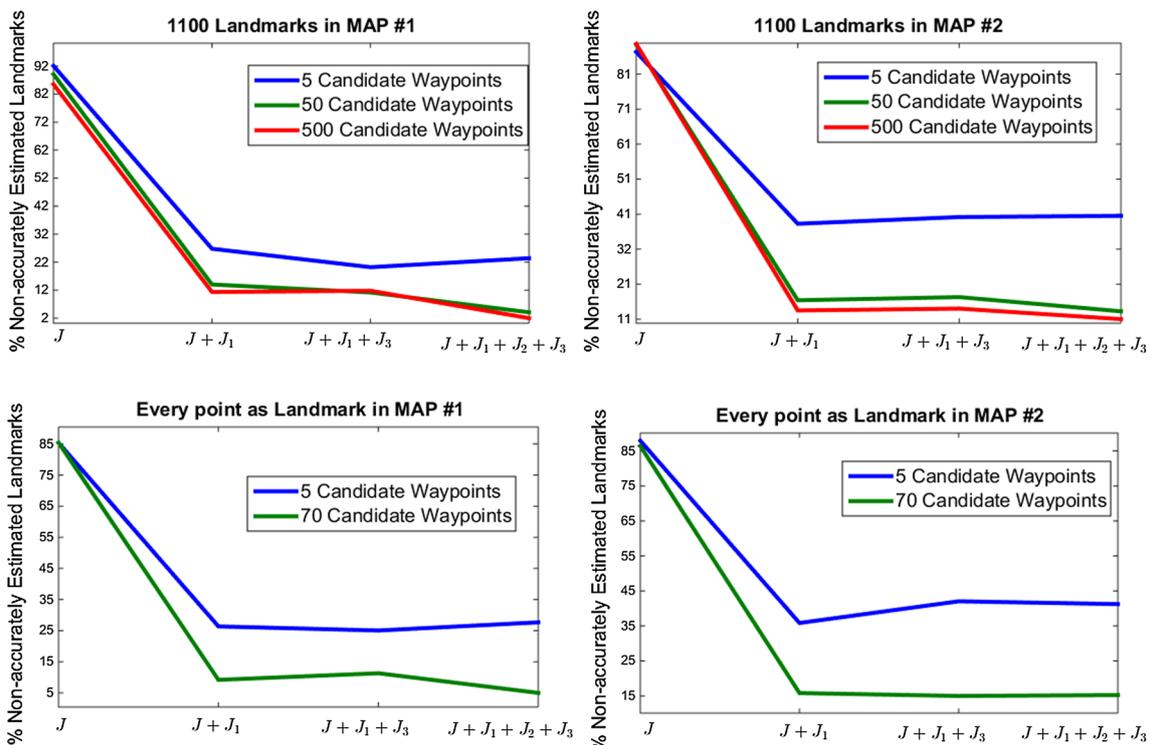
Fig. 2 Comparison of the average percentage of non-accurately estimated Landmarks on two different maps and two different sets of landmarks

where ϵ_1 is a small positive design constant. As it can be seen in Fig. 2, the augmentation of the objective function with the term J_1 leads to dramatic improvements to the navigation/exploration task.

4.3.2 Avoid poor estimation

Additionally to the augmentation using the term J_1, a further augmentation is used that takes into account the distances between every visible, non-accurately estimated landmark and its closest AXV. More precisely, the objective function is augmented with the term J_2 which is used to force the AXVs to come closer to landmarks that are currently visible but not yet estimated, in order to complete their estimation process. Such a term takes into account the non-linear effect of the sensor noise [cf. Eq. 4]: as the effect of sensor noise depends on the distance between the AXV and the landmark, the estimation of the landmark becomes "better" when the AXV moves closer to the landmark. Such an observation is realized by including the term J_2 defined as follows:

$$J_2(t_i) = - \sum_{j=1}^{N_R} s_{2,j}(t_i) \frac{\sum_{j \in B^L(t_i)} \min_{i=1,\dots,N_R} y^2_{x_i^R - \mathbf{X}_j^L}}{t_i - t_{i-1}} \quad (12)$$

where $s_{2,j}$ is a switching function defined similarly to $s_{1,j}$:

$$s_{2,j}(t_i) = \begin{cases} 0 & \text{if } \left| \mathcal{A}^{L,j}(t_{i-1}) \right| - \left| \mathcal{A}^{L,j}(t_{i-2}) \right| \le \epsilon_1 \\ K_2 & \text{otherwise} \end{cases} \quad (13)$$

where K_2 is a positive design constant. In other words, the term J_1 is responsible for moving the AXVs closer to the landmarks that "have not seen before" (or "have been poorly seen"), while the term J_2 is responsible for moving the AXVs closer to landmarks so that they reduce the sensor noise effect and they can "see them better". Due to the trade off between the two terms, the constant K_1 and K_2 are used that serve as weights for giving less or more priority to one of the terms J_1, J_2.

4.3.3 The "curse" of full knowledge of a local region

After performing several experiments by using the cost criterion $\mathcal{J}(t_i) = J(t_{i+1}) + J_1(t_{i+1}) + J_2(t_{i+1})$, it was observed that the use of such a criterion possessed the disadvantage that, in quite a few of instances, the AXVs get trapped in a sub-region that has been fully explored and there are no further non-accurately estimated landmarks that are or can become visible. As a result, any possible movement of the AXVs within the sub-region does not cause any change in the cost function. Consequently, the AXVs are "trapped" in a dead-lock. To avoid this undesirable situation, the term J_3

is introduced. Ideally, this term would attempt to minimize the distance between every AXV and its closest landmark that is currently invisible and not accurately-estimated, so as to force the AXVs to move closer to landmarks that are non-visible and not-accurately estimated. As the calculation of this distance can not be practically performed (as the exact position of this landmark is not known), firstly, the currently-invisible, non accurately-estimated landmarks *that have been visible at some time in the past* are examined. If no such a landmark exists (i.e., all of the landmarks that were visible at some point, were accurately estimated), then the distances between the AXVs and the estimates of landmark positions is taking into account. Evidently, the term J_3 aims to move the AXVs in the "borders" between known and unknown areas and is defined as follows:

$$J_3(t_i) = - \sum_{j=1}^{N_R} s_{3,j}(t_i) \frac{\sum_{j \in \hat{\mathcal{B}}^L(t_i)} \min_{i=1,\dots,N_R} y^2_{x_i^R - \mathbf{X}_j^L}}{t_i - t_{i-1}} \quad (14)$$

The set $\hat{\mathcal{B}}^L(t_i)$ denotes the set of indices of landmarks that were visible in the past but have not been accurately estimated and, if no such landmarks exist, the set $\hat{\mathcal{B}}^L(t_i)$ denotes the set of all landmark estimates that are or have not been visible and accurately estimated. The switching function $s_{3,j}$ becomes equal to 1 only when none of the terms J, J_1, J_2 can lead to a further improvement due to the j-th AXV, i.e.,

$$s_3(t_i) = \begin{cases} 0 & \text{if (Condition } <>) \\ K_3 & \text{otherwise} \end{cases} \quad (15)$$

Condition $<> = \left| A^{L,j}(t_{i-1}) \right| - \left| A^{L,j}(t_{i-2}) \right| \le \epsilon_1$
and
$|J_{1,i}(t_{i-1}) - J_{1,j}(t_{i-2})| \le \epsilon_2$
and
$|J_{2,j}(t_{i-1}) - J_{2,j}(t_{i-2})| \le \epsilon_3$

4.3.4 Performance of the final form of the objective function

Figure 2 highlights the impact of each term of the objective function. It's worth noting that the simulation setup was kept the same for all the experiments, as explained in Sect. 4.3. A conclusion that arises through the study of the results is the observation that the second term (see "Move towards the closest unexplored areas") strongly improves the overall performance. The rest of the terms are employed to guarantee the overall system's robustness, making it less vulnerable to the system's uncertain parameters such as the terrain's morphology and/or the initial arrangement of the landmark estimates. It is remarkable that in certain experiments, e.g. the one with 1100 landmarks on Map #1 with 500 candidate vectors for

OSARSES, the exploration process managed to accurately estimate the 98.2 % of the required landmarks, obtaining an improvement of 700 % as compared to the choice $\mathcal{J} = J$. Summarizing, *the non-practically feasible method OSARSES under the choice* (9) *for the objective function can provide with quite efficient solutions under the condition that the number of candidate vectors for OSARSES is sufficiently large.*

4.3.5 Theoretical analysis of $\mathcal{J}$

Apart from the numerical results that exhibit the efficient performance under the choice (9), it can be seen that such a choice leads to *deadlock-free* solutions and, moreover, the solutions obtained by one-step-ahead maximization this function are efficient. Such a claim is formally stated in the next Theorem.

Theorem 1 *The objective function $\mathcal{J}$ as defined in (9) attains one and only maximum (i.e., it is free of local maxima). More precisely, its global maximum is attained when all the landmarks have been accurately estimated (i.e., when the overall map construction mission is successfully accomplished). Moreover, one-step-ahead maximazation of this function guarantees efficient performance of the overall navigation/exploration mission. More precisely, let us consider the three different sub-teams of the overall AXV team:*

- *the sub-team (A) which includes all AXVs of the team for which there are landmarks within their vicinity that can be accurately estimated;*
- *the sub-team (V) which includes all AXVs of the team for which there are landmarks within their vicinity that cannot become accurately estimated but they can become visible;*
- *the sub-team (I) which includes all AXVs of the team for which there are neither landmarks within their vicinity that can become accurately estimated nor landmarks that can become visible.*

Then, one-step-ahead maximization of $\mathcal{J}$ implies that (a) the sub-team (A) will be cooperatively navigated so as to maximize the number of landmarks to become estimated; (b) the sub-team (V) will be cooperatively navigated so as to maximize the estimation accuracy of the landmarks to become accurately estimated; (c) the sub-team (I) will be cooperatively navigated to the closest to each AXV non-explored areas.

The proof of the above Theorem is straightforward as the functions J, J_1, J_2, J_3 were designed so as to satisfy the performance mentioned in the Theorem.

5 Cognitive adaptive optimization for autonomous multi-robot AXV navigation/exploration

Apart from the theoretical contribution about the construction of an objective function, the approximation of which will be sufficient enough to navigate a team of AXVs in order to accurately map an unknown environment, our advantage against the majority of one-step-ahead optimal algorithms is that we utilize a more realistic approach where the actual evaluation of the objective function is not available at each timestep, only an approximation of it based on historical values. In essence, the actual objective/reward function at each timestep is highly depended on the morphology of the unknown (to be mapped) area and its evaluation without the actual movement should not considered trivial.

5.1 Preliminaries: problem conceptualization

Having defined the active exploration criterion, we will now proceed on presenting the proposed algorithm for autonomously navigating the AXVs towards maximizing such a criterion. The algorithm to be used is based on the so called Cognitive-based Adaptive Optimization (CAO) approach originated in the references Kosmatopoulos et al. (2007), Kosmatopoulos (2009), and Kosmatopoulos and Kouvelas (2009). CAO has been used in the past in a variety of robotics related applications, including implementations in aerial and ground robots, as described in details in Renzaglia et al. (2012), Doitsidis et al. (2012), and Amanatiadis et al. (2013). The version of the CAO algorithm used within the proposed approach, takes the same form and extends the one presented and formaly analysed in Renzaglia et al. (2012).

Below, we provide the main details of the CAO algorithm as employed in the framework of the active exploration problem. Please note that the only difference between the CAO approach used for the multi-robot optimal surveillance coverage problem (Renzaglia et al. 2012; Doitsidis et al. 2012) and the one used here lies in the use of a different optimization criterion which, in turn, leads to different performance metrics (as detailed in Theorem 2 below).

We start by noticing that the active exploration criterion (Eq. 9) is a function of the AXVs positions, i.e.,

$$J_{t_i} = \mathcal{J}\left(\mathbf{X}_{t_i}^R\right) \tag{16}$$

where $t_1, t_2, t_3, \ldots$ denotes the time-index, J_{t_i} denotes the value of the active exploration criterion at the t_i-th time-step, $\mathbf{X}_{t_i}^R$ denote the position vectors of the AXVs (see Sect. 3.2), and $\mathcal{J}$ is a non-linear function which depends—apart from the AXVs positions – on the particular environment where the AXVs operate (e.g., position of landmarks).

Due to the dependence of the function $\mathcal{J}$ on the particular environment characteristics, the *explicit form of the function $\mathcal{J}$ is not known* in practical situations; as a result, standard optimization algorithms (e.g., steepest descent) are not applicable to the problem in hand. However, in most practical cases, like the one treated in this paper, the current value of the active exploration criterion can be estimated from the AXVs sensor measurements. In other words, at each time-step t_i, an estimate of J_{t_i} is available through AXVs sensor measurements,

$$J_{t_i}^n = \mathcal{J}\left(\mathbf{X}_{t_i}^R\right) + \xi_{t_i} \tag{17}$$

where $J_{t_i}^n$ denotes the estimate of J_{t_i} and ξ_{t_i} denotes the noise introduced in the estimation of J_{t_i} due to the presence of noise in the AXVs sensors. Please note that, although it is natural to assume that the noise sequence ξ_{t_i} is a stochastic *zero-mean* signal, it is not realistic to assume that it satisfies the typical Additive White Noise Gaussian (AWNG) property even if the AXVs sensor noise is AWNG: as $\mathcal{J}$ is a non-linear function of the AXVs positions (and thus of the AXVs sensor measurements), the AWNG property is typically lost.

Apart from the problem of dealing with a criterion for which an explicit form is not known but only its noisy measurements are available at each time, efficient AXV navigation algorithms have additionally to deal with the problem of restricting the AXVs positions so that obstacle avoidance and communication constraints are met. In other words, at each time-instant t_i, the vector $\mathbf{X}_{t_i}^R$ should satisfy a set of constraints which, in general, can be represented as follows:

$$\mathcal{C}\left(\mathbf{X}_{t_i}^R\right) \leq 0 \tag{18}$$

where $\mathcal{C}$ is a set of non-linear functions of the AXVs positions. As in the case of $\mathcal{J}$, the function $\mathcal{C}$ depends on the particular environment characteristics (e.g., location of obstacles, terrain morphology) and an explicit form of this function may be not known in many practical situations; however, it is natural to assume that the active exploration algorithm is provided with information whether a particular selection of AXVs positions satisfies or violates the set of constraints (18).

Given the mathematical description presented above, the active exploration problem can be mathematically described as the problem of moving $\mathbf{X}_{t_i}^R$ to a set of positions that solves the following constrained optimization problem:

$$\begin{aligned} &\text{maximize } J_{t_i} \\ &\text{subject to } \mathcal{C}\left(\mathbf{X}_{t_i}^R\right) \leq 0 \, . \end{aligned} \tag{19}$$

As already noticed, the difficulty in solving, in real-time and in real-life situations, the constrained optimization problem (19) lies in the fact that explicit forms for the functions $\mathcal{J}$

and $\mathcal{C}$ are not available. To circumvent this difficulty, the CAO approach of Kosmatopoulos (2009), appropriately modified the original CAO algorithm so as to be applicable to the problem in hand.

5.2 Main steps of CAO approach

Algorithm 2 One time-stamp of Cognitive Adaptive Optimization Algorithm

---

$J_{t(i)}^n - \mathbf{X}_{t(i)}^R$ //update look-up/history tables with the previously evaluated pair

$\vartheta_{t_{i+1}} = \underset{\vartheta}{\arg\min} \frac{1}{2} \sum_{\ell=\ell_{t_i}}^{t_i} \left(J_\ell^n - \vartheta^\tau \phi\left(\mathbf{X}_\ell^R\right)\right)^2$ //Recalibrate the estimator's characteristics

Randomly Generate $\mathbf{X}^{R,cand}(t_{i+1})$

Exclude the ones that violates the environmental and communication constraints $\mathbf{X}_{valid}^{R,cand}(t_i) \subseteq \mathbf{X}^{R,cand}(t_{i+1})$

$c \leftarrow 0$

while $c \leq |cand|$ **do**

$\quad \hat{\mathcal{J}}_c \leftarrow \vartheta_{t_{i+1}}^\tau \phi\left(\mathbf{X}_{valid}^{R,cand(c)}(t_{i+1})\right)$ //Approximate the objective function value for every valid movement

$\quad c \leftarrow c + 1$

end while

$imax \leftarrow \arg\max(\hat{\mathcal{J}})$

$\mathbf{X}^{R,final}(t_{i+1}) \leftarrow \mathbf{X}_{valid}^{R,cand(imax)}(t_{i+1})$

As a first step, the CAO approach, which its pseudo-code is illustrated in Algorithm 2, makes use of function approximators for the estimation of the unknown objective function $\mathcal{J}$ at each time-instant k according to

$$\hat{j}_{t_i}\left(\mathbf{X}_{t_i}^R\right) = \vartheta_{t_i}^\tau \phi\left(\mathbf{X}_{t_i}^R\right) . \tag{20}$$

Here $\hat{j}_{t_i}\left(\mathbf{X}_{t_i}^R\right)$ denotes the approximation/ estimation of $\mathcal{J}$ generated at the t_i-th time-step, ϕ denotes the non-linear vector of L regressor terms, ϑ_{t_i} denotes the vector of *parameter estimates* calculated at the t_i-th time-instant and L is a positive user-defined integer denoting the *size* of the function approximator (20). The vector ϕ of regressor terms must be chosen so that it is a universal approximator, such as polynomial approximators, radial basis functions, kernel-based approximators, etc.

The parameter estimation vector ϑ_{t_i} is calculated according to

$$\vartheta_{t_i} = \underset{\vartheta}{\arg\min} \frac{1}{2} \sum_{\ell=\ell_{t_i}}^{t_i - 1} \left(J_\ell^n - \vartheta^\tau \phi\left(\mathbf{X}_\ell^R\right)\right)^2 \tag{21}$$

where $\ell_{t_i} = \max\{0, t_i - L - T_h\}$ with T_h being a user-defined nonnegative integer. Standard least-squares optimization algorithms can be used for the solution of (21).

As soon as the estimator (J_ℓ^n is constructed according to (20), (21), the set of new AXVs positions is selected as follows: firstly, a set of N candidate AXVs positions is constructed according to[5]

$$x_{t_i}^{i,j} = x_{t_i}^{(i)} + \alpha_{t_i} \zeta_{t_i}^{i,j}, \quad i \in \{1, \ldots, N_R\}, j \in \{1, \ldots, N\} \quad (22)$$

where $x_{t_i}^{i,j}$ denotes the i-th element of $\mathbf{X}_{t_i}^R$, $\zeta_{t_i}^{i,j}$ is a zero-mean, unity-variance random vector with dimension equal to the dimension of $\mathbf{X}_{t_i}^R$ and α_{t_i} is a positive real sequence which satisfies the conditions:

$$\lim_{i \to \infty} \alpha_{t_i} = 0, \quad \sum_{i=1}^{\infty} \alpha_{t_i} = \infty, \quad \sum_{i=1}^{\infty} \alpha_{t_i}^2 < \infty. \quad (23)$$

Among all N candidate new positions $x_{t_i}^{1,j}, \ldots, x_{t_i}^{N_R,j}$, the ones that correspond to non-feasible positions—i.e., the ones that violate the constraints (18)—*are neglected* and then the new AXVs positions are calculated as follows:

$$\left[\mathbf{X}_{t_i+1}^R\right] = \begin{array}{c} \text{argmax} \\ j \in \{1, \ldots, N\} \\ \mathbf{X}_{t_i}^{R,j} \text{ not neglected} \end{array} \hat{J}_{t_i}\left(\mathbf{X}_{t_i+1}^{R,j}\right)$$

The idea behind the above logic is simple: at each time-instant a set of many candidate new AXVs positions is generated. The candidate, among all feasible ones, that provides the best estimated value $\hat{j}_{t_i}$ of the objective function is selected as the new set of AXVs positions. The random choice for the candidates is essential and crucial for the efficiency of the algorithm, as such a choice guarantees that $\hat{J}_{t_i}$ is a reliable and accurate estimate for the unknown function $\mathcal{J}$; see Kosmatopoulos (2009), and Kosmatopoulos and Kouvelas (2009) for more details. On the other hand, the choice of a slowly decaying sequence α_{t_i}, a typical choice of adaptive gains in stochastic optimization algorithms is essential for filtering out the effects of the noise term ξ_{t_i} [cf. (17)]. The next summarizes the properties of the CAO algorithm described above; the proof is among the same lines as this of Theorem 1 of Renzaglia et al. (2012).

Theorem 2 *Let $\mathbf{X}_{t_i+1}^{R,opt}$ denote the "step-ahead-optimal" AXV waypoints, i.e., the feasible waypoints that maximize $\mathcal{J}(t_i)$. Then, the above-described CAO algorithm satisfies:*

$$\mathbf{X}_{t_i+1}^R = \mathbf{X}_{t_i+1}^{R,opt} + \epsilon(t_i) + \nu$$

---

[5] According to Kosmatopoulos (2009) and Kosmatopoulos and Kouvelas (2009) it suffices to choose N to be any positive integer larger or equal to $2 \times$ [the number of variables being optimized by CAO. In our case the variables optimized are the robot positions $\mathbf{X}_{t_i}^R$ and thus it suffices for N to satisfy $N \geq 2N_R \times \dim\left(\mathbf{X}_{t_i}^R\right)$.

where $\epsilon(t_i)$ vanishes to zero exponentially fast *and the term ν is a constant term that depends on the approximator ϕ (and can become as small as desired at the expense of making the convergence of $\epsilon(t_i)$ slower).*

In simple words, the above Theorem states that the CAO-based approaches become (after some time due to learning) approximately equal to the optimal-step-ahead ones.

6 Simulation results

In this section, we describe the simulation set-up used for the analysis presented in Sect. 4 as well as the evaluation of the proposed CAO-based approach as compared to the practically infeasible OSARSES-based approach. The simulation environment for the experiments is described below:

- Simulations conducted using the two different maps described in detail in 3.4 and presented in 1. For simplification the operation area is restricted in the cube $[-1, 1]^3$, so any value, that is afterwards mentioned, including the maps, has been casted to this cube.
- The number of AXVs is equal to $N_R = 3$ for the first set of experiments (Fig. 5) and $N_R = 10$ for the second (Fig. 6).
- For the communication capacity, we assumed that each AXV can send and receive up to 10 landmark measurements according to the procedure described in Sect. 3.4.
- For the number of landmarks two different scenarios were evaluated. First, we assume that each map consists only of $N_L = 1100$ landmarks. In the second scenario, we assume that every point (pixel) of the height-map is a landmark. In this case, Map #1 includes $N_L = 7542$ landmarks while Map #2 includes $N_L = 10500$ landmarks.
- The main constraints imposed to the AXVs are that they remain within the terrain's limits, i.e., within $[X_{min}, X_{max}]$ and $[Y_{min}, Y_{max}]$ in the x- and y-axes, respectively. At the same time, AXVs remain within $[z + d, z_{Max}]$ along the z-axis, in order to avoid hitting the terrain. The value of d was equal to 0.05.
- The communication range is set to $com_{Range} = 0.3$.
- All AXVs were assumed to have range sensors, measuring the AXV distance from the landmark, using the following equation:

$$y_{x^R-q} = \begin{cases} \text{undefined} & \text{if } \|x^R - q\| \geq \text{thres} \\ \text{undefined} & \text{if there is no line-of-} \\ & \text{sight between } x^R \text{ and } q \\ \|x^R - q\|(1+\xi) & \text{otherwise} \end{cases}$$

$$(24)$$

Fig. 3 Multi-AXV navigation/exploration process: the green area corresponds to currently visible landmarks, the brown area (with morphological characteristics) corresponds to landmarks that have been accurately estimated and the red area corresponds to landmarks that have never been seen before. The three big spheres indicate the communication range of each AXV (located at the center of the spheres). **a** $T = 50$. **b** $T = 200$. **c** $T = 350$. **d** $T = 500$
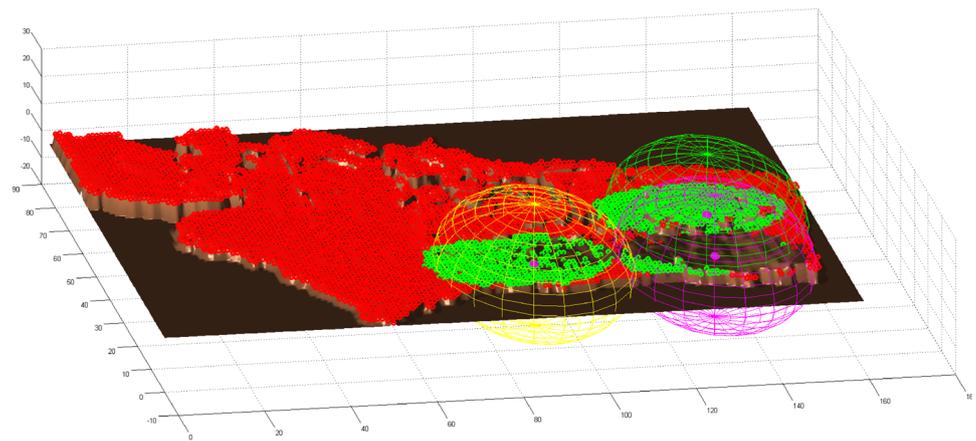
(a)
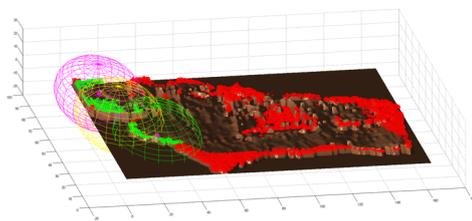
(b)
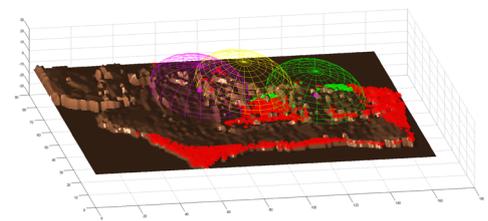
(c)
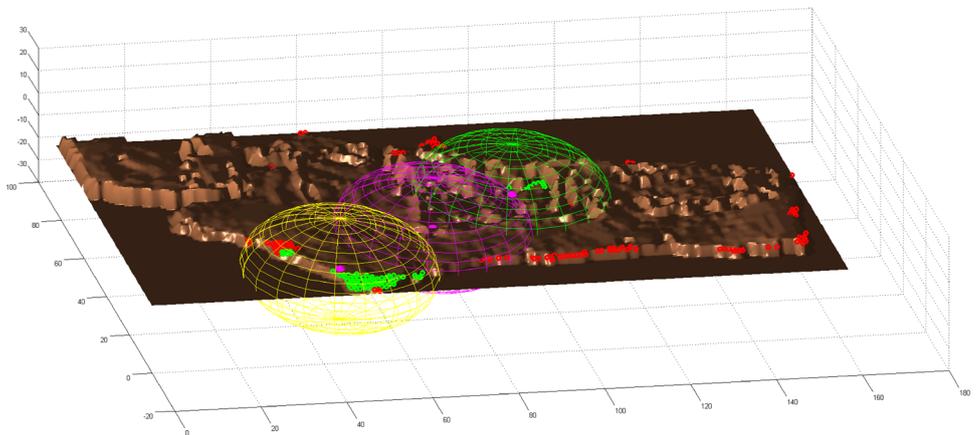
(d)

– Standard weighted least-squares (Ruppert and Wand 1994) was employed for estimated the landmarks, while perfect localization accuracy was assumed.

– As an overall evaluation criterion for the exploration procedure (Figs. 5a,c, 6a,c), the number of the *Remaining Landmarks*, ie the total number of landmarks that are not accurately-estimated after the completion of the experiment, will be used. Additionally to that term, and in order to obtain a better picture about the system's performance, we employed an extra evaluation criterion. The objective of this term is to distinguish between experiments where their final total number of the *Remaining Landmarks* was the same but their estimation progress was

not. This criterion rewards performances that have greater landmarks' estimation ratio over the simulation time, ensuring to reward performances that achieved a satisfactory function from their early steps of the execution. This parameter (Figs. 5b, d, 6b,d) corresponds to the summation of the error in the estimation of total landmarks from the first time-step to the last one, according to the Eq. 25

$$\text{Sum\_of\_Error} = \sum_{i=1}^{T_{max}} ||\hat{\mathbf{X}}^L(t_i) - \mathbf{X}^L(t_i)|| \qquad (25)$$

Fig. 4 The recorded trajectories. **a** The trajectories recorded during the navigation on map #1. **b** The trajectories recorded during the navigation on map #2

(a)

(b)
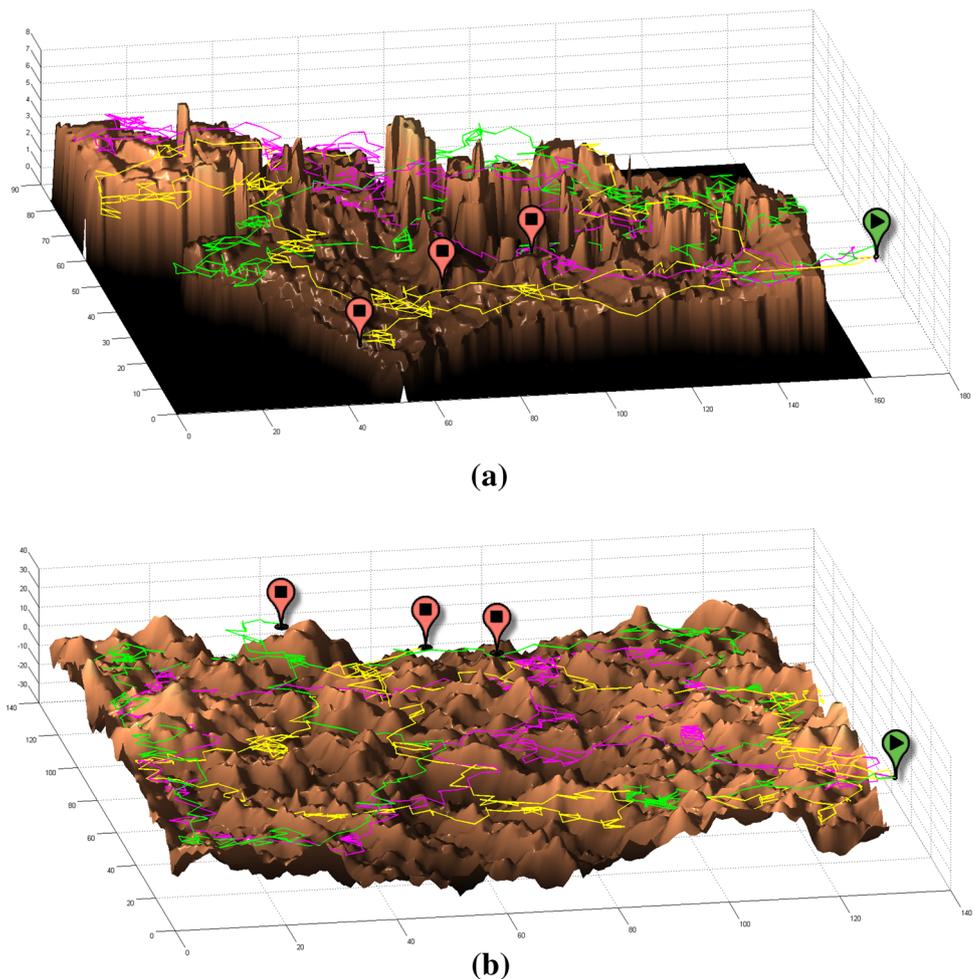
A set of experiments has been conducted in order to evaluate the performance of the proposed approach. One instance of the above experiments, using 3 AXVs, has been illustrated in details, in Fig. 3, where the green area corresponds to currently visible landmarks, the brown area (with morphological characteristics) corresponds to landmarks that have been accurately estimated and the red area corresponds to landmarks that have never been seen before. The three big spheres indicate the communication range of each AXV (located at the center of the spheres). Figure 4 depicts the trajectories of the 3 AXVs in both maps, with their starting and ending positions, as they were calculated by the proposed CAO approach[6].

Overall the results are presented in Fig. 5 for 3 AXVs and Fig. 6 for 10 AXVs. The results from OSARSES are marked with the blue bars while the corresponding results of the CAO-based proposed approach with green ones. In each figure, the x-axis represents the number of real configurations that are evaluated at each timestep from the AXVs, before the final movement selection. It is worth highlighting that, the

proposed approach is only located on the bar that corresponds to 0 real evaluations, as it does not need any actual movement in order to be able to make its decisions (see Sect. 5).

The results indicate clearly that the CAO approach, can obtain better performance even from the OSARSES algorithm which uses five real positions before being able to produce its control decision. Regarding to the 3 AXVs experiment (Fig. 5), an improvement of about 79.7 and 24.1 % in MAP #1 as well as 147.3 and 9 % in MAP #2 map is achieved, compared to the OSARSES with 0 and with 5 candidate set of waypoints ($\mathbf{X}^{R,cand}$) respectively. The study of the summation of error, has further strengthened the conclusions that the proposed approach outperforms the performance of the OSARSES in cases of zero and 5 candidate set of waypoints. Inevitably, in case where the OSARSES uses a large number of candidate set of real waypoints, it performs better than CAO. As it is seen in Fig. 6, in case of ten AXVs with 1100 landmarks, the proposed algorithm can scale up well, retaining its improvements' levels. Concretely, it achieves an improvement of about 80.2 and 71.4 % in each map respectively (Fig. 6a), with a corresponding improvement in the summation of error (Fig. 6b). Interestingly, in the scenario

---

[6] A video footage of this experiment can be found on https://www.youtube.com/watch?v=menK5tMRw-s.

Fig. 5 Conducted experiments for three AUVs, for both the algorithms with 1100 landmarks and *every point as landmark*. **a** % remaining landmarks. **b** Summation of error. **c** % remaining landmarks. **d** Summation of error

(a)

(b)

(c)

(d)
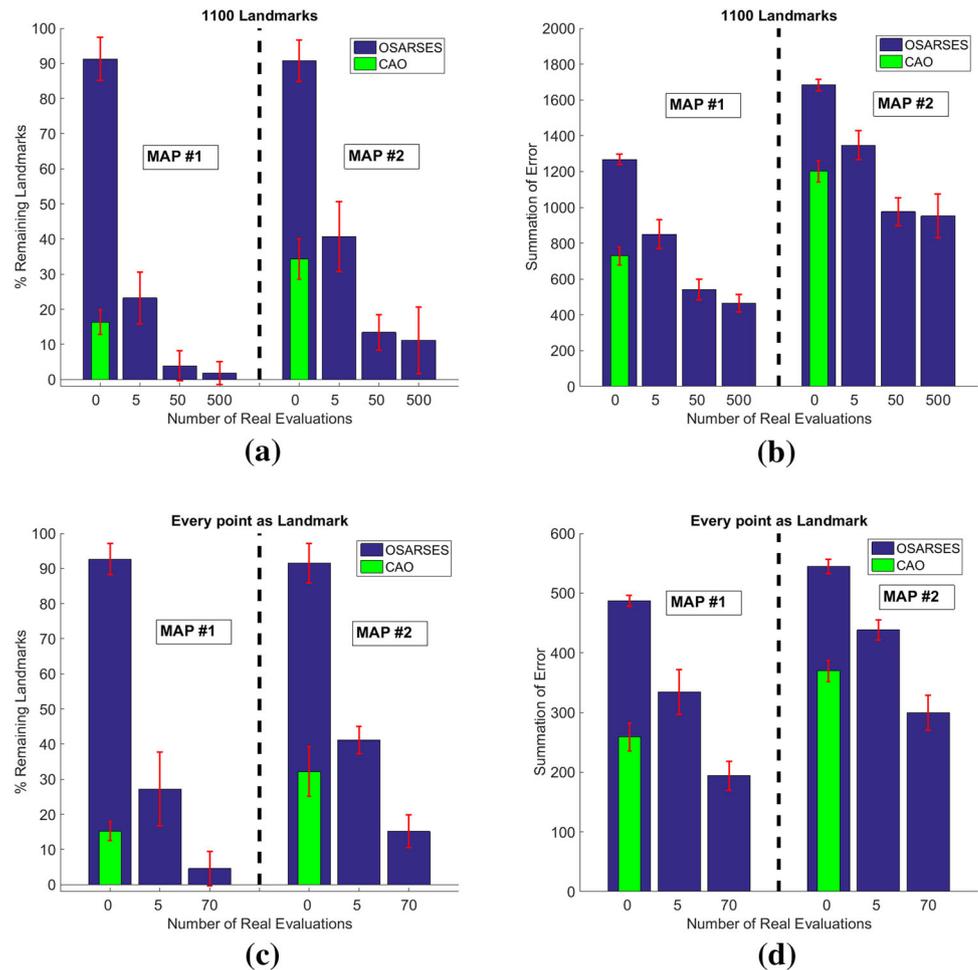
where every point of the map is considered as landmark and as a result a fine-grained mapping is required, the proposed algorithm is able not only to outperform the 0 and 5 real evaluation cases of OSARSES (acquiring the impressive improvement of 88.4 and 63.8 % in Map #1 as well as 72.3 and 40 % in MAP #2 respectively), but also to approximate the performance of OSARSES with 70 real evaluation per timestep.

The aforementioned result is not out of the blue. As the number of AXVs and the landmarks is increasing, the 70 real evaluation became a rather insufficient number for the OSARSES algorithm. Unfortunately, the further increasing is prohibited even in the simulation test-bed and it will take forever to statistically remove the randomness of the results. On the contrary, in the case of CAO approach we are able to efficiently/securely increase the number of candidates, due to the fact that

– these candidates are not actually evaluated, so the operational cost is zero
– even the computational cost to test the candidates on the CAO's estimator (Eq. 20) is extremely inferior compared to the one of OSARSES which has to calculate all the terms of Eq. 9 for every single candidate configuration.

7 Experimental results

The proposed methodology has been also evaluated through real-life tests concerning sea-floor mapping of unknown areas using two autonomous underwater vehicles (AUVs). The tests were conducted in the Leixes Por, located in the city of Oporto, Portugal. Next we describe the details of the experiments.

7.1 System details

A schematic diagram of the system developed for implementing the proposed approach is illustrated in Fig. 7.

The entire procedure can be separated into 2 parts: a web-service one and the client-side part. The web-based interface undertakes the role of coordinating the AUVs. This service requires from the operators minimum amount of information such as the number of the vehicles, so as to begin the procedure. The web-application was designed to provide the operators with a general purpose tool through which the progress of mission is streamlined in real-time. It is worth noticing that the web-service is compatible with the existing systems and can be adopted so as to navigate any type

Fig. 6 Conducted experiments for 10 AUVs, for both the algorithms with 1100 landmarks and *every point as landmark*. **a** % remaining Landmarks. **b** Summation of error. **c** % remaining landmarks. **d** Summation of error
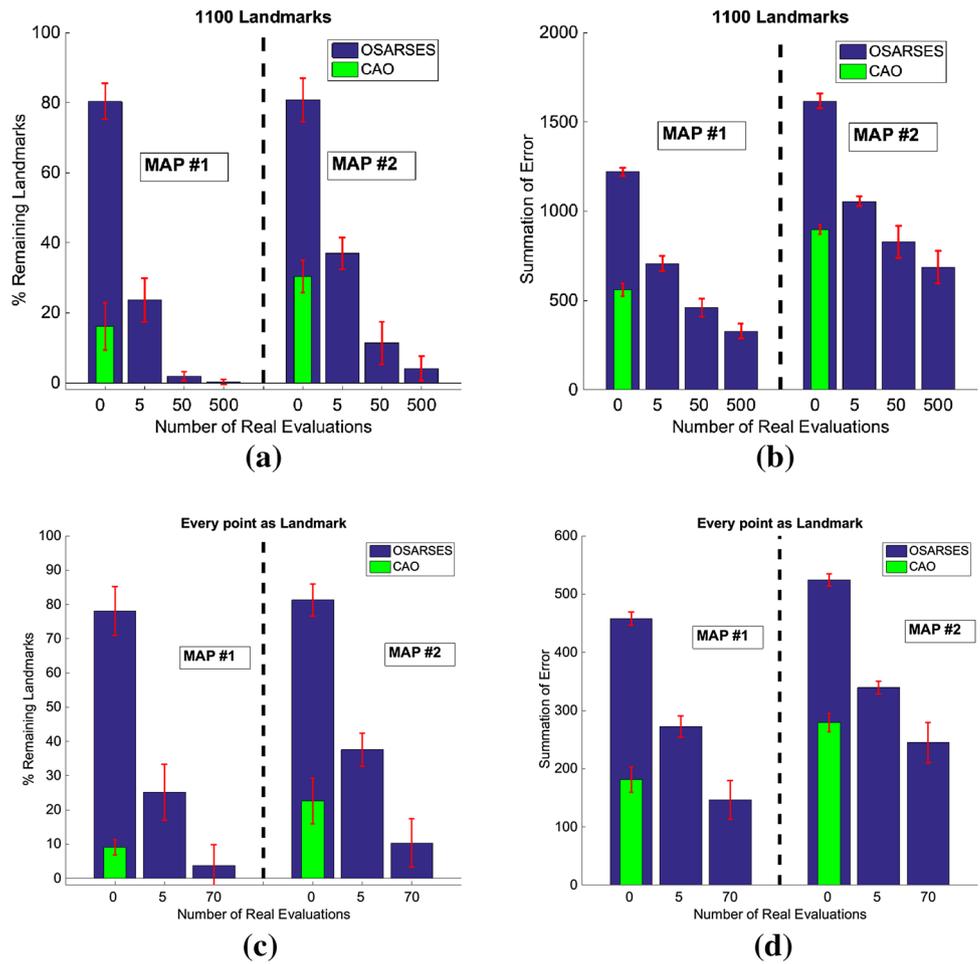
Fig. 7 Flowchart of system used in the experiments
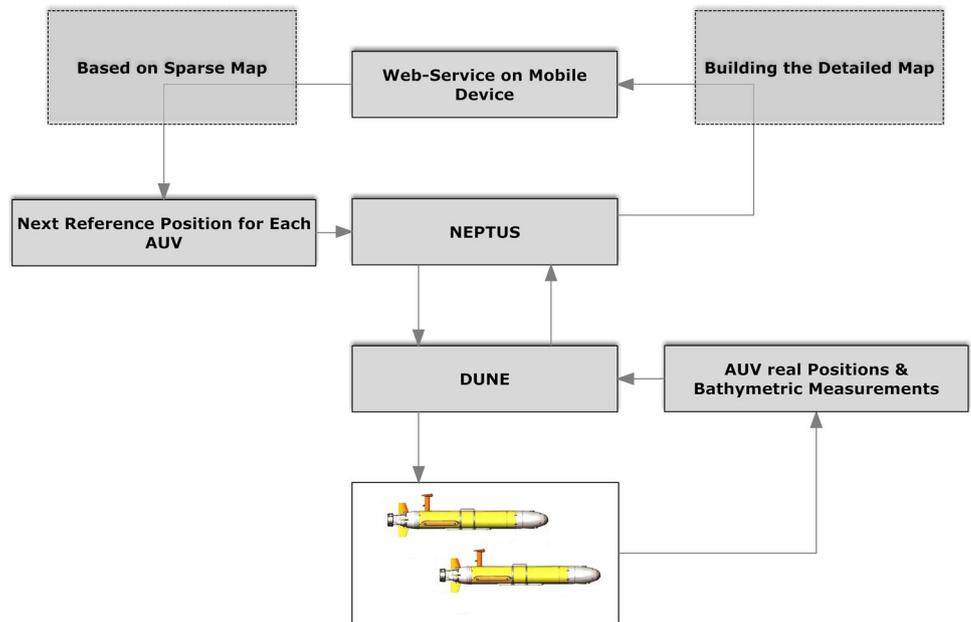
of AUV. The web-service produces in every time step the next optimal position (waypoint) for each AUV, based on proposed approach, the landmarks estimates, and the current position of the AUVs. Next, the new waypoints are transferred to the client-side part of the application.

The client-side part consists of 2 software: NEPTUS [56] and DUNE [57]. NEPTUS is a command and control software that can be used to plan, simulate, monitor and review missions executed by AUVs. DUNE is the runtime environment for vehicle on-board software. It is used to write generic embedded software at the heart of the vehicle, e.g. code for control, navigation, communication, sensor and actuator access, etc. Through these tools, AUVs receive the information about the next waypoint.

When the AUVs reach their respective desired location and stabilize, they activate their exteroceptive sensors. In the sequel, the AUVs communicate their sensor measurements together with their exact positions through NEPTUS and DUNE to the web-service.

Subsequently, the web-service incorporates the actual positions of the AUVs into the proposed model as well as updates the landmarks estimates, based on the sensor information received. A non-linear least squares algorithm is used perform such an estimation (SLAM) procedure. At the same time the web-service, based on the sensor measurements and by a employing a Gaussian based interpolation algorithm, builds the detailed map. It has to be emphasized here that the overall procedure fully relies on each AUV "local" localization system (i.e., the localization system of each AUV does not incorporate measurements from the other AUVs) and, thus, the SLAM system only performs landmark estimation.

7.2 Ground truth: usual practice

Through earlier measurements, observations and calculations, a detailed map of each region we want to capture is available. Hereafter, we will refer to this version of the map as *ground truth*. To obtain a ground truth map, several AUVs have to operate –in a non-cooperative fashion– for many hours using multi-beam sensors, following a predefined iterative procedure, collecting an enormous amount of measurements. Apparently, this is a very time consuming and expensive task. The *ground truth* map, mainly due to the fact that it is considered the best available perception about the morphology of the seabed, constitutes the reference map, the one that is going to be used in order to evaluate, in terms of accuracy, the exploration's results of alternative, feasible methodologies

Simultaneously, several *usual practice* versions of each map are also available. These maps were captured using the today's usual practice for the exploration of unknown underwater environments. According to this approach, the AUVs are following a predefined trajectory collecting data from the seabed using a specific sampling rate, until a predefine time. This methodology despite its simplicity, holds many advantages, that have established its usage in the most real-life exploration/coverage missions:

– predictability (a-priori information about the morphology of the area can be easily incorporated in the global plan by designer).
– it does not require any online communication link -the paths are predefined and the measurements are gathered at the completion of the experiment - between them or ground/vessel station, minimizing the energy consumption.
– fully-coverage is guaranteed.
– The navigation-scheme (straight lines) enhances the localization accuracy.

On the other hand, the *usual practice* has some vital drawbacks that limit its performance:

– The navigation scheme is constant, dealing the same way (number of samples), sub-areas with different height discrepancies or in more abstract terms with different kind of importance.
– For realistic time/energy consuming missions, there is always a risk of completely missing some important subpart of the area, which is located between the AUV's paths.
– The coordination/cooperation of more than one AUV is not trivial and in the most cases does not taking into consideration the initial positions of the AUVs.
– Human intervention is necessary to appropriately define the multi-robots' not-overlapping routes.

It is emphasized that the proposed approach does not use any information from the ground truth map or the usual practice map. Both these two maps (*usual practice* and *ground truth*) are needed for evaluation purposes.

7.3 Experiments in Oporto's harbor

The objective of the experiments is to build a detailed map of two different sub-areas of the Oporto's harbor (Fig. 8a). To accomplish such a mission, in each case, 2 AUVs (Fig. 8c) are deployed equipped with single beam bathymeter sensor. It's worth mentioning that such sensors provide us a small amount of information about the sub-region where the AUVs are deployed, practically only one point.

The experimental environment can be described as follows:

– In both cases the map is a square area with dimensions equal to 100×100 meters.

Fig. 8 The available modules(hardware/software) for real-word experiment. **a** Deploying area, Oporto harbor. **b** Neptus—Command and Control Software. **c** Autonomous underwater vehicle. **d** AUV in operation mode

(a)

(b)

(c)

(d)
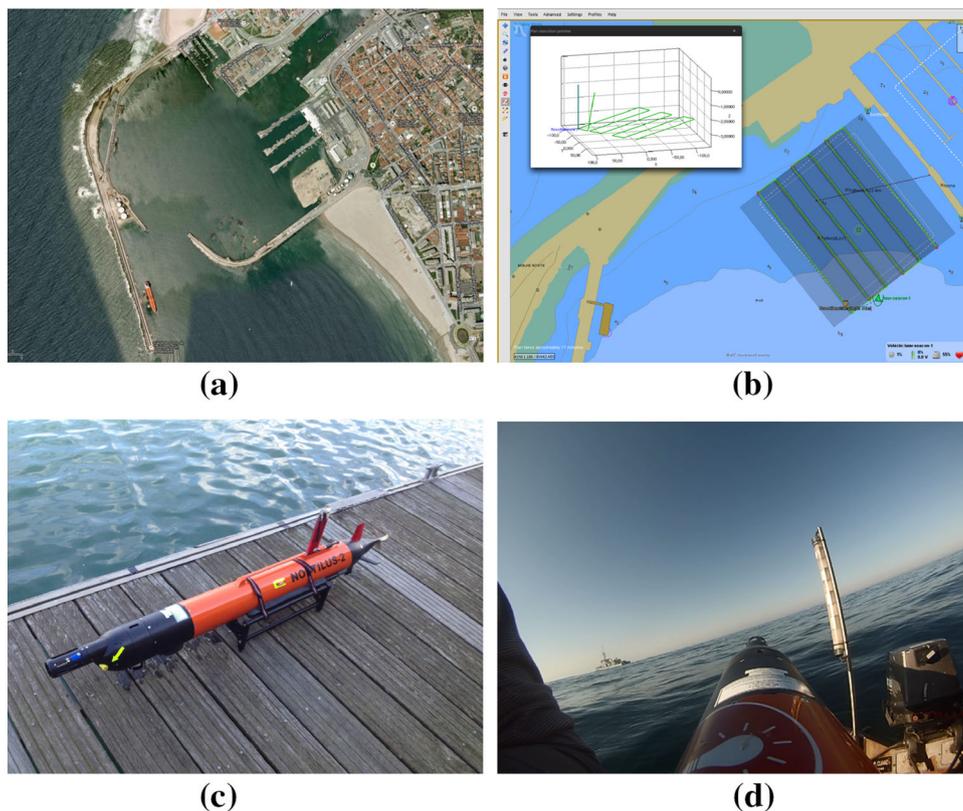
– The number of AUVs is equal to $N_R = 2$.
– The AUVs are moving within the terrain's limits, i.e., within $[X_{min}, X_{max}]$ and $[Y_{min}, Y_{max}]$ in the x- and y-axes, respectively. Each AUV remain in constant z so as to neglect any collision possibility.
– Experiment contacted for $T = 500$ time-steps.

In each time step, the web-service navigates the AUVs to the next desired position through the NEPTUS (Fig. 8b) application and builds the detailed map. Overall, upon completions of the procedure, 500×2 single beam bathymeter sensor measurements are available, in order to form the high detailed version of the observable area. Figures 9 and 10 illustrate the *ground truth*, the *usual practice*, as well as the generated, from the *Proposed CAO-based Approach*, version of the maps for the 2 areas on with we performed the experiments respectively. The first map, will be referred hereafter as #Sharp_Surface while the second one as #Slop_Surface.[7]

In order to evaluate the effectiveness of the proposed approach, we calculate the L^2-$Norm$[8], between the *ground truth* and the *usual practice* map as well as between the

ground truth and the *Proposed Approach* map. Alongside the accuracy of the *Proposed Approach* map, we also evaluate the reliance of the proposed approach on the number of measurements. The results are depicted in Tables 2 and 3. Apart from the estimation-accuracy, expressed in terms of euclidean norm, we display the number of points (measurements) that were used, to present an overview about the total work that is needed in each case.

As one can see, the proposed approach, using only 1000 measurements (86.7 % reduction in the #Sharp_Surface and 77.8 % in the #Slop_Surface), is able to construct a map with more than 32 % accuracy in the first case and 2.1 % in the second, as compared to the today's usual practice.

8 Conclusions

A novel method for dealing the problem of exploring an unknown area using multi-AXV teams under environmental and communication constraints, while simultaneously building a detailed map of the environment has been proposed. Based on this approach we are transforming a standard trajectory generation problem so as to optimize a transformed version of trajectory generation efficiency, employing CAO algorithm. The methodology proposed is independent of requirements regarding operational characteristics of the robots like communication range and type of sensors.

---

[7] Please note that both interpolated versions of *usual practice* present some ridges along the constructed terrain. These ridges correspond to the areas where the AUVs traversed and therefore the samples' concentration is greater than the rest of the terrain.

[8] In order to implement this, at first we discretize, with a sufficient small step, the areas to be compared and afterwords we apply the L^2-$Norm$ on the vectorized versions of the sampled areas.

Fig. 9 *Ground truth*, *usual practice* and produced by the *proposed approach* map employing two AUVs for #Sharp_Surface Map. **a** *Ground truth* Map. **b** *Usual Practice* Map. **c** *proposed approach* Map based on 1000 samples
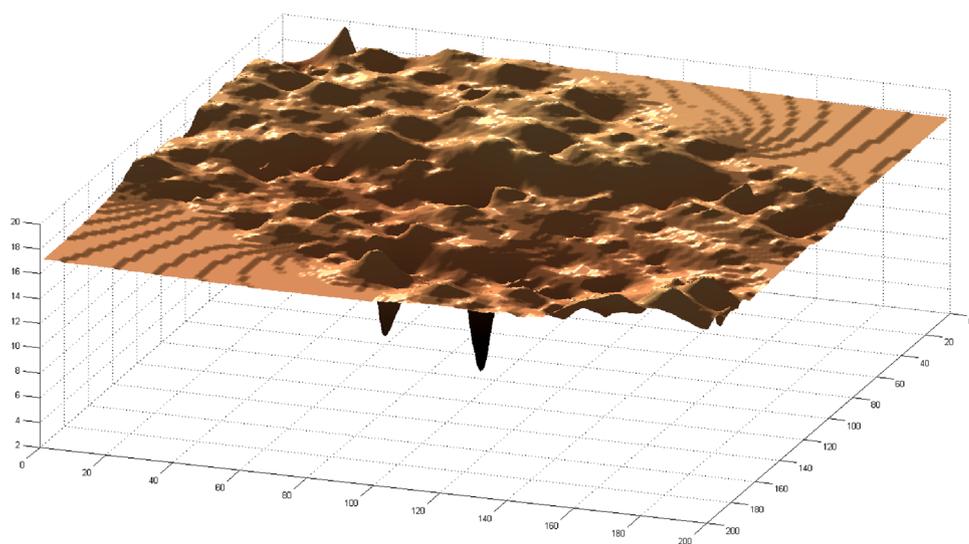
(a)
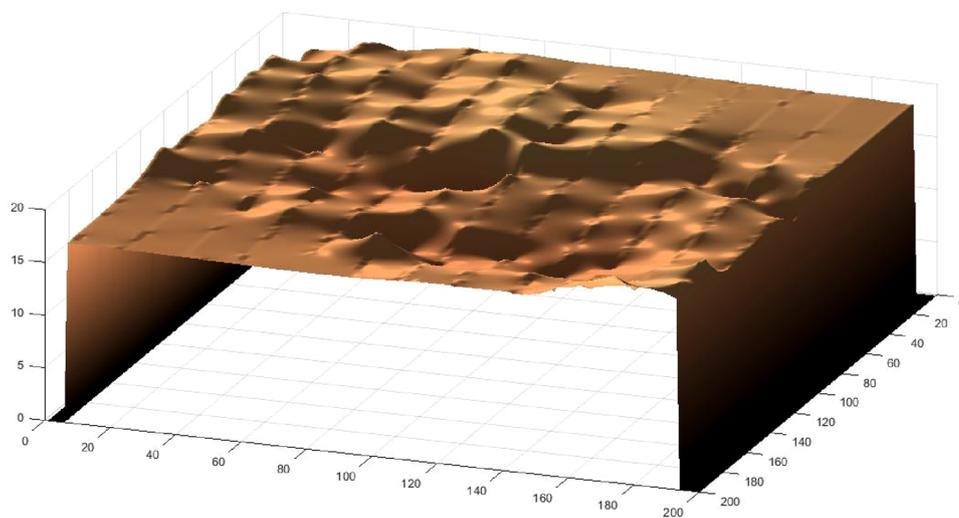
(b)
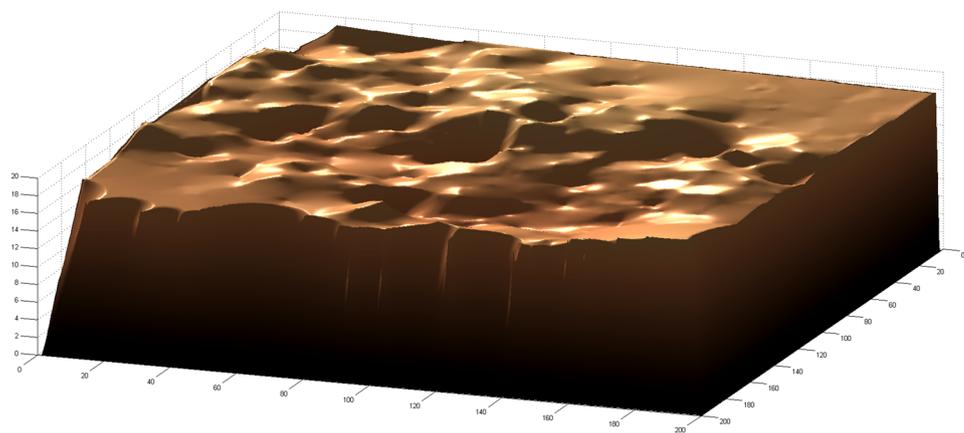
(c)

Fig. 10 *Ground truth*, *usual practice* and produced by the *proposed approach* Map employing 2 AUVs for #Slop_Surface. **a** *Ground truth* Map. **b** *usual practice* Map. **c** *proposed approach* Map based on 1000 samples
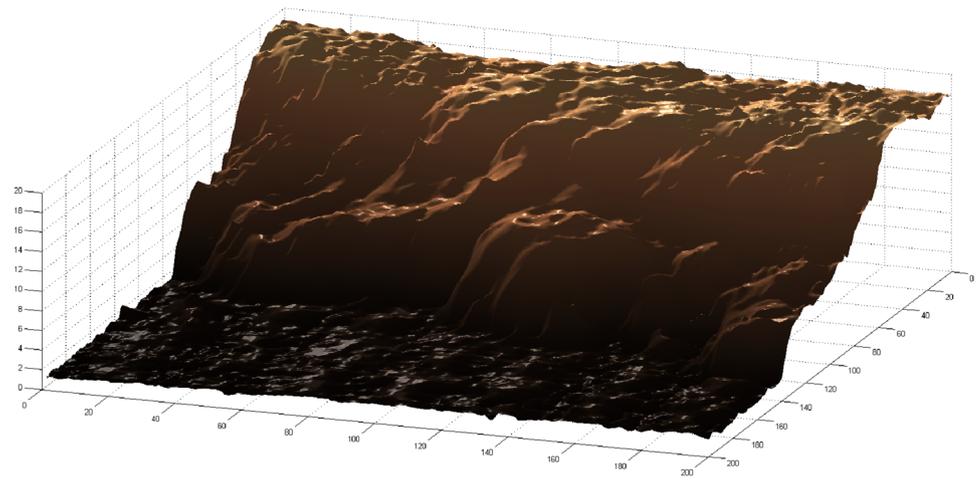
(a)
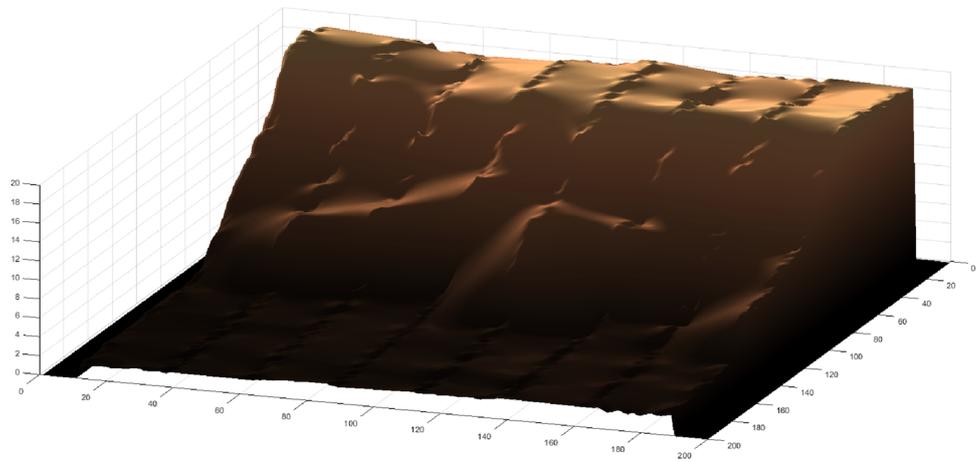
(b)
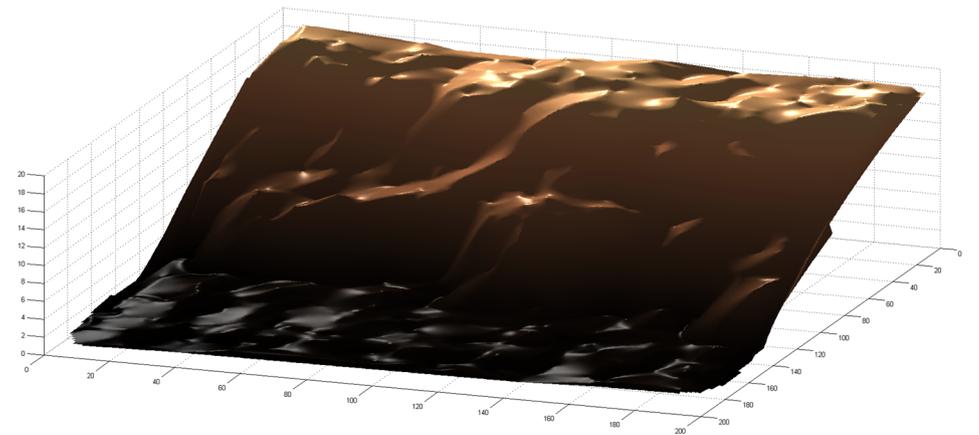
(c)

Table 2 L^2-$Norm$ and number of samples, between the *ground truth* version of the #Sharp_Surface map *vs* the *proposed approach* (cao-generated) map and the *usual practice* map

Approach	Number of samples	% of samples, w.r.t. ground truth (27961 samples)	L^2-norm accuracy
Proposed	1000	3.58	609.88
Usual	7503	26.83	900.27

Table 3 L^2-$Norm$ and number of samples, between the *ground truth* version of the #Slop_Surface map *vs* the *proposed approach* (cao-generated) map and the *Usual Practice* map

Approach	Number of samples	% of samples, w.r.t. ground truth (35846 samples)	L^2-norm accuracy
Proposed	1000	2.79	828.17
Usual	4509	12.58	846.24

Additionally, the proposed scheme, in relation to the vast majority of the optimal/dynamic programming approaches, takes into account the non-linear characteristics of the AXVs sensors and the fact that the operation area is unknown. In a nutshell, the proposed methodology aims to bridge the gap between the state-of-the-art algorithms and the actual practices, by successfully navigating the AXVs through environments, where the objective/reward function cannot be calculated a-priori, due to the afore-mentioned reasons.

The applicability and adaptability of our approach in realistic scenarios has been demonstrated through simulated and real-life underwater sea-floor mapping experiments in the port of Porto, Portugal using a team of AUVs. The proposed approach is independent of the SLAM methodology employed since it is based on the approach "to do the best it can be done" based on the current configuration, given the communication/sensing/SLAM system, allowing even cases where the multi-robot team comprises of vehicles with mutually different sensing capabilities or operating different SLAM algorithms.

With an outlook to the future work, we consider incorporating the localization problem to our decision making mechanism. The proposed approach can be safely classified under the spectrum of the optimization based ones. These algorithms allow to interface additionally/secondary objectives, by appropriately modifying/revising the performance criterion. Moreover, we will focus our efforts on the development of an approach that will primarily retain all the achievements of the proposed method (operate under unknown terrain, without actual evaluate the candidate configurations, etc.), but at the same time will able to provide near-globally-optimal solutions for all the experiment's horizon and not only for the next timestep. In order to build such a non-greedy algorithm, we should apply an offline learning scheme where the algorithm will translate the sensors' measurements into new commands/AXVs directions, by applying some transformation on them, that has been learnt from numerus simulation or/end real-life experiments.

References

Birk, A., Pfingsthorn, M., & Bülow, H. (2012). Advances in underwater mapping and their application potential for safety, security, and rescue robotics. In *IEEE International Symposium on Safety, Security, Rescue Robotics (SSRR)*. IEEE Press.

Reed, S., Wood, J., & Hawort, C. (2010). The detection and disposal od ied devices within harbor regions using auvs, smart rovs and data processing/fusion technology. In *2010 international Waterside Security Conference (WSS)*, (pp. 1–7).

Rodningsby, A., & Bar-Shalom, Y. (2009). Tracking of divers using probabilistic data association filter with a bubble model. *IEEE Transactions on Aerospace and Electronic Systems*, *45*(3), 1181–1193.

Kessel, R. T., & Hollett, R. D. (2006). Underwater intruder detection sonar for protection: State of the art review and implementations. In *IEEE International Conference on Technologies for Homeland Security and Safety*.

Murphy, R., Steimle, E., Hall, M., Lindemuth, D., Trejo, D., Hurlebaus, Z., Medina-Catina, Z., & Slocum, D. (2009). Robot-assisted bridge inspection after hurricane ike. In *2009 International Workshop on Safety, Security and Rescue Robotics (SSRR)*, (pp. 1–5).

Roman, C., & Mather, R. (2010). Autonomous underwater vehicles as tools for deep-submergence archaeology. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the the Maritime Environment*, *224*, 327–340.

DeVault, J. (2000). Robotic system for underwater inspection of bridge piers. *IEEE Instrumentation and Measurement Magazine*, *3*(3), 32–37.

Blondel, P. (2008). A review of acoustic techniques for habitat mapping. *Hydroacoustics*, *11*, 29–38.

Khurshid, J., & Bing-rong, H. (2004). Military robots—a glimpse from today and tomorrow. In *ICARCV 2004 8th International Conference on Control, Automation, Robotics and Vision Conference, 2004*, (Vol. 1, pp. 771–777).

Samad, A. M., Kamarulzaman, N., Hamdani, M. A., Mastor, T. A., & Hashim, K. A. (2013). The potential of unmanned aerial vehicle (uav) for civilian and mapping application. In *2013 IEEE 3rd International Conference on System Engineering and Technology (ICSET)*, (pp. 313–318).

Kosmatopoulos, E. B., Papageorgiou, M., Vakouli, A., & Kouvelas, A. (2007). Adaptive fine-tuning of nonlinear control systems with application to the urban traffic control strategy tuc. *IEEE Transactions on Control Systems Technology*, *15*(6), 991–1002.

Kosmatopoulos, E. B. (2009). An adaptive optimization scheme with satisfactory transient performance. *Automatica*, *45*(3), 716–723.

Kosmatopoulos, E. B., & Kouvelas, A. (2009). Large-scale nonlinear control system fine-tuning through learning. *IEEE Transactions Neural Networks*, *20*(6), 1009–1023.

Renzaglia, A., Doitsidis, L., Martinelli, A., & Kosmatopoulos, E. B. (2012). Multi-robot three-dimensional coverage of unknown areas. *The International Journal of Robotics Research*, *31*(6), 738–752.

Martijn, N. (2007). Rooker and Andreas Birk. Multi-robot exploration under the constraints of wireless networking. *Control Engineering Practice*, *15*(4), 435–445.

Pfingsthorn, M., Birk, A., & Bulow, H. (2010). An efficient strategy for data exchange in multi-robot mapping under underwater communication constraints. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (pp. 4886–4893). IEEE.

Johnson, B., Hallin, N., Leidenfrost, H., O'Rourke, M., & Edwards, D.. (2009). Collaborative mapping with autonomous underwater vehicles in low-bandwidth conditions. In *OCEANS 2009-EUROPE*, (pp. 1–7). IEEE.

Rajala, A., & Edwards, D. (2007). Allocating auvs for mine map development in mcm. IEEE.

Donald, Bruce, Xavier, Patrick, Canny, John, & Reif, John. (1993). Kinodynamic motion planning. *Journal of the ACM (JACM)*, *40*(5), 1048–1066.

Pasqualetti, Fabio, Franchi, Antonio, & Bullo, Francesco. (2012). On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms. *IEEE Transactions on Robotics*, *28*(3), 592–606.

Ny Le J., & Pappas, G. J. (2009). On trajectory optimization for active sensing in gaussian process models. In *Proceedings of the 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009*, (pp. 6286–6292). IEEE.

Milam, M. B., Mushambi, K., & Murray, R. M. (2000). A new computational approach to real-time trajectory generation for constrained mechanical systems. In *Proceedings of the 39th IEEE Conference on Decision and Control, 2000*, (Vol. 1, pp. 845–851). IEEE.

Kelly, Alonzo, & Nagy, Bryan. (2003). Reactive nonholonomic trajectory generation via parametric optimal control. *The International Journal of Robotics Research*, *22*(7–8), 583–601.

Low, K. H., Dolan, J. M., & Khosla, P. (2011). Active markov information-theoretic path planning for robotic environmental sensing. In *The 10th International Conference on Autonomous Agents and Multiagent Systems, Vol. 2*, (pp. 753–760). International Foundation for Autonomous Agents and Multiagent Systems.

Tabuada, Paulo, & Pappas, George J. (2005). Hierarchical trajectory refinement for a class of nonlinear systems. *Automatica*, *41*(4), 701–708.

Msechu, Eric J, Roumeliotis, Stergios I, Ribeiro, Alejandro, & Giannakis, Georgios B. (2008). Decentralized quantized kalman filtering with scalable communication cost. *IEEE Transactions on Signal Processing*, *56*(8), 3727–3741.

Zhou, Ke, & Roumeliotis, Stergios I. (2011). Multirobot active target tracking with combinations of relative observations. *IEEE Transactions on Robotics*, *27*(4), 678–695.

Feder, Hans Jacob S, Leonard, John J, & Smith, Christopher M. (1999). Adaptive mobile robot navigation and mapping. *The International Journal of Robotics Research*, *18*(7), 650–668.

Bourgault, F., Makarenko, A., Williams, S. B., Grocholsky, B., & Durrant-Whyte, H. F. (2002). Information based adaptive robotic exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002*, (Vol. 1, pp. 540–545). IEEE.

Stachniss, C., & Burgard, W. (2003). Exploring unknown environments with mobile robots using coverage maps. In *Proceedings of the International Conference on Artificial Intelligence (IJCAI)*.

Spletzer, J. R., & Taylor, C. J. (2003). Dynamic sensor planning and control for optimally tracking targets. *The International Journal of Robotics Research*, *22*(1), 7–20.

Beard, Randal W, McLain, Timothy W, Goodrich, Michael A, & Anderson, Erik P. (2002). oordinated target assignment and intercept for unmanned air vehicles. *IEEE Transactions on Robotics and Automation*, *18*(6), 911–922.

Nesterov, Y. (2007). Gradient methods for minimizing composite objective function. CORE Discussion Papers 2007076, Universit catholique de Louvain, Center for Operations Research and Econometrics (CORE).

Rathnam, Ravi Kulan, & Birk, Andreas. (2013). A distributed algorithm for cooperative 3d exploration under communication constraints. *Paladyn, Journal of Behavioral Robotics*, *4*(4), 223–232.

Fox, Dieter, Ko, Jonathan, Konolige, Kurt, Limketkai, Benson, Schulz, Dirk, & Stewart, Benjamin. (2006). Distributed multirobot exploration and mapping. *Proceedings of the IEEE*, *94*(7), 1325–1339.

De Hoog, J., Cameron, S., & Visser, A.. (2009). Role-based autonomous multi-robot exploration. In *Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009. COMPUTATIONWORLD'09. Computation World:*, (pp. 482–487). IEEE.

Freda, L., & Oriolo, G. (2005). Frontier-based probabilistic strategies for sensor-based exploration. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005*, (pp. 3881–3887). IEEE.

Burgard, W., Moors, M., Fox, D., Simmons, R., & Thrun, S. (2000). Collaborative multi-robot exploration. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation, ICRA'00*, (Vol. 1, pp. 476–481). IEEE.

Yamauchi, B. (1998). Frontier-based exploration using multiple robots. In *Proceedings of the Second International Conference on Autonomous Agents*, (pp. 47–53). ACM.

Kavraki, Lydia E, Svestka, Petr, Latombe, J.-C., & Overmars, Mark H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, *12*(4), 566–580.

Kavraki, E. E., Kolountzakis, Mihail N, & Latombe, J.-C. (1998). Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on Robotics and Automation*, *14*(1), 166–171.

Prentice, S., & Roy, N. (2009). The belief roadmap: Efficient planning in belief space by factoring the covariance. *The International Journal of Robotics Research*.

Kuffner, J. J., LaValle, S. M. (2000). Rrt-connect: An efficient approach to single-query path planning. In *Proceedings of the IEEE International Conference on Robotics and Automation, 2000, Proceedings. ICRA'00*, (Vol. 2, pp. 995–1001). IEEE.

LaValle, Steven Michael. (2006). *Planning algorithms*. Cambridge: Cambridge university press.

Valencia, R., Andrade-Cetto, J., & Porta, J. M. (2011). Path planning in belief space with pose slam. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 78–83). IEEE.

Birk, Andreas, & Carpin, Stefano. (2006). Merging occupancy grid maps from multiple robots. *Proceedings of the IEEE*, *94*(7), 1384–1397.

Kollar, Thomas, & Roy, Nicholas. (2008). Trajectory optimization using reinforcement learning for map exploration. *The International Journal of Robotics Research*, *27*(2), 175–196.

Martinez-Cantin, Ruben, de Freitas, Nando, Brochu, Eric, Castellanos, José, & Doucet, Arnaud. (2009). A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots*, *27*(2), 93–103.

Peng, Wu, Suzuki, Hiromasa, & Kase, Kiwamu. (2005). Model-based simulation system for planning numerical controlled multi-axis 3d surface scanning machine. *JSME International Journal Series C*, *48*, 748–756.

Doucet, A. (1998). On sequential simulation-based methods for bayesian filtering. Technical report.

Doitsidis, L., Weiss, S., Renzaglia, A., Achtelik, M. W., Kosmatopoulos, E. B., Siegwart, R., et al. (2012). Optimal surveillance coverage for teams of micro aerial vehicles in gps-denied environments using onboad vision. *Autonomous Robots*, *33*(1–2), 173–188.

Renzaglia, A., Doitsidis, L., Chatzichristofis, S. A., Martinelli, A., & Kosmatopoulos, E. B. (2013). Distributed multi-robot coverage using micro aerial vehicles. In *21st Mediterranean Conference on Control and Automation* (pp. 963–968). MED13 Greece: Chania.

Agile project's website.

Amanatiadis, A., Chatzichristofis, S. A., Charalampous, K., Doitsidis, L., Kosmatopoulos, E. B., Tsalides, P., et al. (2013). A multi-objective exploration strategy for mobile robots under operational constraints. *IEEE Access*, *1*, 691–702.

Ruppert, D., & Wand, M. P. (1994). Multivariate locally weighted least squares regression. *The Annals of Statistics*, (pp. 1346–1370).

LSTS lab. *Neptus Command and Control Software*. http://lsts.fe.up.pt/software/neptus.

LSTS lab. *DUNE: Unified Navigation Environment*. http://lsts.fe.up.pt/software/dune.

Athanasios Ch. Kapoutsis graduated from the Department of Electrical and Computer Engineering of the Democritus University of Thrace (D.U.Th.) (2007–2012). He is currently a PhD candidate at the Automatic Control Systems and Robotics Lab at the same department, as well as research assistant with the Centre for Research and Technology Hellas, Information Technologies Institute, Thessaloniki, Greece. During the past years, he has been involved in several EU FP7 and H2020 funded IP Research and Development projects. His research is mainly focused on robotics, machine intelligent, nonlinear and adaptive control, pattern recognition and image processing.

Savvas A. Chatzichristofis received the Diploma and Ph.D. degrees from the Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, Greece, in 2005 and 2010, respectively. He is currently a postdoctoral researcher with the Democritus University of Thrace, as well as a post-doctoral researcher with the Centre for Research and Technology Hellas, Information Technologies Institute, Thessaloniki, Greece. During the past years, he has been involved in several EU FP6, FP7 and H2020 funded IP and STReP Research and Development projects.

Lefteris Doitsidis received his diploma degree from the Production Engineering and Management Department of the Technical University of Crete, Chania, Greece, in 2000, 2002 and 2008, respectively. From 2002 to 2008, he has been a researcher at the Intelligent Systems and Robotics Laboratory of the same department. From August 2003 to June 2004 he was a visiting scholar at the Department of Computer Science and Engineering, University of South Florida, FL, USA. He was a member of the Center of Robot Assisted Search and Rescue. He was an adjunct senior researcher at the Informatics & Telematics Institute of Greece, CERTH, from 2010 until 2013. Currently he is an Assistant Professor at the Department of Electronic Engineering of the Technological Educational Institute of Crete, Greece. During the past years he has been involved in various research projects related to multi robot teams applications, funded by the EU. His research interests lie in the areas of multi robot teams, design of novel control systems for robotic applications, autonomous operation and navigation of unmanned vehicles, cooperative control and optimization. He is also active in the areas of fuzzy logic and evolutionary computation.

João Borges de Sousa is a lecturer at the Electrical and Computer Engineering Department at Porto University in Portugal and the director of the Underwater Systems and Technologies (LSTS) Laboratory at Porto University. He received the M. Sc. degree in Electrical and Computer Engineering from Porto University in 1992. He did his Ph.D. studies under the supervision of Prof. Pravin Varaiya from the University of California at Berkeley. His research interests include unmanned vehicle systems, networked control, control and coordination of multiple dynamic systems, hybrid systems, systems engineering, and control architectures for multi-vehicle systems. Since 1997 he has been leading the design, implementation and deployment of advanced unmanned vehicle systems in projects funded by the Portuguese Foundation for Science and Technology, Nato and the EU, in Europe, and by ONR and DARPA in the US. In 2006 he received the national BES Innovation National Award for the design of the Light Autonomous Underwater Vehicle. In 2007 he received an outstanding teaching award from Porto University. He authored more than 200 publications, including 30 journal papers.

Jose Pinto has a B.Sc. in Computer Science a postgraduate diploma in Software Engineering both from Porto University. Currently he is pursuing a Ph.D. degree on the topic "multi-robot coordination and planning". His main research activities at LSTS (Underwater Systems and Technology Laboratory) are on mixed-initiative planning interfaces, onboard plan deliberation and multi-agent systems.

Jose Braga has a M.Sc. in Eletronics Engineering from University of Porto (FEUP). He works as a software developer/reseacher at LSTS (Underwater Systems and Technology Laboratory) in FEUP, Portugal. His main activities focus on autonomous vehicles on-board software, navigation and sensor integration. In recent years José was involved in several experiments at sea using LSTS unmanned autonomous systems.

Elias B. Kosmatopoulos received the Diploma, M.Sc. and Ph.D. degrees from the Technical University of Crete, Greece, in 1990, 1992, and 1995, respectively. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, Greece. Previously, he was a faculty member of the Department of Production Engineering and Management, Technical University of Crete (TUC), Greece, a Research Assistant Professor with the Department of Electrical Engineering-Systems, University of Southern California (USC) and a Postdoctoral Fellow with the Department of Electrical & Computer Engineering, University of Victoria, B.C., Canada. Dr. Kosmatopoulos' research interests in the areas of nonlinear and adaptive control, robotics, energy-efficient buildings and intelligent transportation systems. He is the author of over 40 journal papers. He has been currently leading many research projects funded by the European Union with a total budget of about 10 Million Euros.